Deriving Unit Cost Coefficients for Linear Programming-Driven Priority-Based Simulations

By

INES CID LOPES FERREIRA
B.Sc. (University of Surrey, UK) 1985
G.C.E. (University of London, UK) 1986
M.S. (University of California, Davis) 1992
M.S. (University of California, Davis) 1993

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Civil Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

____

____

____

Committee in Charge

2007

ACKNOWLEDGEMENTS

I would like to thank Professor Jay Lund for the constant motivation, enthusiasm, and advice. Without his help and support this work would never have been completed. I am grateful to Mimi Jenkins for all her helpful comments, for keeping me grounded, and for the many words of encouragement when a solution seemed unattainable. I am indebted to Stacy Tanaka for her up to the last minute help with just about anything. I am also thankful to Marlene Elias Ferreira for taking the time to read this dissertation providing useful comments.

I would like to thank DWR for providing the funding, test case, and the feedback necessary to complete this work.

I am also thankful to Jim Byer of Sunset Technologies, who adapted the version of XA I had for my needs in this research and was always there to answer my questions. The use of XA simplified my work considerably.

Finally, this work would not have been completed without the support and patience from my family. I am grateful to my husband, Tracy, for his patience, encouragement, and love. I also thank my precious daughters, Emma and Olivia, for their patience and for accepting many weekends and vacations without their Mama.

I dedicate this work to my father, Wilson F. Ferreira, for his unfailing support and for teaching me the most valuable lessons.

ABSTRACT

Throughout the United States, water resources projects are experiencing reduced ability to fulfill demands. Increases in water demands have intensified competition over water allocation and operations. Water resources system models are often used to analyze trade-offs, facilitate better decision-making, and resolve conflict. Most newer water supply simulation models employ optimization methods to allocate water and operations according to fixed operational priorities for each time-step, simulating the efforts of capable system operators attempting to achieve a given set of operational priorities. For extensive complex networks with return flows, loops arising from pumping, and proportional delivery reductions for equal-priority deliveries, the assignment of unit weights in the objective function can be a matter of some art and controversy.

This dissertation presents a generalized method for automating the computation of unit weights that guarantees priority-preserving behavior for network flow programming-based simulation models and a step-by-step procedure to generate priority preserving weights for linear programming driven simulations models. Many test case examples are presented, including a LP driven model California's Central Valley. The examples illustrate various network configurations and how priority preserving weights are computed and used to allocate water by priority. An analysis of the LP driven simulation problem itself is presented to validate the use of the proposed procedure for LP driven simulations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Throughout the United States, water resources projects are experiencing limited ability to fulfill demand. While original yield estimates for many storage projects have been reduced due to environmental and dam safety considerations, water demand, both consumptive and non-consumptive, has increased substantially since the authorization of most storage projects. Increase in water demand intensified competition over water allocation. Moreover, with the construction of large-scale storage projects at a virtual standstill in the US, more effective use of existing resources becomes of paramount importance.

As the competition for water is heightened, the operation of storage projects became more complex. Several storage projects originally authorized for water supply and/or hydroelectric power generation now provide water for many other uses, including recreation and environmental protection and enhancement (Labadie 2004).

Despite the strain in the surface and ground water systems being at an all time high, never has there been greater ability and, in many cases willingness, to resolve disputes and reach compromise (Ferreira et al. 2005 and Palmer 2000). Advances in the research and application of operations research techniques have given water managers powerful analytical tools, which, coupled with substantial improvements in computer technology have greatly improved the ability to examine complex water allocation issues comprehensively and efficiently. Furthermore, the widespread use of computers in day-to-day life has caused a high level of familiarity and comfort with computers and their use.

Water resources systems models often are used to analyze trade-offs, facilitate better decision-making, and resolve conflict. Whether called shared vision modeling (Palmer 2000) or decision support systems (Sigvaldason 1976, Kuczera and Dimenti 1988, Andreu et al. 1996, Fredericks 1998, Eschenbach et al. 2001, Labadie 2004, and Karamouz et al. 2005), water resources system models are extensively used to resolve water disputes and improve the management of water systems.

Water resources models have evolved with time and technology. Site-specific simulation models were first developed in the 1950s. These include models of the Missouri River Basin, Potomac River Basin, Colorado River Basin, and the Sacramento River Basin (Wurbs 1996). More recently, generalized reservoir system simulation models that can be adapted and used to represent different reservoir systems, replaced some site-specific custom models. Notable examples of generalized simulation models are the vintage Hydrologic Engineering Center (HEC) models HEC-3 and HEC-5, and the more recent RES-SIM model.

Significant advances in operation research have given water resource engineers and academics efficient algorithms to solve a broad range of problems. Priority driven simulation models with optimization engines are widely used in many generalized water resources management models, particularly for multi-purpose and water supply applications. Linear Programming (LP) techniques, in particular, have been used extensively, both by academics and practitioners. The most commonly used sub-set of LP is the Network Flow Programming (NFP), for which efficient solution algorithms (Out-of-Kilter Algorithm, OKA, and variants) have been developed and are widely used.

Many generalized computer models use NFP engines with various versions of out-of-kilter algorithm for pure network flow formulations or primal-dual algorithms for generalized NFP formulations (allowing direct incorporation of gains and losses). Among them are MODSIM (Labadie et al. 1995, Frevert et al. 1994, Graham et al. 1996, Labadie et al. 1994, and Fredericks et al. 1998), CRAM (Brendecke et al. 1989), ARSP (Sigvaldason 1976), and WASP (Kuczera and Diment 1988), and HEC-PRM (Israel and Lund 1999). Additional applications of the OKA include DWRSIM in California (Chung et al. 1989) and KCOM (Andrews et al. 1992).

While the OKA and its variants are still widely used, those specialized NFP algorithms are no longer necessary (ReVelle et al. 2004). The dramatic increase of computer processing speed in recent years has allowed the successful use of more general LP formulation in simulation models. Examples of generalized models that are driven by mixed-integer LP solvers are CALSIM (Draper et al. 2004), OASIS (Hydrologics, Inc.), and WEAP (SEI 2001).

The great advantage of using priority driven optimization algorithms is that time and effort can be better spent in data development and system representation, leaving the allocation of water to be done by algorithms that are not only efficient, but being continuously improved by experts in operations research.

Since real operations attempt to maximize overall deliveries within a set of priorities and physical constraints, these simulation models also mimic actual operating objectives. The link among the LP (or NFP), the simulation, and the system operations are the objective function penalties (or weights, or unit cost coefficients), representing water allocation priorities. The objective function coefficients define a hierarchy of flow requirements in the system, and, in effect, define the operating policy and strategy.

While it may seem straightforward to set coefficients in the form of penalties or weights for a system regulated by the prior appropriation doctrine (most rivers in the western states), Israel and Lund (1999) have shown that, particularly where return flows are incorporated in the network flow formulation, setting coefficients is not a straightforward task. Also, many current uses of water, such as recreation and environmental flows, fall outside of the system of prior appropriation. To accurately represent systems' operations, the model user must assign unit penalty coefficients to these uses as well. In such cases, it is necessary to develop priority preserving unit penalties.

While a model developer with enough experience in applying this type of model may successfully determine unit cost coefficients, the procedure usually used is more art than science. Trial and error and experience are not easily duplicated, and the lack of a clear rationale for setting these parameters, may result in unnecessary controversy, particularly under the adversarial conditions. Furthermore, when no explanation can be given for the selection of a parameter, it becomes more permissible to "tinker" with its value. This tinkering often results in model user frustration or may exacerbate the suspicion that often dominates adversarial conditions, especially where different modelers use different weight sets.

Some experienced users of simulation models with LP and NFP engines suggest using weights that differ by orders of magnitude. While reducing the risk that model results are overly sensitive to the value of the unit cost coefficients, particularly for large networks, this approach could lead to a wide numerical range of coefficients and potentially numerical instability in the solution algorithm.

Israel and Lund (1999) proposed an algorithm to derive priority-preserving unit cost coefficients for network flow with gains. The algorithm is initially presented as a set of rules, which accommodates storage and flow related water uses over single or multiple simulation periods and accounts for the effects of return flows on water allocation. The rules are then compiled into an LP problem that is solved as a preprocessor to the actual simulation model. The purpose of the research presented here is to further test the Israel and Lund method to compute unit penalty coefficients for NFP driven simulation models and investigate its applicability to the more general LP formulation.

OUTLINE

This disseration has eight chapters and two appendices.

- This chapter introduces the topic and also contains a review of literature.
- Chapter 2 presents the automated method for unit penalty coefficient computation.
- Chapter 3 describes the algorithm implementation and examples.
- Chapter 4 describes the application of the algorithm to an LP driven model, the Two River System model of California's Central Valley.
- Chapter 5 discusses two methods for dealing with equal priorities.
- Chapter 6 focuses on the computation of negative weights.
- Chapter 7 presents an analysis of LP driven simulations and how the automated procedure for generating priority preserving weights can be implemented for LP driven simulations.
- Chapter 8 presents the summary, major conclusions, and some ideas for future research.
- Appendix A contains LP listings for examples from Chapter 3.
- Appendix B contains LP listings for test cases from Chapter 7.

CHAPTER 2: GENERALIZED ALGORITHM

Network flow programming (NFP) models have been used extensively to model prioritized water system operations. Commonly, the priority weights (or unit cost coefficients, c_k) used to simulate the system are derived by trial and error. While feasible for every model, trial and error procedures tend to be extremely time consuming for large models and can cause some concern for the reliability of simulated priority operations, especially where alternatives involve changes in priorities.

Israel and Lund (1999) present an algorithm for determining values for unit cost coefficients that preserve priorities for network flow programming models. The Israel and Lund method is described in this chapter, including improvements, simplifications and the re-writing of some equations. In a later chapter, the method is extended to simulations driven by linear program solvers.

PRIORITY PRESERVING ALGORITHM FOR NETWORK FLOW PROBLEMS

Network flow model consists of an objective function, mass balance constrains, and upper and lower capacity constraints. The NFP is usually set up as follows (Israel and Lund, 1999):

Minimize: $$Z = \sum_{k=1}^{K} c_k q_k \qquad (1)$$

Subject to:
i. mass balance at each node

$$\sum_{k \in Kin} a_k q_k = \sum_{k \in Kon} q_k \qquad \text{for all nodes n = 1, 2, … , N} \qquad (2)$$

ii. upper and lower capacity constraints for each arc

$$0 \le l_k \le q_k \le u_k \qquad \text{for all arcs k = 1, 2, … , K} \qquad (3)$$

where:

Z = total system penalty
N = number of nodes
K= number of arcs
q_k = flow entering arc k
c_k = cost or penalty per unit flow in arc k
a_k = flow multiplier for arc k
K_{in} = arcs flow into node n
K_{on} = arcs flow out of node n
l_k = lower bound flow for arc k
u_k = upper bound flow for arc k

The priority preserving algorithm is based on two main principles: (i) senior unit penalties must exceed the combined junior unit penalties for any feasible competing space-time path through the system for any unit of water potentially available at the senior location; and (ii) the set of priority-preserving unit penalties is non-unique.

Israel and Lund (1999) present the algorithm in three ways. First, the algorithm is introduced in the form of nine rules, depending on the type of water priority use (consumptive, with or without return flows, instream, and storage) and its location relative to junior priorities. As the rules are linear, they are combined to form the bulk of the constraints of an LP, the solution of which provides the unit cost coefficients for the network flow model. Lastly, the algorithm is generalized for a location connectivity matrix and a vector of unit priority weights. In this last section, linear algebra is extensively used. The purpose of this chapter is to describe the algorithm presented in Israel and Lund and to correct some linear algebra errors found in the last section of the paper.

RULES FOR COMPUTING PRIORITY-PRESERVING UNIT COST COEFFICIENTS

The guidelines for determining priority-preserving unit cost coefficients are presented as rules in which a senior priority (consumptive uses, instream flow, or storage) is compared to junior priorities. The most junior priority in the system is used as a baseline and unit penalties are determined based on the junior priorities, in order of increasing priority.

Senior flow priority:

Rule1: Upstream senior without return vs. downstream juniors

$$P_s > \max_i \left\{ \sum_{j=1}^{Ni} P_j \right\} \qquad (4)$$

where P_s represents the unit penalty on a diversion with senior priority and P_j represents all the (N_i) junior priorities j, and i represents all the possible stream paths between the senior priority and the flow sink. This rule simply states that, for an upstream senior, the senior unit penalty must exceed the sum of all junior priorities on all flow paths between the senior demand and the flow sink.

Rule 2: Upstream senior with return vs. downstream juniors

$$P_s > \max_i \left\{ (1 - a_s) \sum_{j=1}^{Ni} P_{dj} + \sum_{j=1}^{Mi} P_{uj} \right\} \qquad (5)$$

where a_s $(0 \le a_s \le 1)$ is the return flow factor for the senior user, P_{dj}, all (N_i) junior users downstream of the senior return flow, and P_{uj} represents all (M_i) downstream junior users upstream of the senior return flow location. When $a_s = 0$, this rule reduces to Rule 1.

Rule 3: Downstream Senior with upstream junior return flows

$$P_s > \frac{P_j}{1 - a_j} \quad \text{for each upstream j with } a_j < 1 \qquad (6)$$

where a_j is the return fraction of the upstream junior demand.

Rule 4: General flow-based seniority penalty

For senior demands with both up and downstream users, the senior unit penalty cost is the greatest of the largest upstream and downstream values.

$$P_s > \max\left\{\max_i\left\{(1-a_s)\sum_{j=1}^{Ni} P_{dj} + \sum_{j=1}^{Mi} P_{uj}\right\}, \frac{P_j}{1-a_j}\right\} \quad (7)$$

Storage related:

Rule 5: Storage vs. storage priorities

If only storage priorities exist in the system, the senior storage priority (P_{ss}) must be greater than the next highest junior priority (P_{sj})

$$P_{ss} > P_{sj} \quad (8)$$

Rules 9, 10, and 11 follow directly from rules 4, 6, and 7.

Rule 6: Senior storage with downstream junior flow priorities

$$P_{ss} > \sum_{j=1}^{N} P_j \quad \text{for all downstream j} \quad (9)$$

Rule 7: Senior storage with upstream junior flow priorities

$$P_{ss} > \frac{P_j}{1-a_j} \quad \text{for each upstream j where } a_j \neq 1 \quad (10)$$

Rule 8: Mixture of Storage and Flow Priorities

$$P_{ss} > \max\left\{\sum_{j=1}^{N} P_j, \frac{P_j}{1-a_j}, P_{sj}\right\} \quad (11)$$

Storage and Flow related:

Rule 9: Senior Flow vs. Junior mix of storage and flow priorities

$$P_s > \max\left\{\max_j\left\{\frac{P_j}{1-a_j}\right\}, T\cdot\max(P_{sj}), \max_i\left\{(1-a_s)\sum_{j=1}^{Ni} P_{dj} + \sum_{j=1}^{Mi} P_{uj} + T\cdot\max(P_{dsj})\right\}\right\} \quad (12)$$

LP FOR ASSIGNING UNIT PENALTIES

The rules for determining unit penalty coefficients for priority-based penalty functions are formulated as a LP. To avoid scaling problems, the objective function is set to minimize the difference between the highest and the lowest penalty costs. A ranking (or ordinal) rule is added to ensure that each penalty coefficient is greater than the penalty coefficient for the next junior user (equation 14). Also, the penalty coefficient for the most junior user, P_N, is set to a baseline value (equation 21).

Minimize: $Z = P_1 - P_N$ (13)

Subject to:

$$P_p \geq P_{p+1} + \varepsilon \qquad \forall p = 1,\ldots,N-1 \quad (14)$$

$$P_p \geq (1-a_p)\sum_{j>p}^{K} P_j^{'} + \sum_{j>p}^{L} P_j^{''} + TP_{ds,p+1} + \varepsilon \qquad \forall p = 1,\ldots,N \quad (15)$$

$$P_p \geq \left(\frac{1}{1-a_j}\right)P_j + \varepsilon, \quad \text{for all upstream juniors j; p=1,…,N} \qquad (16)$$

$$P_p \geq T \cdot P_{sj} + \varepsilon, \quad \text{for all junior reservoirs j; p=1,…,N} \qquad (17)$$

$$P_{sp} \geq P_{sj} + \varepsilon, \quad \text{for all junior reservoirs j; p=1,…,N} \qquad (18)$$

$$P_{sp} \geq \sum_{j=1}^{N} P_j + \varepsilon \quad \forall j = p+1,..., N; \quad \text{p=1,…,N} \qquad (19)$$

$$P_{sp} \geq \left(\frac{1}{1-a_j}\right)P_j + \varepsilon, \quad \text{for all upstream juniors j; p=1,…,N} \qquad (20)$$

$$P_N = \text{Base} \qquad (21)$$

GENERALIZED ALGORITHM

While the LP formulation can successfully generate unit penalties, the setting of all LP constraints becomes unwieldy as the number of water use priorities increases. In the section entitled “Generalized Algorithm with Network Connectivity Matrices”, Israel and Lund present a generalization of the LP formulation based on network connectivity matrices. The use of connectivity matrices allows for automation of the process of setting up all the necessary LP constraints. Although conceptually correct, a few linear algebra errors were found in the Israel/Lund paper. These errors are corrected in this section. Additional simplifications are made, demonstrating that the nine equations can be reduced to three equations when considering a single time-step optimization.

The equations in the generalized algorithm contain the following vectors and matrices:

- **M** is the location connectivity matrix. **M** is a square matrix, of size *nxn*, where *n* is number of locations (water users) of interest. Its elements, m_{ij}, indicate the ability to move water from j to i. If $m_{ij} = 1$, water can move from location j to location i. Conversely, if $m_{ij} = 0$, water cannot move from location j to location i. Each column k of **M**, defined as $\vec{M}^{k}$, represents the vector of locations downstream of k. The diagonal elements, m_{ii}, indicate whether location i represents a storage node ($m_{ii} = 1$) or not ($m_{ii} = 0$).
- **S** is the storage matrix. It is a diagonal matrix in which the diagonal entries are the same as the diagonal entries of **M**, i.e., $s_{ij} = m_{ij}$ if i = j, and $s_{ij} = 0$ elsewhere.
- **L** is the loss matrix. It is a diagonal matrix in which the diagonal entries $l_{ii} = 1/(1-r_j)$ for $r_j \neq 1$. When $r_j = 1$, then $l_{ii} = 0$ (i.e., no return from that use/diversion).
- The unit penalty vector $\vec{P} = (P_A P_B P_C ...)^T$ contains the decision variable set. These are the unknown values of the LP, or unit penalty for each of N water uses, which may occur at any location.

- The vector $\vec{P}^{j}$ is defined as the vector of uses junior to some particular senior water use. The vector $\vec{P}_{u}^{j}$ is the vector of upstream uses junior to a particular senior water use.

The matrix operations in equations (22) to (30) below yield either a scalar (1x1 matrix) or a nx1 vector. When the matrix operations result is a vector, the operator $\max_i$ selects the greatest entry from the vector.

Senior flow vs. Junior Flow

Equations (22) and (23) relate a senior flow to upstream and downstream junior flows, respectively. Equation (22) is the matrix form of Rule 2 (equation 5) and equation (23) is the matrix form of Rule 3 (equation 6). The matrix operation on the right-hand side of equation (22) yields a scalar, while $L\vec{P}_{u}^{j}$ equation (23) yields a *nx1* vector. The operator *max*$_i$ selects the greatest row value of $L\vec{P}_{u}^{j}$. The superscripts s and r of **M** refer to the location of the senior use and the return flow location of diversion to senior user s, respectively.

$$P_s \geq (1-d_s)\vec{P}^{jT}\vec{M}^{r} + \vec{P}^{jT}(\vec{M}^{s} - \vec{M}^{r}) \qquad (22)$$

$$P_s \geq \max_i\left(L\vec{P}_{u}^{j}\right) \quad \text{where i is the row number, i = 1, … , n} \qquad (23)$$

Senior Storage vs. Junior Storage

Equation (24) relates a senior storage to all junior storages, both up and downstream. It is equivalent to Rule 5 (equation 8). Equation (24) computes the greatest row value of the *nx1* vector $S\vec{P}^{j}$.

$$P_{ss} \geq \max_i\left(S\vec{P}^{j}\right) \quad \text{where i is the row number, i = 1, … , n} \qquad (24)$$

Senior Storage vs. Mix of Junior Flow and Storage

Equations (25) and (26) relate a senior storage penalty to downstream and upstream junior flows penalties, respectively. Equation (27) is the same as equation (24), relating storage to junior storage both up and downstream. The combined set (equations 25 to 27) is equivalent to Rule 8 (equation 11).

$$P_{ss} \geq \vec{P}^{jT}\vec{M}^{s} \qquad (25)$$

$$P_{ss} \geq \max_i\left(L\vec{P}_{u}^{j}\right) \quad \text{where i is the row number, i = 1, … , n} \qquad (26)$$

$$P_{ss} \geq \max_i\left(S\vec{P}^{j}\right) \quad \text{where i is the row number, i = 1, … , n} \qquad (27)$$

Senior Flow vs. mix of junior flow and storage uses

Equation (28) relates a senior flow to downstream junior flows and storage. Equations (29) and (30) relate a senior flow penalty to the penalties of upstream junior flow and storage users, respectively. Equations (28) to (30) are equivalent to Rule 9 (equation 12).

$$P_s \geq (1-d_s)\vec{P}^{jT}M^{r} + \vec{P}^{jT}(\vec{M}^{s} - \vec{M}^{r}) + (S\vec{P}^{j})^{T}\vec{M}^{s} \qquad (28)$$

$$P_s \geq \max_i\left(L\vec{P}_{u}^{j}\right) \qquad (29)$$

$$P_s \geq T \cdot \max_i \left(S\vec{P}_u^{\,j} \right) \quad (30)$$

Equations (22) to (30) represent, in matrix form, the LP constraints (15) to (20) in algebraic form. To complete the constraint set, equations (14) and (21) must be added to the equation set (22) to (30).

EQUATION SIMPLIFICATION

The two sets of equations defined above, in algebraic and matrix (generalized) forms, can be significantly simplified, particularly when the NFP is used to drive the simulation for a single time step (i.e., T=1). This simplification is seen by examining the constraints in their algebraic form, equations (13) to (21).

If T=1, then equation (17) simply states that the penalty coefficient for a senior must exceed the penalty coefficient for a junior storage priority. However, the ordinal character of the decision variables (equation 14) guarantees that equation (17) is satisfied. The same reasoning applies to equation (18). Consequently, both equations (17) and (18) can be eliminated.

Equation (19) is a particular case of equation (15). Equation (19) relates upstream storage seniors to downstream junior priorities. Similarly, equation (15) relates upstream flow seniors to downstream junior priorities. However, for a senior storage user P_p, a_p=0, so that equation (15) reduces to equation (19). Therefore, equation (19) can be eliminated.

Equations (16) and (20) are essentially the same equations but for the left hand side, where the senior priority is for flow (16) or storage (20). By allowing the left hand side to be either storage or flow, one of these equations can be eliminated.

In summary, for T=1, the LP problem, in algebraic form, becomes:

Minimize: $$Z = P_1 - P_N \quad (31)$$

Subject to:

$$P_p \geq P_{p+1} + \varepsilon \qquad \forall p = 1,..., N-1 \quad (32)$$

$$P_p \geq (1 - a_p)\sum_{j>p}^{K} P_j^{'} + \sum_{j>p}^{L} P_j^{''} + P_{ds,p+1} + \varepsilon \qquad \forall p = 1,..., N \quad (33)$$

$$P_p \geq \left(\frac{1}{1 - a_j} \right) P_j + \varepsilon, \quad \text{for all upstream juniors j; p=1,…,N} \quad (34)$$

$$P_N = \text{Base} \quad (35)$$

Following the same logic as above and adding the ordinal rule and the starting base, the generalized algorithm becomes:

$$P_p \geq P_{p+1} + \varepsilon \qquad \forall p = 1,..., N-1 \quad (36)$$

$$P_s \geq (1 - d_s)\vec{P}^{jT} M^r + \vec{P}^{jT}(\vec{M}^s - \vec{M}^r) + (S\vec{P}^j)^T \vec{M}^s \quad (37)$$

$$P_s \geq \max_i \left(L\vec{P}_u^{\,j} \right) \qquad \text{where i is the row number, i = 1, … , n} \quad (38)$$

$$P_N = \text{Base} \qquad (39)$$

To each equation generated by equations (37) to (39), a small constant ε is added. This constant allows the user to determine the smallest difference between consecutive priorities.

The implementation of several test cases is described in the following chapter.

SUMMARY

This chapter presents a method to derive unit cost coefficients for LP driven simulations. The algorithm presented is based on the algorithm proposed by Israel and Lund (1999), with some corrections and simplifications. The generalized matrix form of the algorithm simplifies its implementation considerably, particularly for large scale simulation models.

Next chapter describes the implementation of the generalized algorithm to NFP driven simulations with a number of simple examples. Chapter 4 focuses on the algorithm application to an LP driven simulation.

CHAPTER 3: ALGORITHM IMPLEMENTATION AND EXAMPLES

Following the method described in Chapter 2, a preprocessor program that computes unit cost coefficients was developed. The preprocessor generates the LP constraints (equations 36-39) in the form $A\vec{x} \geq \vec{b}$, and provides the LP solver, XA Software (Sunset Software Technology), the matrix ***A***, the vector $\vec{b}$, and the objective function. XA solves the LP providing unit penalty coefficients for a generalized NFP model formulation.

ALGORITHM IMPLEMENTATION

As described in Chapter 2, the generalized algorithm greatly simplifies formulating the LP constraints. Instead of writing constraints for each user, the generalized algorithm automates the LP definition procedure. To implement the generalized algorithm, a program was written in FORTRAN 90. Using equations (36) to (39) the LP generator program sets up the objective function (equation 31) and the constraint set (equations 32-35). The constraint set is stored in matrix form as $A\vec{x} \geq \vec{b}$, where *A* is a matrix of coefficients, $\vec{x}$ is the vector of decision variables, and $\vec{b}$ is the vector of constants. The objective function, the matrix *A,* and the vector $\vec{b}$ are passed to the LP solver, which computes decision variables values (unit penalties for the LP) that are then used as simulation model objective function weights (penalties or costs).

If no water users have equal priorities, the matrix *A* has $(n^2 + 2n)$ rows and *n* columns, the vector of decision variables $\vec{x}$ has dimension (*n*), and the vector of constants $\vec{b}$ has dimension $(n^2 + 2n)$. The entries in the decision variable vector $\vec{x}$ are in order of priority, so that x_1 is the highest priority and x_n the lowest. If two or more water users have the same priority, the number of rows of A will be greater. The actual number of rows will depend on the number of repeated priorities and their locations within the network.

This chapter examines a simple mainstem network, a downstream branching network, a downstream branching network with upstream tributaries, and a looped network with return flows. Systems with equal priorities will be examined in Chapter 5.

The inputs to the LP generator program are:

1. The number of decision variables, *n*. For simple network configurations, *n* is the same as the number of water users. For complex networks, with branching and merging of network links, *n* can exceed the number of users.
2. The (*n*x*n*) location connectivity matrix, **M**.
3. The (*n*x*1*) vector of priorities for each user/location *i* in the network.
4. The (*n*x*1*) vector of return flow locations for each user/location *i* in the network.
5. The (*n*x*1*) vector of return flow factors for each user/location *i* in the network.
6. The constant ε, the smallest difference between any two unit penalties (or weights).

7. Parameter *dim*, is used to allocate computer memory. Its value must be greater than the largest anticipated number of rows of matrix A.

Entries in the location connectivity matrix and the input vectors are ordered from upstream to downstream. As the program reads the inputs, it assigns a location number i ($1 \le i \le n$) to each of the n priorities/users in the network, in the order in which the information regarding that priority is read (from upstream to downstream).

To simplify the programming of the LP generator, the program loops through all priorities for each of the three equation types. This results in the generation of more constraints than if the LP were set up manually. However, these additional constraints are redundant and are discarded by the LP solver. As seen later in this chapter, in addition to simplifying the coding of the LP generator program, creating these redundant constraints maintains the same structure of the constraint set for all networks. This structure enables the user to identify each constraint by its number and to associate it to the rule and user/priority to which it refers, which aids in any interpretation.

LP GENERATOR CODE

The LP generator code consists of five modules written in FORTRAN 90, including an interface with the XA dynamic link library (dll) and library (lib) files.

After reading the input file, the program:

1. Determines if more than one user has the same priority.
2. Determines which priority is a storage priority by extracting the main diagonal of **M**, the location connectivity matrix. These values are stored in a one-dimensional array.
3. Calls a subroutine that generates the priorities vector $\vec{P}^j$ (users junior to a particular senior water user, in order of appearance in the network). These n vectors are stored in a two-dimensional array.
4. Generates the LP constraints:
   i. The ordinal rule, $P_p \ge P_{p+1} + \varepsilon$. If there are no repeated priorities, n equations are created. (40)
   ii. The downstream rule, $P_s \ge (1-d_s)\vec{P}^{jT} M^r + \vec{P}^{jT}(\vec{M}^s - \vec{M}^r) + (S\vec{P}^j)^T \vec{M}^s$. This equation generates n equations. (41)
   iii. The upstream rule, $P_s \ge \max_i \left(L\vec{P}_u^j\right)$. This equation generates n[2] constraints, as it relates each priority to all other priorities. The LP determines the largest value of $L\vec{P}_u^j$. (42)
5. The objective function $Z = P_1 - P_N$ is set. (43)
6. The LP solver (XA) is called and the unit costs coefficients are calculated.

EXAMPLES

The procedure is tested for several network configuration and priority combinations. Prototype river systems are created with corresponding input files. The LP generator computes the unit costs coefficients, which are then used in a simulation of the system to test that water is allocated properly by priorities. The system simulation is performed with the CalSim software, developed by the California Department of Water Resources (Draper et al, 2004). To simplify the testing procedure, the LP generator creates one additional output file, the *weight-table.wresl*, the CalSim input file containing objective function weights.

Initial tests and debugging of the program were carried out with a simple single branch network. Detailed explanation of the LP generated for this example is given below. Explanation of later examples explanation will be more succinct.

Single Maintem Network

Description

The simplest network configuration created to test the weight generator procedure, Network 1, appears in Figure 1. Network 1 consists of two reservoirs and five diversions on a single stretch of river. There are no branching or tributaries in Network 1. Inflow to the system occurs at the upstream reservoir, S1.

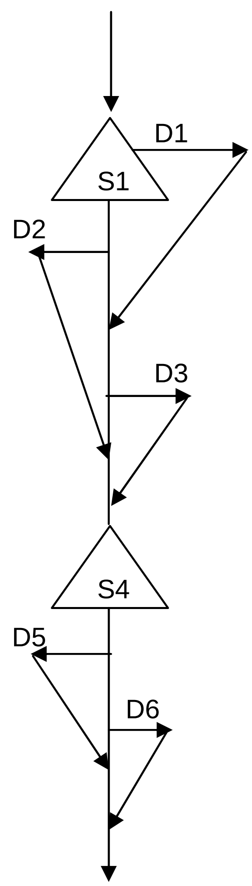

Figure 1. Network Schematic for Single Stem

The input file for a sample problem based on Network 1 is shown in Figure 2. The location connectivity matrix and the *return location* vector, rl(i), uniquely define the network configuration, i.e., its links, nodes, and their relative location. The column with heading CalSim nodes contains labels for the network nodes used to create the CalSim input file *weight-table.wresl.* The columns labeled *Priority* and *Return Factor* contain the priority and return flow factor for each of the seven water demands. For storage demands, the return flow fraction is zero. Priorities and return flow factors are independent of the network configuration and the corresponding columns can be modified to create additional examples with the same network. This chapter presents one example problem using this network.

```
7     | number of columns (n)
100   | dimension of A matrix (rows) must be >n*n + 2n
1.0   | epsilon
1.0   | baseline weight
                                 Priority   Return Loc   Return Factor    CalSim
S1 D1 D2 D3 S4 D5 D6                           rl(i)         r(i)          Nodes
1  0  0  0  0  0  0                 6            1            0.            S1
1  0  0  0  0  0  0                 5            4            .5            D1
1  1  0  0  0  0  0                 1            5            .5            D2
1  1  1  0  0  0  0                 7            5            .5            D3
1  1  1  1  1  0  0                 4            6            0.            S4
1  1  1  1  1  0  0                 2            8            .5            D5
1  1  1  1  1  1  0                 3            8            .5            D6
```

Figure 2. Input File for Example 1.

As the LP generator program reads the inputs, it assigns a location number i (1≤i≤n) to each of the n priorities in the network, in the order in which the information regarding that priority is read (from upstream to downstream). In this example, S1, D1, D2, D3, S4, D5, D6, are assigned the location numbers 1 through 7, respectively. The return flow location (column rl(i)) points to the location (or user) in the network at which a return flow first becomes available. For instance, the return from D3 first becomes available at location 5, that is, S4. Returns from D5 and D6 occur downstream of the last demand in the network, consequently, their return location is n+1 (in this case 8).

Appendix A-1 lists the XA solver output for this example. It contains the LP, its solution, and additional information about the LP. The LP generated, once clearly redundant constraints are removed, is shown in Figure 3. Once redundant constraints are removed, the original 63 constraints (n^2+2n) are reduced to 16 constraints.

Constraints C1 to C7 reflect the ordinal rule, $P_p \geq P_{p+1} + \varepsilon$. This rule translates into constraints in the form $x_j - x_{j+1} >= \varepsilon$. The constraints are ordered from upstream to downstream, and the decision variable subscript j refers to the priority, so that x_1 represents the highest priority and x_n the lowest.

The next n=7 constraints (constraints 8 to 14) listed in Appendix A-1 refer to the downstream rule $P_s \geq (1-d_s)\vec{P}^{jT} M^r + \vec{P}^{jT}(\vec{M}^s - \vec{M}^r) + (S\vec{P}^j)^T \vec{M}^s$. Constraint 8 compares the unit cost coefficient of S1 (priority=6, and location=1) to downstream users. In this case the only demand junior to S1 is D3, so the equation generated is $X_6 - x_7 >= 1$. This constraint is the same as constraint C1, and thus redundant. Constraint 9 compares the unit cost coefficient of D1 to all downstream juniors, in this case only D3. Because D3 is below the point of return for D1, its coefficient is (1-r(2))=0.5. The resulting equation is x_5-0.5x_7>=1.

The tenth constraint relates D2 to junior downstream users. Because D2 has the highest priority in this network, its equation is the most complex and contains all downstream users (which are junior to it). Users having a point of diversion upstream of the return flow of D2 and storage demands have a coefficient 1, while the users with diversions located downstream of the point of diversion of D2 have coefficient (1-r(3))=0.5. The resulting equation is shown as C10 in Figure 3.

Because each of the next two users in the network (D3 and S4) do not have juniors downstream of themselves, the downstream constraint (equation 2) is reduced to x_j>= 1, where j corresponds to the priority. Constraints of this kind are redundant, as they are guaranteed by the ordinal rule. Demand D5 is located upstream of a junior user, and generates constraint C14, $x_2 - x_3 \geq 0$.

```
Minimize: X1 - X7

Subject to:
C1: X6 - X7 >= 1
C2: X5 - X6 >= 1
C3: X1 - X2 >= 1
C4: X7 >= 1
C5: X4 - X5 >= 1
C6: X2 - X3 >= 1
C7: X3 - X4 >= 1
C9: X5 - 0.5 X7 >= 1
C10: X1 - 0.5 X2 - 0.5 X3 - X4 - X7 >= 1
C30: X1 - 2 X5 >= 1
C44: X4 - 2 X5 >= 1
C46: X4 - 2 X7 >= 1
C51: X2 - 2 X5 >= 1
C53: X2 - 2 X7 >= 1
C58: X3 - 2 X5 >= 1
C60: X3 - 2 X7 >= 1
```

Figure 3. LP for Example 1.

The remaining n^2=49 constraints in this LP relate a senior user to each upstream junior user. As every demand is compared to all other demands, many of these 49 constraints are redundant. Unless the comparison is between a downstream senior to an upstream junior, the constraint generated will be x_j>=1. Also, an upstream junior storage (return flow = 0) or a diversion without return flow will generate the same constraint, x_j>=1. Because the constraint generator creates constraints from upstream to downstream, the first n=7 constraints refer to S1. S1 is the most upstream node in the network, therefore, constraints 15 to 21 (Appendix A-1) are simply x_6 >= 1. For the next n=7 constraints, 22 to 28, D1 is compared to all other demands. There are no junior flow demands upstream of D1, therefore, constraints 22 to 28 are x_5 >= 1. D2 is the next user in the network. The only junior flow demand upstream of D2 is D1, resulting in constraint C30. For the remaining demands, D3, S4, D5, and D6, a similar analysis can be made, where an equation of the form $x_j - x_k /(1-r_k) >= 1$ will be generated for a senior demand s having a junior flow demand k located upstream of senior j, where r_k is the return flow fraction for the upstream junior. Preprocessor LP results are presented in Table 1.

Another output of the preprocessor is the CalSim input *weight-table.wresl* (Figure 4) which contains the objective function weights for use in a CalSim simulation of the system.

Table 1. Summary Results for Example 1

Demand	Priority	Weight
S1	6	2
D1	5	3
D2	1	17.5
D3	7	1
S4	4	7
D5	2	9
D6	3	8

```
Objective obj = {[S1,   2.00],
[D1  ,   3.00],
[D2  ,  17.50],
[D3  ,   1.00],
[S4  ,   7.00],
[D5  ,   9.00],
[D6  ,   8.00]}
```

Figure 4. **xample 1.**

<u>Test and Results</u>

Using the CalSim software, simulation models were developed to test that the weights generated by the preprocessor result in the desired water allocation. Except for the objective function weights file, all other CalSim system input files were created manually. They are listed in Appendix A-2.

This section presents two test runs performed using Example 1. For these two test runs, flow demands were set 10 taf per month and storage capacity for both reservoirs was set to 80 taf. Initial storage for both reservoirs was set to zero. For the first test run, the inflow was set to 10 taf every month. The simulated water allocation was as expected, 10 taf for D2. Its return flow was diverted by D5, the user with highest priority downstream of the point of return of D2.

The second test case was devised to verify water allocation with increasing amounts of available water. The inflow was set to 10 taf for the first month and increased by 10 taf per month for subsequent months. Table 2 presents the simulation results.

In the first month of the simulation, the 10 TAF inflow is diverted by the highest priority user, D2. The 5 TAF return flow from D2 is captured by D5, the next highest priority below the point of return of D2. In the second month, D2 and D5 are each allocated 10 TAF and 5 TAF return from D2 is diverted by D6 while the return from D5 and D6 flow out of the system. In the third month, D2, D5, and D6 demands are fully met. The return flow from D2 is stored in S4 while the returns from D5 and D6 flow out of the system. Of the 40 TAF inflow in the fourth month, 10 TAF each are diverted by D2, D5, and D6. These diversions consume 15 TAF, 10 TAF flow out of the system and the remainder 15 TAF is stored in S4. In the fifth and sixth month, a similar pattern of diversions is observed, with S4 storing the available 25 and 35 TAF, respectively. This brings the storage at S4 to its capacity of 80 TAF. In the seventh month, D2, D5, and D6 again divert 10 TAF each. The next in priority S4 is at capacity, so that D1 can divert 10 TAF and the remaining 40 TAF can be stored in S1.

In the eighth and subsequent months, both reservoirs are at capacity, all demands are met and the outflow from the system increases as the inflow to the system increases. The water allocation for this example occurred exactly as one would have expected given the original priorities. More examples using Network 1 are presented in Chapter 5.

Table 2. Simulated Water Allocation for Example 1

Month	I1 Inflow TAF	S1 Storage TAF	D1 Delivery TAF	D2 Delivery TAF	D3 Delivery TAF	S4 Storage TAF	D5 Delivery TAF	D6 Delivery TAF
1	10	0	0	10	0	0	5	0
2	20	0	0	10	0	0	10	5
3	30	0	0	10	0	5	10	10
4	40	0	0	10	0	20	10	10
5	50	0	0	10	0	45	10	10
6	60	0	0	10	0	80	10	10
7	70	40	10	10	0	80	10	10
8	80	80	10	10	10	80	10	10
9	90	80	10	10	10	80	10	10
10	100	80	10	10	10	80	10	10
11	110	80	10	10	10	80	10	10
12	120	80	10	10	10	80	10	10
Priority		**6**	**5**	**1**	**7**	**4**	**2**	**3**
Weight		**2**	**3**	**17.5**	**1**	**7**	**9**	**8**

Branching Network

Description

Network 2 (Figure 5) is the simplest branching network created in this study. It consists of one upstream reservoir and seven diversions. There are no duplicate priorities or tributaries, but the river branches into two channels, and diversions and returns occur in either one of the channels. As with Network 1, the inflow to the system occurs at the upstream reservoir, S1. The input file for a problem based on this network, Example 2, is shown in Figure 6.

Appendix A-3 lists the XA solver output for this Example 2, and Figure 7 contains the simplified LP (once obviously redundant constraints are removed).

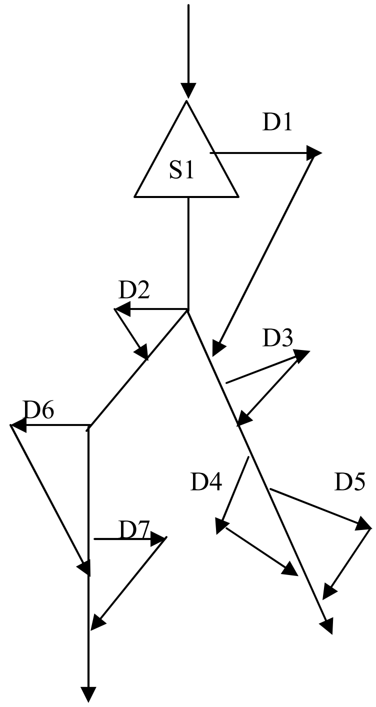

Figure 5. Network 2 Schematic - Downstream Branch.

As with Example 1, the first n (in this case 8), constraints reflect the ordinal rule, from upstream to downstream. The following n constraints relate upstream seniors to downstream juniors; four non-trivial constraints are generated. In the case of demands S1, D1, and D2, water available at either of these points of diversion is also available to downstream users on either branch. Their equations, therefore, include downstream juniors in both branches. Water available at the point of diversion of D3 would only be available to D4 and D5, as D6 and D7 divert from another network branch. Consequently, downstream constraint for D3 (C12) include only the decision variables corresponding to D4 and D5. Remaining demands (D4-D7) are not located upstream of junior demands and thus generate trivial constraints.

```
8          | number of columns (n)
100        | dimension of A matrix (rows) must be >n*n +2n
1.0        | epsilon
1.0        | baseline weight
                                   Priority  Return Loc  Return Factor   Calsim
S1 D1 D2 D3 D4 D5 D6  D7                         rl(i)       r(i)         Nodes
1  0  0  0  0  0  0   0           4           1          0.            S1
1  0  0  0  0  0  0   0           1           4          .5            D1
1  1  0  0  0  0  0   0           3           7          .5            D2
1  1  1  0  0  0  0   0           7           5          .5            D3
1  1  1  1  0  0  0   0           8           9          .5            D4
1  1  1  1  0  0  0   0           5           9          .5            D5
1  1  1  0  0  0  0   0           6           9          .5            D6
1  1  1  0  0  0  1   0           2           9          .5            D7
```

Figure 6. Input File for Example 2.

```
Minimize: X1-X8

Subject to:
C1: X4 - X5 >= 1
C2: X1 - X2 >= 1
C3: X3 - X4 >= 1
C4: X7 - X8 >= 1
C5: X8 >= 1
C6: X5 - X6 >= 1
C7: X6 - X7 >= 1
C8: X2 - X3 >= 1
C9: X4 - X5 - X6 - X7 - X8 >= 1
C10: X1 - X2 - X3 - 0.5 X5 - X6 - 0.5 X7 - 0.5 X8 >= 1
C11: X3 - X5 - 0.5 X6 - X7 - X8 >= 1
C12: X7 - 0.5 X8 >= 1
C60: X5 - 2 X7 >= 1
C61: X5 - 2 X8 >= 1
C75: X2 - 2 X3 >= 1
C79: X2 - 2 X6 >= 1
```

Figure 7. LP for Example 2.

The third set of constraints, relate downstream seniors to upstream juniors. Non-trial constraints are only formed if the demands being compared can be considered to be in the same branch, and the junior demand is located upstream of the senior demand. In this case, four constraints are formed, C60, C61, C75, and C79. Preprocessor results are presented in Table 3.

Table 3. Summary Results for Example 2

Demand	Priority	Weight
S1	4	12
D1	1	48
D2	3	13
D3	7	2
D4	8	1
D5	5	5
D6	6	3
D7	2	27

Test and Results

One test run of Example 2 is presented in this chapter. This simulation is similar to the second test run of Example 1 in that all demands are 10 taf and the monthly inflow starts at 10 taf in the first month and is increased by 10 taf in each subsequent month. Storage capacity is 15 taf. Simulated water allocation is presented in Table 3.

Table 4. Simulated Water Allocation for Example 2

Month	I1 Inflow TAF	S1 Storage TAF	D1 Delivery TAF	D2 Delivery TAF	D3 Delivery TAF	D4 Delivery TAF	D5 Delivery TAF	D6 Delivery TAF	D7 Delivery TAF
1	10	0	10	0	0	0	5	0	0
2	20	0	10	0	0	0	5	0	10
3	30	5	10	10	0	0	5	0	10
4	40	15	10	10	0	0	10	0	10
5	50	15	10	10	10	5	10	10	10
6	60	15	10	10	10	10	10	10	10
7	70	15	10	10	10	10	10	10	10
8	80	15	10	10	10	10	10	10	10
9	90	15	10	10	10	10	10	10	10
10	100	15	10	10	10	10	10	10	10
11	110	15	10	10	10	10	10	10	10
12	120	15	10	10	10	10	10	10	10
Priority		**4**	**1**	**3**	**7**	**8**	**5**	**6**	**2**
Weight		**12**	**48**	**13**	**2**	**1**	**5**	**3**	**27**

In the first month of simulation, the 10 taf available is diverted by D1, the demand with the highest priority. D5 diverts the return flow from D1, as it is the highest priority downstream of D1 return flow location. In the second month, users with the two highest priorities divert their full demand, while, as in the first month, D5 diverts D1's return flow. As the flow increases to 30 taf in the third month, the third highest priority, D2, is also allocated water. Because the return flow of D2 can be used to meet 5 of the 10 taf demand of D7, S1 is able to store 5 taf. In the fourth month, the fourth in priority, S1 is able to reach capacity by storing 10 taf. As in the previous month, D1, D2, and D7 are allocated 10 taf. D5 is also allocated 10 taf, 5 of which from D1 return flow and the remaining 5 taf from the inflow to the system. In the remaining months, S1 is at capacity and therefore no longer being allocated water. Of the 50 taf inflow in the fifth month, all users except D4 and S1 are allocated their full demand while D8 is allocated 5 taf. In month six, all demands are met. The results from this simulation show that the weights generated by the preprocessor do allocate water according to the original priorities.

Branching Network with Upstream Tributary

Description

A third network, Network 3, was created to test the weight generator procedure. As depicted in Figure 8, Network 3 contains both upstream tributaries and downstream branching. The inflow to the system occurs at each upstream reservoir, S1 and S4. The preprocessor input file for this example is listed in Figure 9.

The configuration of Network 3 requires an additional node type. Unlike the previous example, no diversions are located at the confluence node. However, the return flow from D3 is located at the confluence. Therefore, to ensure that the return from D3 is available to both downstream branches, a node needs to be specified for the confluence. Node J8 represents the confluence of the branches.

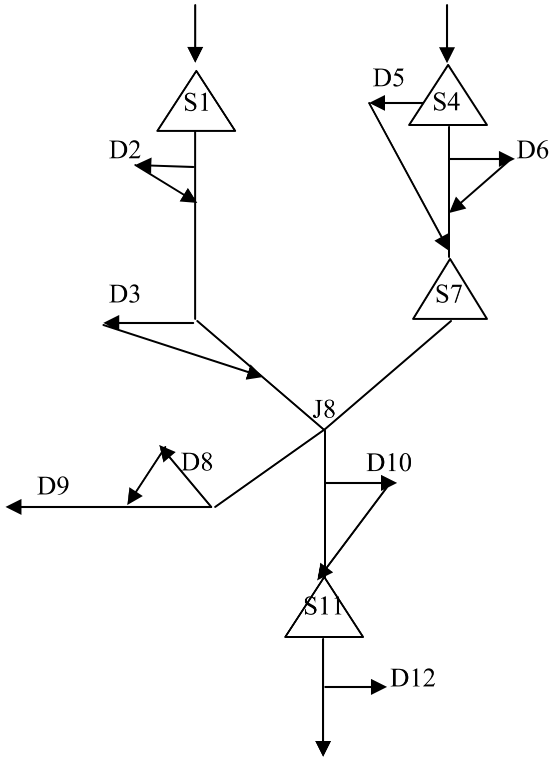

Figure 8. Network 3 Schematic - Branching with Upstream Tributary.

```
13            | number of columns (n)
500           | dimension of A matrix (rows) must be >n*n +2n
1.0           | epsilon
1.0           | baseline weight
                                          Priority  Return Loc Ret Fac  Calsim
S1 D2 D3 S4 D5 D6 S7 J8 D8 D9 D10 S11 D12             rl(i)    r(i)     Nodes
1  0  0  0  0  0  0  0  0  0   0   0   0      9          1        0       S1
1  0  0  0  0  0  0  0  0  0   0   0   0      7          3       .5       D2
1  1  0  0  0  0  0  0  0  0   0   0   0      6          8       .5       D3
0  0  0  1  0  0  0  0  0  0   0   0   0      1          4        0       S4
0  0  0  1  0  0  0  0  0  0   0   0   0      5          7       .5       D5
0  0  0  1  1  0  0  0  0  0   0   0   0      8          7       .5       D6
0  0  0  1  1  1  1  0  0  0   0   0   0      4          7        0       S7
1  1  1  1  1  1  1  0  0  0   0   0   0      0          8        0       J8
1  1  1  1  1  1  1  1  0  0   0   0   0      2         10       .5       D8
1  1  1  1  1  1  1  1  1  0   0   0   0     10         10        0       D9
1  1  1  1  1  1  1  1  0  0   0   0   0      3         12       .5       D10
1  1  1  1  1  1  1  1  0  0   1   1   0     11         12        0       S11
1  1  1  1  1  1  1  1  0  0   1   1   0     12         14       .5       D12
```

Figure 9. Input File for Example 3.

Node J8 is is given zero priority so constraints are not generated for J8. J8 is only identified to preserve the connectivity of the network and allow water to be routed correctly. Appendix A-4 contains preprocessor LP solver output and Figure 10 contains the LP once trivial constraints are removed.

```
Minimize: X1-X12
Subject to:
C1: X9 - X10 >= 1
C2: X7 - X8 >= 1
C3: X6 - X7 >= 1
C4: X1 - X2 >= 1
C5: X5 - X6 >= 1
C6: X8 - X9 >= 1
C7: X4 - X5 >= 1
C8: X2 - X3 >= 1
C9: X10 - X11 >= 1
C10: X3 - X4 >= 1
C11: X11 - X12 >= 1
C12: X12 >= 1
C13: X9 - X10 - X11 - X12 >= 1
C14: X7 - 0.5 X10 - 0.5 X11 - 0.5 X12 >= 1
C15: X6 - 0.5 X10 - 0.5 X11 - 0.5 X12 >= 1
C16: X1 - X2 - X3 - X4 - X5 - X8 - X10 - X11 - X12 >= 1
C17: X5 - X8 - 0.5 X10 - 0.5 X11 - 0.5 X12 >= 1
C18: X8 - 0.5 X10 - 0.5 X11 - 0.5 X12 >= 1
C19: X4 - X10 - X11 - X12 >= 1
C20: X2 - 0.5 X10 >= 1
C22: X3 - 0.5 X11 - 0.5 X12 >= 1
C23: X11 - X12 >= 1
C50: X6 - 2 X7 >= 1
C101: X4 - 2 X5 >= 1
C102: X4 - 2 X8 >= 1
C110: X2 - 2 X7 >= 1
C111: X2 - 2 X6 >= 1
C113: X2 - 2 X5 >= 1
C114: X2 - 2 X8 >= 1
C134: X3 - 2 X7 >= 1
C135: X3 - 2 X6 >= 1
C137: X3 - 2 X5 >= 1
C138: X3 - 2 X8 >= 1
```

Figure 10. LP for Example 3.

As with examples 1 and 2, the first n (in this case 13), constraints reflect the ordinal rule, from upstream to downstream. The second set of n constraints relates an upstream senior to downstream juniors and eleven non-trivial constraints are generated. For the two branches upstream of the confluence J8, water that is available in one branch is not available at the other. For instance, water available at S1, D1, and D2 is not available to S4, D5, D6, and S7, but is available to the other users on either downstream branch. Their equations, therefore, include downstream juniors in both branches. A similar logic can be applied to the tributary branch on the right. The constraints C14 to C20 reflect this logic. Constraint 22 reflects that only node D9 is downstream of D8. D10, S11, and D12 are on a separate branch, as reflected in and constraints 23 to 26.

The third set of constraints, relate downstream seniors to upstream juniors. Non-trivial constraints are only formed if the demands being compared are in competition for the same unit of water. This happens if there is a direct path between the two demands and the junior demand is upstream of the senior demand. In this case, 15 non-trivial constraints were formed. Preprocessor results are shown in Table 5.

Test and Results

Test run results for Example 3 are shown in Table 6. For this simulation all demands are 10 taf, storage capacity in all reservoirs is 50 taf, and the monthly inflow at both upstream reservoirs, S1 and S4, starts at 10 taf in the first month and is increased by 10 taf in each subsequent month. Simulated water allocation for Example 3 is presented in Table 6.

In the first month of simulation, demands with the highest two priorities are allocated water. While the 10 taf inflow into S1 is allocated to D8 (D9 diverting D8 return flow), S4 stores its inflow. In the second month the 20 taf inflow into S1 are allocated to D3 and D10, with D9 and S11 being allocated the return flows. On the right side branch, S4 stores all the inflow. In month three, D2, D3, and D8 are each allocated 10 taf and return flows go to D9 and D10. S11 diverts the return flow from D10. On the right hand branch, S4 reaches capacity by storing 20 of the 30 taf inflow; the remaining 10 taf being stored by S7**.**

Table 5. Summary Results for Example 3

Demand	Priority	Weight
S1	10	7
D2	8	9
D3	7	19
S4	2	161
D5	6	20
D6	9	8
S7	5	41
D8	3	43
D9	11	3
D10	4	42
S11	12	2
D12	13	1

Table 6. Simulated Water Allocation for Example 3

Month	I1 & I4 Inflow TAF	S1 Storage TAF	D2 Delivery TAF	D3 Delivery TAF	S4 Storage TAF	D5 Delivery TAF	D6 Delivery TAF
1	10	0	0	0	10	0	0
2	20	0	0	0	30	0	0
3	30	0	10	10	50	0	0
4	40	10	10	10	50	10	10
5	50	45	10	10	50	10	10
6	60	50	10	10	50	10	10
Priority		**9**	**7**	**6**	**1**	**5**	**8**
Weight		**7**	**9**	**19**	**161**	**20**	**8**

Month	S7 Delivery TAF	D8 Delivery TAF	D9 Delivery TAF	D10 Delivery TAF	S11 Storage TAF	D12 Delivery TAF
1	0	10	5	0	0	0
2	0	10	5	10	5	0
3	10	10	5	10	10	0
4	50	10	5	10	15	0
5	50	10	10	10	40	0
6	50	10	10	10	50	10
Priority	**4**	**2**	**10**	**3**	**11**	**12**
Weight	**41**	**43**	**3**	**42**	**2**	**1**

In the fourth month S1 is allocated 10 of the 40 taf, and the remaining 30 taf going, as in the previous month, to D2, D3, and D8. Return flows from these diversions are, as before, allocated to D9 and D10, with S11 diverting return flow from D10. S4 filled in the third month, so water is available to be allocated among the other demands on the right side branch. D5 and D6 are each allocated 10 taf and the remaining 30 taf plus the returns from D5 and D6 are stored in S7.

In the fifth month, both reservoirs on the right side branch have reached capacity. Inflow to S4 is allocated to D5 and D6, and 40 taf enters the node J8 from the right side branch. Of the 50 taf inflow into S1, 15 taf are released from the reservoir to meet the demands D2 and D3, the latter being partially met by the returns from D2. S1 is now able to store 35 taf as the senior demands in the downstream branches are being met by water available in the right side branch. Of the 45 taf entering the junction node J8, 15 goes to meet demands D8 and D9, and 30 taf enters the branch above D10. D10 diverts 10 taf and returns 5 taf to the river. S11 stores the full 25 taf and D12 is not allocated any water.

In the sixth month all reservoirs reach capacity and all flow demands are met. Of the 120 taf that enters the system, 40 taf is consumed by the eight flow diversions, 15 taf is allocated to storage (S1 and S11) and 65 taf leaves the system.

Looped Networks

<u>Description</u>

Interesting situations arise when looped networks are considered. Network 4 (Figure 11) was used to investigate the computation of priority preserving unit costs coefficients for looped networks (example 4). The preprocessor input file for example 5 appears in Figure 12.

Network 4 is a simple single stem network, not unlike the first network examined in Example 1. The main difference is that in Network 4 water can return to an upstream node, creating a looped network. The existence of a looped network is identified by the return location of D4 being upstream of D4 itself.

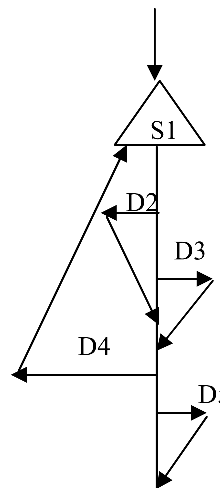

Figure 11. Network 4 Schematic.

```
5              | number of columns (n)
500            | dimension of A matrix (rows)  must be >n*n +2n
1.0            | epsilon
1.0            | baseline weight
                   Priority    Return Loc      Return Factor   Calsim
S1 D2 D3 D4 D5                   rl(i)            r(i)          Nodes
1  0  0  0  0        2           1                0              S1
1  0  0  0  0        3           4               .5              D2
1  1  0  0  0        1           4               .5              D3
1  1  1  0  0        5           1               .5              D4
1  1  1  1  0        4           6               .5              D5
```

Figure 12. Input file for Example 5.

For looped networks, all demands within the loop must be considered both upstream and downstream of each other, since a portion of the water diverted at D4 will return to S1 and thus be considered upstream of S1, D2, and D3. To create constraints that correctly represent the relative location of nodes within a loop in a looped network, the location connectivity matrix must therefore be modified. To avoid input entry errors, this new matrix is created, in run time, by the FORTRAN preprocessor when it encounters rl(i)=k<i; which prompts the program to save i as the end node of the loop and rl(i) as the

beginning node of the loop. The location connectivity matrix thus becomes

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

, which reflects the pump back loop between S1 and D4.

Recall from Chapter 2 that ones in column j of the location connectivity matrix show the priorities downstream of j, and ones in row i show the priorities upstream of i. In this new location connectivity matrix, column 1 shows that water available to S1 can stay at S1 a(1,1)=1 or go to any other demand downstream. Row 1, on the other hand, shows which priorities are upstream of S1. Because of the loop, all priorities except for D5 can be considered to be upstream of S1. The second column shows that all priorities except for itself (a(2,2)=0) can be considered downstream of D2. The second row of this matrix shows that S1, D3, and D4 are considered to be upstream of D2.

Appendix A-5 contains preprocessor output and Figure 15 contains the simplified LP once trivial constraints are removed.

```
Minimize: X1 - X5

Constraints
C1: X2 - X3 >= 1
C2: X3 - X4 >= 1
C3: X1 - X2 >= 1
C4: X5 >= 1
C5: X4 - X5 >= 1
C6: X2 - X3 - X4 - X5 >= 1
C7: X3 - 0.5 X4 - 0.5 X5 >= 1
C8: X1 - X2 - 0.5 X3 - 0.5 X4 - 0.5 X5 >=1
C12: X2 - 2 X3 >= 1
C14: X2 - 2 X5 >= 1
C19: X3 - 2 X5 >= 1
C22: X1 - 2 X3 >= 1
C24: X1 - 2 X5 >= 1
C34: X4 - 2 X5 >= 1
```

Figure 13. LP for Example 5.

The first five constraints listed in Figure 13 reflect the ordinal rule. Constraint C6 and C7 reflects the downstream rule for S1 and D2. Constraint C8 is the downstream rule for D3, which has priority 1. Because D3 is within the loop, X2 and X3, representing S1 and D2, are considered to be downstream of D3, and therefore included in constraint C8.

Constraints C12, C14, C19, and C24 also show the modified special relationship between the nodes, as they consider nodes that are in reality downstream to be upstream as a result of the loop in the network.

<u>Test and Results</u>

Preprocessor computed weights and water allocation are shown in Table 10. Simulation results are shown for a single month. As expected, the simulated inflow of 10 taf is diverted by D3, the highest priority demand. An interesting allocation occurs with the return flow from D3. Because the next highest priority is S1, and S1 is upstream of D3, water is diverted at D4, the lowest priority in this system, so that its return flow can be allocated to S1. This is an interesting paradox, where water is allocated to a lower priority, D4, so that it can be used for a higher priority, S1. This result, however, may not be implemented in a system governed by priorities. Depending on the actual plumbing in the system, it is very likely that D4 would not be able to consume any water, with the entire return flow of D3 being routed through D4 and allocated to S1.

Table 7. Example 5 Results

Demand	Priority	DV	Weight	Water Allocation
S1	2	X_2	9	2.5
D2	3	X_3	4	0
D3	1	X_1	14	10
D4	5	X_5	1	5
D5	4	X_4	3	0

SUMMARY

In this Chapter, the implementation of the generalized algorithm was presented. Results obtained are consistent with what would be expected, where the weights generated, indeed preserve the priorities and allocate water accordingly. Simple mainstem, branched, tributaries and looped systems were examined. With looped networks, lower priority juniors can be allocated water before higher priorities to ensure that a still higher priority is satisfied.

In the next chapter the generalized algorithm is used to generate weights for an LP driven simulation. In Chapter 5 the common issue of equal priorities is examined and two approaches are presented on how to compute priority preserving unit cost coefficients for equal priority demands.

CHAPTER 4: CALSIM APPLICATION

INTRODUCTION

Two-River System Calsim model was used to test the weight generator for an LP driven model. This model, provided by the California Department of Water Resources (DWR) staff, is a simplified Calsim II network, consisting of the Sacramento-San Joaquin Delta, two north-of-Delta storage facilities, Shasta (Central Valley Project, CVP) and Oroville (State Water Project, SWP), and two south-of-Delta storage facilities, CVP and SWP San Luis Reservoir. A schematic of the Two-River Model network appears in Figure 14.

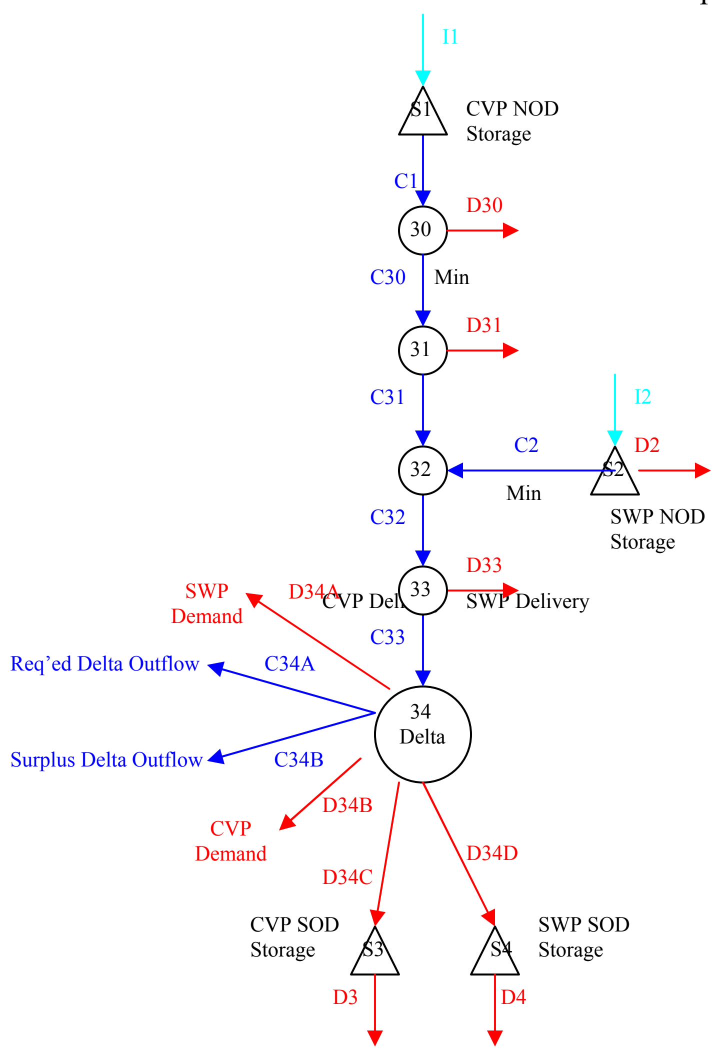

Figure 14. Two-River System Model Network.

Unlike the examples presented in Chapter 3, the Two-River System model is a LP-driven simulation model, including several non-NFP constraints. Many non-NFP constraints are associated with Delta operations, from which the most interesting insights regarding this problem emerge.

Among the non-NFP constraints included in the Two-River System model are: (i) the export-inflow ratio restrictions which limit exports from the Delta (*D34C* and *D34D*) to be no greater than a given fraction of the inflow to the Delta (*C33*), (ii) soft constraints defining a minimum desired pumping at each of the export pumps, and (iii) the Coordinated Operations Agreement (COA) between the CVP and SWP. The COA apportions, between the CVP and SWP, the rights to export water in the Delta and also the responsibility to protect other beneficial uses of water in the Delta and the Sacramento Valley.

As a test case, the Two-River System model has led to several insights into the applicability of the weight generating method proposed in the previous chapters to more general LP models. The COA equations and variables, in particular, have led to the understanding of two important aspects of this problem, namely the computation of negative weights and additional qualities of priority preserving sets needed when the simulation is driven by a LP rather than a NFP formulation. These insights are presented in this chapter and discussed more fully in chapters 6 and 7.

PROCEDURE

To understand the interaction between general LP constraints and weights, four runs of the Two-River model were performed. Two of the four runs (DWR and DWR-zero) use the original weights provided by DWR, and the two other runs (UCD and UCD-zero) use weights obtained with the weight generator. The difference between each pair of runs is that both "DWR" and "UCD" include both positive and negative weights. "DWR-zero" and "UCD-zero", on the other hand, include only the positive weights. Table 1 presents the priorities and weights associated with weighted decision variables for the four runs.

Close inspection of the priorities listed in Table 1 and the DWR weights revealed inconsistencies between priorities and weights provided by DWR. While the minimum instream flow demands *C2_MIF* and *C30_MIF* are listed as having the same priority (2) as the diversion demands *D2, D30, D31, D33, D34A, D34B*, and *C34A*, the weights for the instream demands exceed those for the diversion demands listed above. This larger weight will always result in water allocation to the instream demands before diversion demands. To be consistent with the DWR weights, in the computation of weights (UCD) it was assumed that the minimum instream flow requirements have priority 2 while the diversions demands listed above have priority 3.

Also, *D3* and *D4* (south of Delta demands) are listed as having the same priority as the San Luis storage pools *S3_2, S3_3, S4_2*, and *S4_3*, but have higher weight. Given the relative location of these facilities, the higher weight associated with the demands will guarantee that water is allocated to those south of Delta demands rather than be placed in storage. To be consistent with the DWR weights, we assumed, therefore, that *D3* and *D4* have higher priority (priority 4) than the second and third San Luis storage pools (priority 5).

Table 8. Priorities and Weights

Priority	Variable	True Priority	DWR	DWR-Zero	UCD	UCD-Zero
1	S1_1	1	20000	20000	21140	21140
1	S2_1	1	20000	20000	18540	18540
1	S3_1	1	20000	20000	2570	2570
1	S4_1	1	20000	20000	2570	2570
2	C2_MIF	2	5500	5500	2560	2560
2	C30_MIF	2	5500	5500	2560	2560
2	D2	3	5000	5000	2550	2550
2	D30	3	5000	5000	2550	2550
2	D31	3	5000	5000	2550	2550
2	D33	3	5000	5000	2550	2550
2	D34A	3	5000	5000	2550	2550
2	D34B	3	5000	5000	2550	2550
2	C34A	3	5000	5000	2550	2550
3	D3	4	1265	1265	420	420
3	D4	4	1265	1265	420	420
3	S3_2	5	1235	1235	410	410
3	S4_2	5	1235	1235	410	410
3	S3_3	5	1225	1225	410	410
3	S4_3	5	1225	1225	410	410
4	S1_2	6	93	93	400	400
5	S2_2	7	92	92	390	390
6	S1_3	8	88	88	200	200
7	S2_3	9	87	87	190	190
8	S1_4	10	84	84	100	100
9	S2_4	11	80	80	90	90
10	S3_4	12	65	65	40	40
11	S1_5	13	62	62	30	30
12	S4_4	14	60	60	20	20
13	S2_5	15	56	56	10	10
14	UNUSED_FS	?	-1285	0	-450	0
14	UNUSED_SS	?	-1285	0	-450	0
15	C34B_CVP	3 (-)	-2000	0	-550	0
15	C34B_SWP	3 (-)	-2000	0	-550	0
16	S1_6	2 (-)	-10000	0	-3300	0
16	S2_6	2 (-)	-10000	0	-2100	0
16	S3_5	2 (-)	-10000	0	-650	0
16	S4_5	2 (-)	-10000	0	-650	0
17	F1	1 (-)	-100000	0	-3400	0
17	F2	1 (-)	-100000	0	-3400	0
17	F3	1 (-)	-100000	0	-3400	0
17	F4	1 (-)	-100000	0	-3400	0

RESULTS

Plots of simulated storages under the four weighting schemes show some interesting results. For the CVP north-of-Delta storage (Figure 15), DWR and UCD runs match fairly well in most years. The UCD-zero time series, albeit lower in some periods, also tracks fairly well. The DWR-zero run, on the other hand, simulated consistently lower storage than the other three runs.

A similar outcome is present in the SWP north of the Delta storage plot (Figure 16), where the DWR-zero run has considerably lower storage than the other three runs. The lower Oroville storage in the DWR-zero and UCD-zero runs is caused by the weights associated with the surplus Delta outflow (*C34B_CVP* and *C34B_SWP*) being set to zero. The inflow/export limitation on pumping from the Delta limits exports to a percentage of inflow to the Delta. If the variables representing surplus Delta outflow are not negatively weighted, the system will loose water to surplus Delta in an attempt to move water to the south-of-Delta storage facilities. In the DWR and UCD runs, the negative weights discourages water allocation to those variables. In the DWR-zero run, however, the north of Delta storages are more severely affected than the in the UCD-zero run because, in the DWR runs, the weights on storage and deliveries south of the Delta are considerably greater than weights on storage north of the Delta.

It is interesting to compare the sharp reductions in Oroville storage in the DWR-zero run with the spikes in SWP portion of surplus Delta outflow (Figure 17) and spikes in SWP south-of-Delta storage (Figure 18). A sharp reduction in Oroville storage is usually accompanied by a large increase in surplus Delta outflow and increase in the State San Luis storage, as seen in 1922, 1927, 1939, 1944, 1959, 1965, and 1987. The DWR-zero run attempts to fill the state portion of San Luis at the expense of Oroville storage comes with considerable reduction in surplus Delta outflows.

In both the UCD and DWR runs the weights for south of Delta demands exceed the weights of all Oroville pools except the dead pool (*S2_1*). However, because the UCD weights for storage north-of-Delta and demands south-of-Delta are more evenly scaled, in the UCD-zero run more water is kept in Oroville than in the DWR-zero run. For the DWR set of weights, the negative weights associated with *C34B_SWP* and *UNUSED_SS* are needed to keep water in Oroville. The negative weights in the DWR run compensate for the uneven scaling of positive weights, and once the negative weights are removed the positive weights alone perform poorly. A similar, albeit less striking, pattern can be observed in Federal system (figures 15, 19, and 20).

DISCUSSION

To eliminate the effects non-NFP constraints have on the allocation of water, we tested both DWR and UCD weight sets in a NFP driven model. All non-NFP constraints and variables were removed from the Two-River System model, leaving only continuity and capacity constraints. Most Delta operations constraints were eliminated, including the COA and export-inflow restrictions, and the non-NFP variables *UNUSED_SS* and *UNUSED_FS*. *C34B* (surplus Delta outflow) was assigned the weight of *C34B_CVP* (or *C34B_SWP*). The test was performed much like those presented in Chapter 3, in which the inflow into the system increased at a constant rate every month.

NFP-Driven Simulations

Both the UCD and the DWR weight sets resulted in allocations that preserved priorities. The simulated results were practically identical for the four NFP runs, differing only in the allocation to equal priority demands (multiple optima). These results indicate that all weight sets are priority preserving for NFP driven simulation.

LP-Driven Simulations

Once it was determined that all weight sets were priority preserving for an NFP simulation, we sought to understand why the DWR-zero LP simulation resulted in such different allocations from the other three runs.

The difference seemed to result from how the Delta was being operated in the four simulations. Closer scrutiny of Delta operations constraints and the weights assigned to decision variables associated with those constraints led to interesting insights into the role of non-NFP constraints and priority preserving weight sets in LP driven simulations.

Without negative weights, the DWR-zero run resulted in water being sent to higher priority demands south of the Delta. However, in the other three runs, the water was allocated to lower priority storage demands north of the Delta. Initially, therefore, it appeared that the DWR-zero run was being priority preserving while the other runs were not. By considering the LP constraints and the weights of the simulations presented above, we can investigate the effects of non-NFP constraints in priority driven simulations.

COA

The COA constraints are used to apportion, between the CVP and SWP, both the responsibilities for in-basin-use (*IBU*) of water that requires storage withdrawals (25% to SWP and 75% to CVP) and the right to export excess water in the system, unstored-water-for-export (*UWFE*, 45% to SWP and 55% to CVP). The COA is therefore a balance of water ownership within the Delta.

Consider the COA constraints:

$$D34A - SWPDS + D34D\_EXP1 + C34B\_SWP + UNUSED\_SS + 0.25\ IBU - 0.45\ UWFE = 0 \quad (44)$$

$$D34B - SWPDS + D34C\_EXP1 + C34B\_CVP + UNUSED\_FS + 0.75\ IBU - 0.55\ UWFE = 0 \quad (45)$$

and the constraint:

$$D34C\_EXP2 - UNUSED\_SS <= 0 \quad (46)$$

$$D34D\_EXP2 - UNUSED\_FS <= 0 \quad (47)$$

where:

D34A = SWP demand in Delta
D34B = CVP demand in Delta
C34B_SWP = SWP portion of surplus Delta outflow
C34B_CVP = CVP portion of surplus Delta outflow
UNUSED_SS = Unused State share of Delta surplus
UNUSED_FS = Unused Federal share of Delta surplus
D34C_EXP2 = CVP export of *UNUSED_SS*
D34D_EXP1 = SWP export
D34D_EXP2 = SWP export of *UNUSED_SF*

D34C_EXP1 = CVP export
IBU = Total In-Basin-Uses met with storage withdrawals
UWFE = Total Unstored-Water-For-Export
SWPDS = SWP change in storage (*SWPDS*=C2 + D2 – I2).
CVPDS = SWP change in storage (*CVPDS*=C1 – I1).

Referring to equation (44) above[1], the decision variables *D34A, C34B_SWP, UNUSED_SS, D34D_EXP1*, and *IBU* have positive coefficients in the COA constraint. They are balanced by the variables with negative coefficients, *SWPDS* and *UWFS*, representing water available in the Delta.

Under the COA, each project is allowed to pump the other project's unused share of water in system (equations 46 and 47). The negative weight associated with *UNUSED_SS* is used to ensure that the State pumps as much as it can under its COA allowance and only when it cannot, due to physical constraints, is the CVP entitled to pump the unused State share. Without this negative weight, the LP may allocate the State share of water to Federal demands south of the Delta if it results in a higher objective function value. In this case, the allocation of water would be to a higher priority, but the simulation would not be accurate in terms of rights and operating agreements, as the State would not be pumping up to its allowance and SWP water (under the COA) would be allocated to the CVP.

Indeed, the allocation of one project's water to the other project occurs frequently in DWR-zero and UCD-zero runs, particularly when the State and Federal portions of San Luis Reservoir are not in balance (e.g., one reservoir has water in the fourth pool while the other only has water up to the second pool). The negative weights on *UNUSED_SS* and *UNUSED_FS* avoid one project taking the other project's water, in what would otherwise be a purely priority-driven allocation, rather than one driven by the operational rules of constraints imposed the system. This is an example in which a constraint supersedes a purely priority-driven allocation to water demands.

In the DWR run, the weight associated with *UNUSED_SS* (-1285) is just slightly greater in magnitude than the DWR weights assigned to priorities south of the Delta (1265 for diversion and 1235 for storage). In effect, the negative weight associated with *UNUSED_SS* is just high enough to counterbalance the weights on south-of-Delta demands so that allocation of water to *UNUSED_SS* is avoided.

Export-Inflow Limit
The allocation of water within the system is also influenced by the "Export-Inflow" limit. The Export-Inflow limit constrains the total export from the Delta (*D34C* + *D34D*) to a fraction (*E/I ratio*) of the inflow to the Delta (*C33*). The *E/I ratio* is set to 0.35 in March through June, 0.65 in July through January, and variable (between 0.35 and 0.45) in February.

Consider the DWR and UCD weight sets presented in Table 1. The DWR weights associated with south of Delta demands (*D3* and *D4*) and second and third storage pools at San Luis Reservoir (*S3_2, S3_3, S4_2*, and *S4_3*) are one order of magnitude greater

[1] Although the discussion refers to SWP, a similar argument can be made for the CVP (equations 45 and 47).

than those of all priorities except dead storage pools at Shasta (*S1*) and Oroville (*S2*) and senior demands north-of-Delta. When there is no negative weight on *C34B_SWP* and *C34B_CVP*, the DWR weighting scheme favors allocation to south of Delta users at the expense of north of Delta storage and because of the inflow-export restrictions, at a great increase of surplus Delta outflow. The UCD-zero run behaves more like the DWR and UCD runs because the weights on the second, third, and fourth pools in Shasta and Oroville are smaller than, but in the same order of magnitude as those assigned to the south of the Delta priorities.

The UCD weights for north-of-Delta storage and south-of-Delta demands are closer in magnitude because the algorithm used to calculate them computed weights that are just high enough to preserve priorities. The DWR weights, on the other hand, appear to have been derived using the common practice of assigning weights of different orders of magnitude to ensure that a senior weight is high enough to guarantee appropriate allocation of water. Although this method appears to perform fairly well when all weights are in place, when a subset of the weights is removed, the results can become skewed.

When weights are obtained by trial and error, it is possible to create a priority preserving weight set with subsets that do not perform well. While this should not be a problem if the model is not modified, models often are used to analyze scenarios that may exclude or alter some of the physical, institutional, or regulatory components or priorities that were initially included in the model, and for which the original weight set was developed. Furthermore, complex models of water resources systems often are used by people that are not involved in the development of the model, are less knowledgeable about the inner workings of the model, and thus less likely to be able to understand and accommodate such changes. It is important, therefore, that subsets of the weight set be priority preserving, something that is more readily achieved if the objective function weights are computed using a tractable algorithm.

For a NFP driven simulation, the scaling of weights is less important. To compute weights for a NFP driven simulation, the value of ε can vary from equation to equation (Chapter 2, equations 31 to 35), and the resulting weight set is still priority preserving. This is so because the only constraints in a NFP are continuity and capacity constraints and all the variables in the objective function represent water allocated to a network arc. However, for an LP driven simulation, with more complex constraints and non-arc flow variables included in the objective function, the scaling of weights becomes important.

In simulation models with embedded optimization, as is the case with NFP and LP driven simulations, the objective function does not have a tangible meaning as when an optimization model is used to maximize or minimize a known quantity such as profit, economic value, total water deliveries, loss, time, etc. When the objective function has a tangible meaning, unit consistency is ensured and the relative magnitude, or appropriate scaling, of the weights is automatically ensured.

While not optimizing a tangible quantity, in a NFP driven simulation all decision variables have the same physical meaning of water flow in an arc and the constraints are simple mass balance constraints on all nodes. A LP driven simulation, on the other hand, often includes variables that mean something other than arc flow. For instance, some non-NFP variables provide compliance with regulatory or operating rule as is the case with *UNUSED_SS* and *UNUSED_FS*. Others variables are included in the NFP to provide a degree of compliance with some operating criteria or rule. These non-arc flow variables may be assigned weights and thus be included in the objective function. For a LP driven simulation, therefore, it is more important that the weight set not be out of scale.

In the example presented in this chapter, removal of some of the weights (negative weights) from the DWR weight set resulted in a subset of positive weights that, due to its uneven scaling, resulted in much lower model performance than the more evenly scaled UCD weight set. The weights given to demands south of the Delta in the DWR weight set are one order of magnitude greater than the weights given to the storage pools north of the Delta. Given the inflow-export limits and the absence of negative weights on Delta surplus, available water is always sent to the SOD storage despite its great "cost" in increased surplus outflow. This spilling of water from the system resulted in a sharp decline in model performance. If the weights are properly scaled, the exclusion or addition of one or more model components is less likely to significantly affect overall model performance.

CONCLUSIONS

The purpose of this chapter is to present the application of the weight generator to a LP driven model and the insights gained from the analysis. To test the weight generator applicability to LP driven models, a simplified model of the CVP/SWP system, the Two-River System model was used. Model results were compared between runs using the DWR assigned weights and runs using weights developed with the method presented in chapters 2 and 3. The following are insights drawn from the analysis presented in this chapter.

Negative Weights

For practical purposes, the main insight gained from the work presented in this chapter is how to set negative weights. Negative weights associated with excess water in the system and coupled with priorities of where not to deliver water can be computed using the same method as for positive weights.

However, two other types of negative weights emerged in the Two-River System model. The negative weights on slack-like variables designed to adjust the "degree of hardness" of a constraint or a set of constraints (generally associated with a non-arc flow variable), must be chosen so the non-NFP constraints work as intended. Examples of this type of negative weights are the soft constraint on minimum desired pumping at Banks and Tracy pumping plants and the negative weights associated with Delta surplus outflow variables (*C34B_CVP* and *C34B_SWP*). These weights are calibration parameters rather than weights derived from a water allocation priority system. The choice of these "calibration" coefficients must be done individually and once the prioritized weights have been computed.

The variables *UNUSED_SS* and *UNUSED_FS* are slightly different in that their coefficients must balance the weights on demands south of the Delta to ensure accurate implementation of the COA in the simulation. In this case, one is not adjusting the degree of hardness of the constraint, but rather ensuring the constraint reflects the legal and administrative rules governing the operations of the system. Computation of negative weights is more fully discussed in Chapter 6.

Subsets of priority preserving weight sets and scaling of weights

Ideally, one would like a set of weights in which the removal of one of more weights does not greatly affect the performance the remaining weight set. The examples in this chapter show that, for LP driven models, if weights are not properly scaled, the model performance can be sharply reduced once one or more weights are removed from the model. This was the case of the DWR-zero run, in which disproportionally low weights given to storage north of the Delta resulted in water being allocated south of the Delta at great “cost” in surplus outflow.

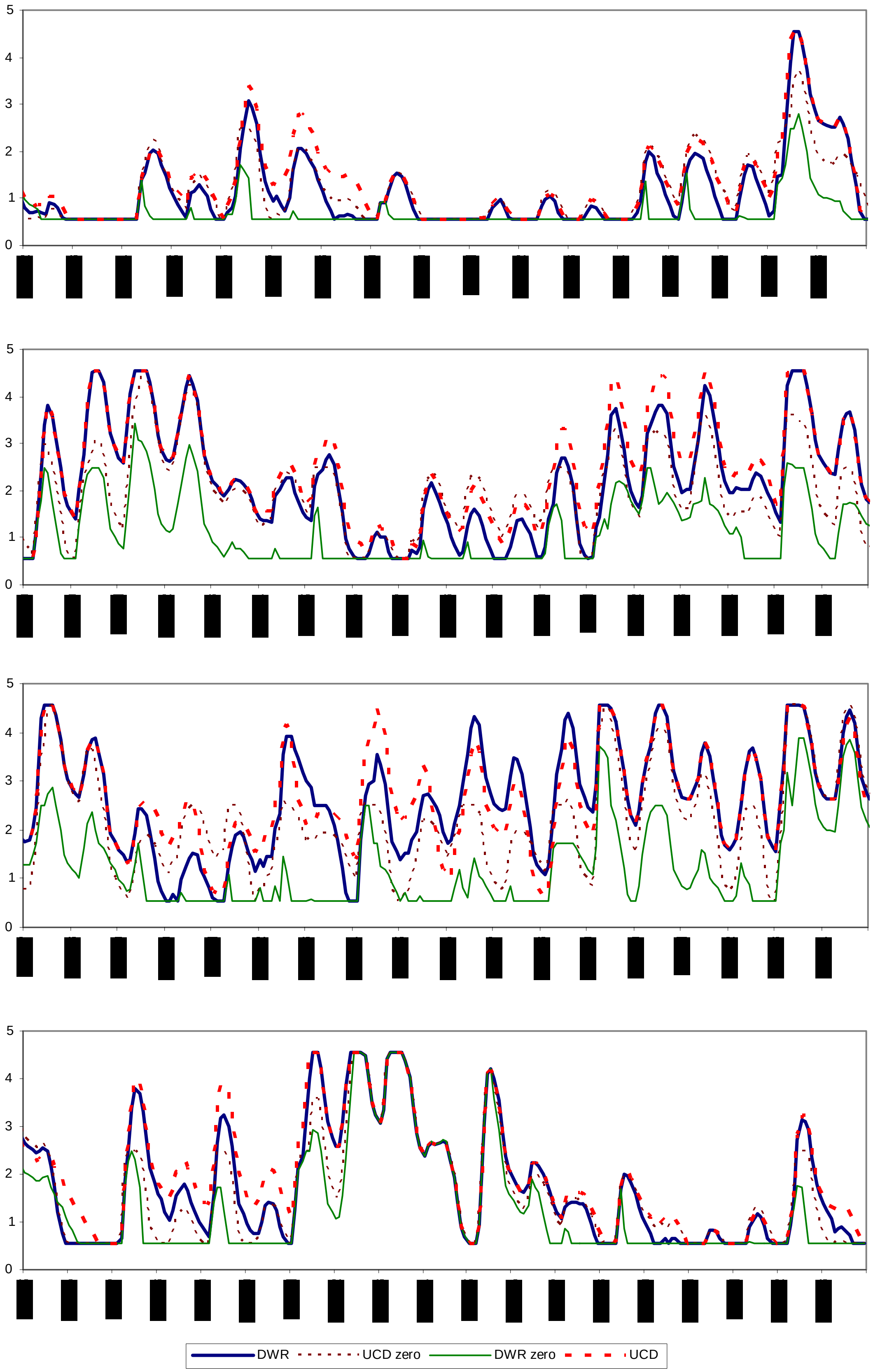

Figure 15. Shasta Storage (MAF).

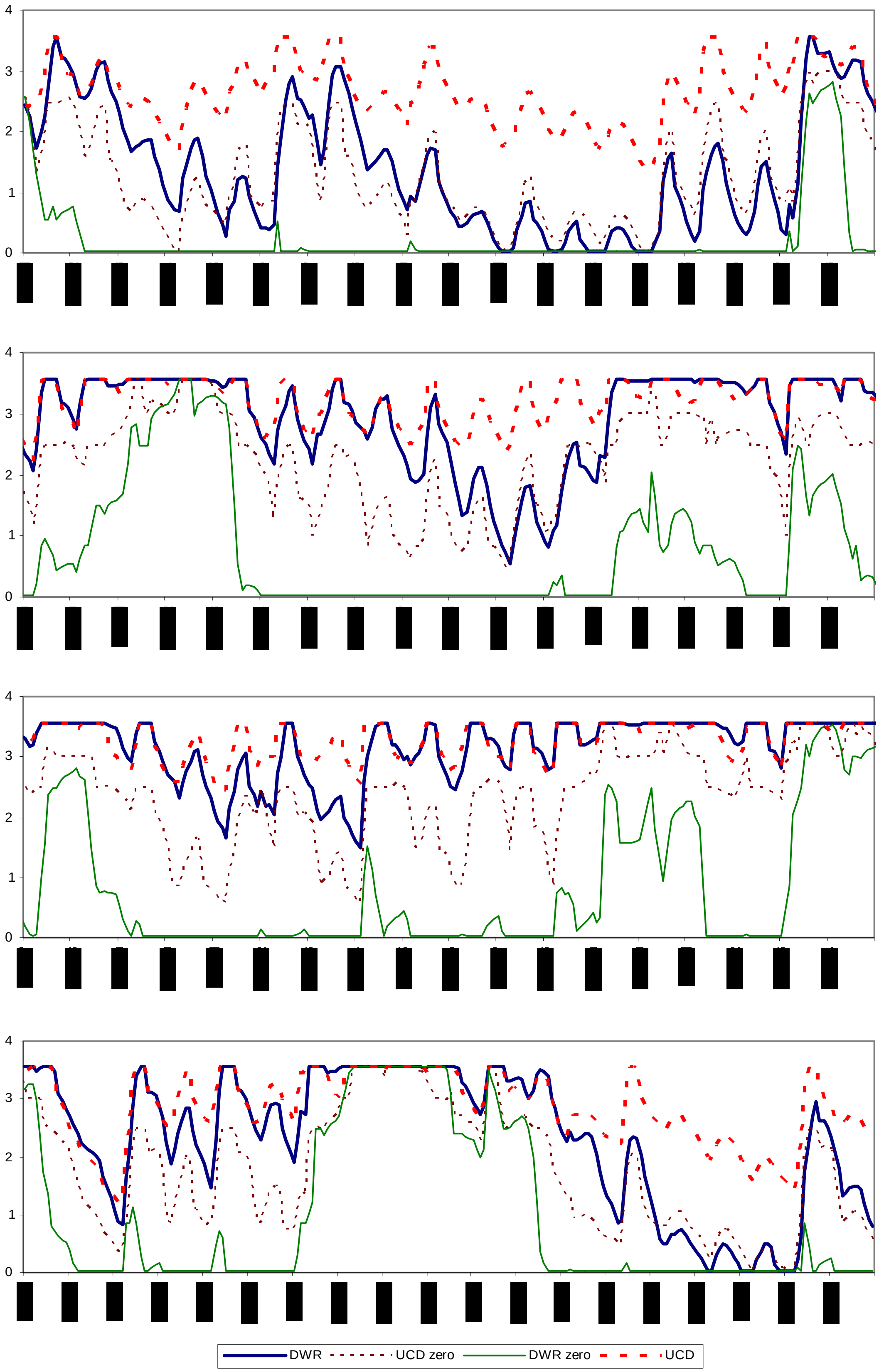

Figure 16. Oroville Storage (MAF).

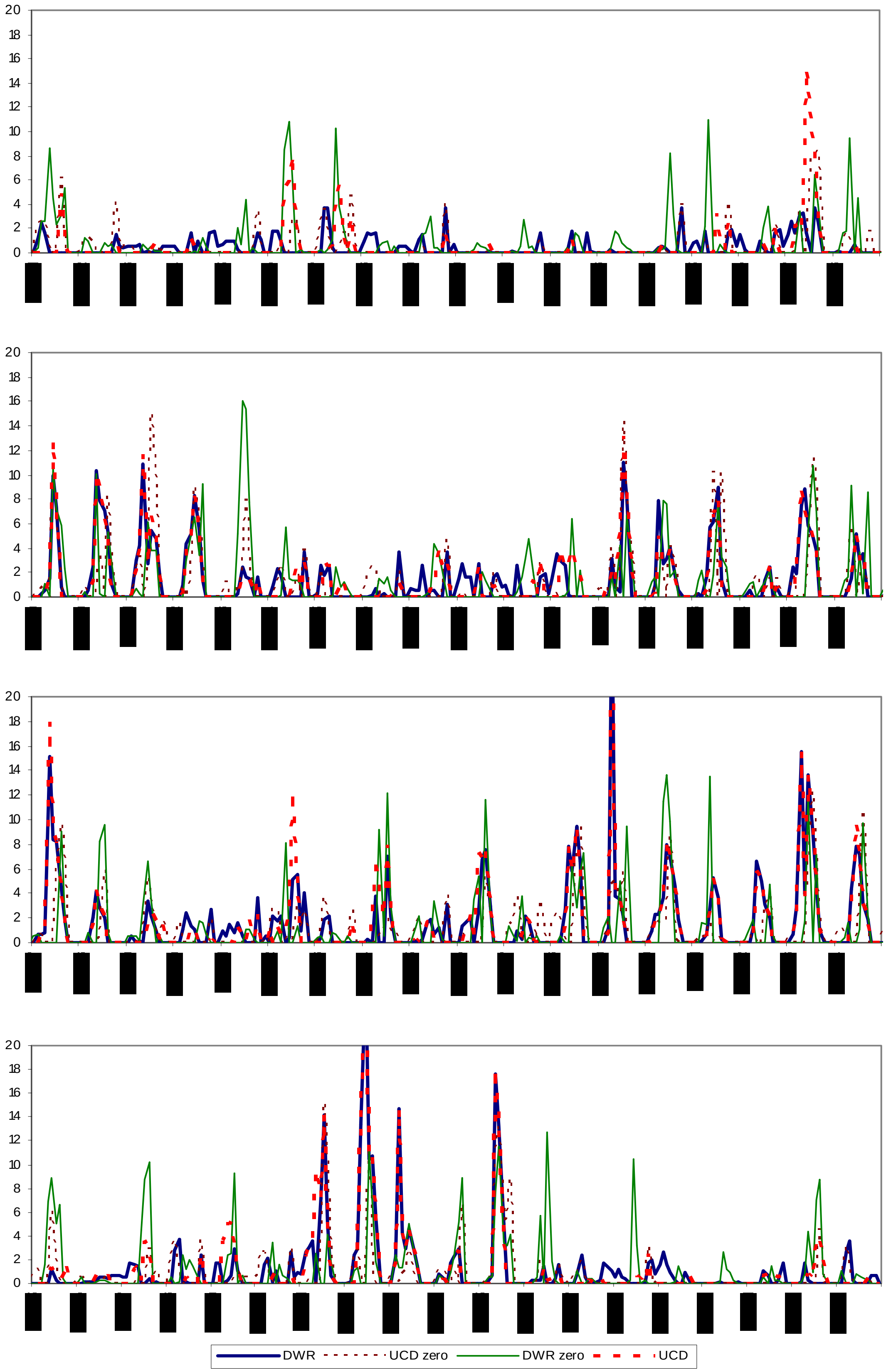

Figure 17. State share of Delta Surplus (thousands cfs).

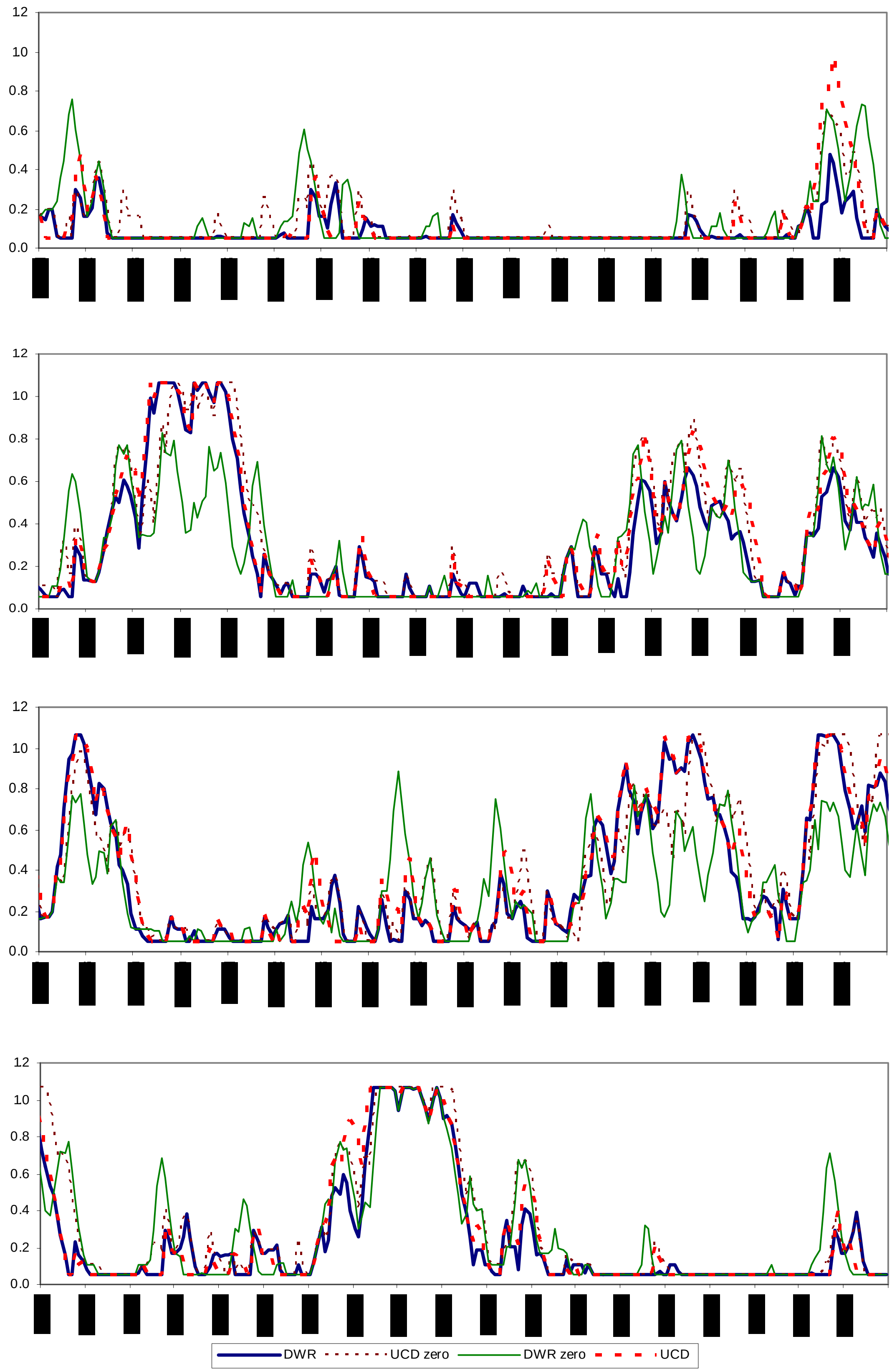

Figure 18. State San Luis Storage (MAF).

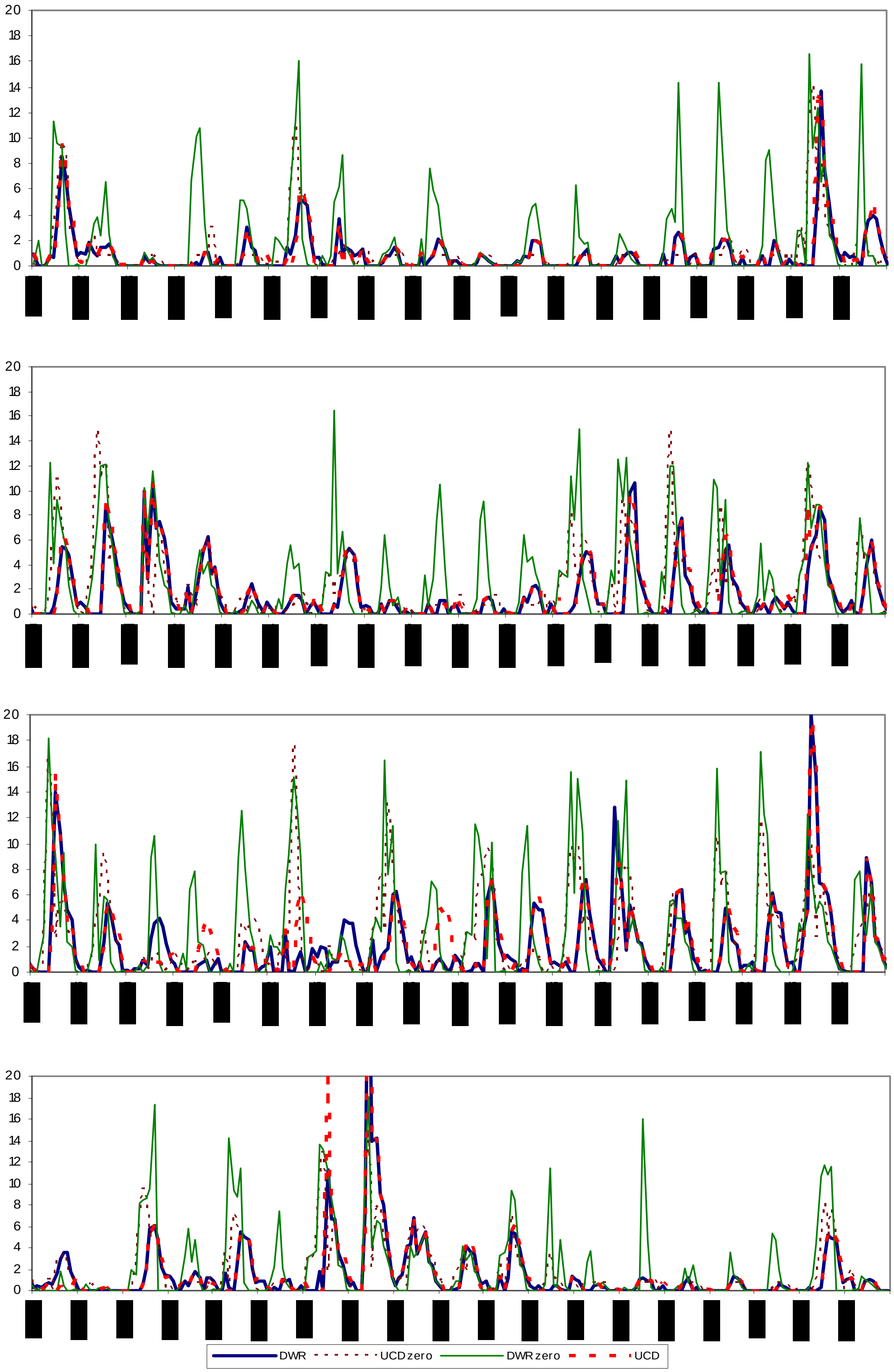

Figure 19. Federal share of Delta Surplus (thousands cfs).

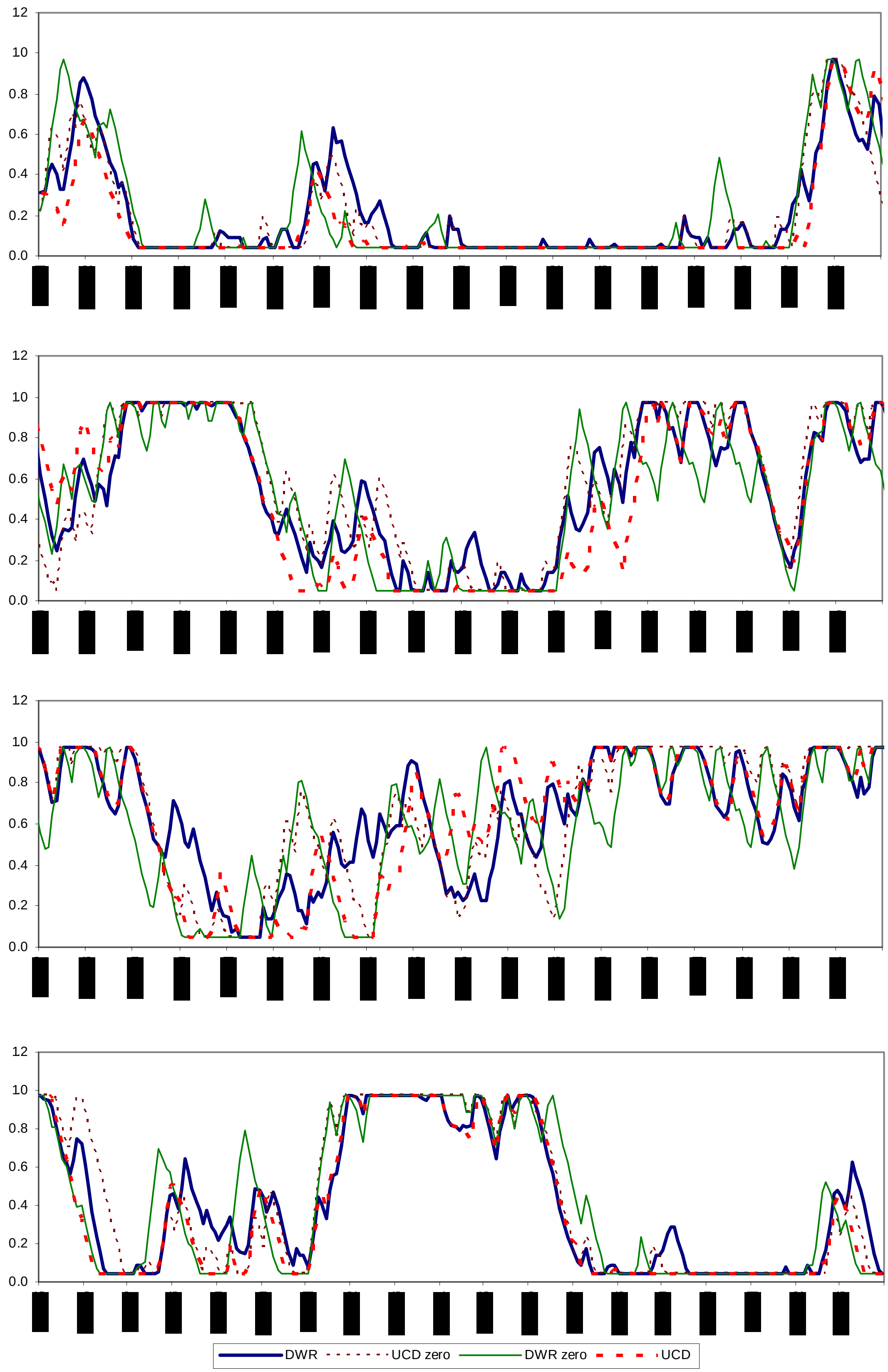

Figure 20. Federal San Luis Storage (MAF).

CHAPTER 5: EQUAL PRIORITIES

Many water distribution systems include demands with equal priority. The initial inclination might be to set equal unit penalty coefficients (penalties or weights) to equal priority demands. However, as seen in Chapter 3, return flows and location within a network both affect the values of priority preserving penalties. Consequently, equal priorities often should result in unequal unit penalties (or weights).

For LP and NFP driven simulations, two questions arise when equal priorities occur. The first question is how to set priority preserving unit penalty coefficients (or weights) for equal priorities given network location and return flows. The second question is how to ensure that water is properly distributed among equal priority users as, by definition, weights or penalties associated with equal priority demands result in multiple optima. When faced with multiple optima, LP or NFP solvers will not distribute water among equal priority demands, but rather, meet demands completely, one at a time, as water is available in an arbitrary order among equal priorities.

Two procedures that ensure proper water allocation among equal priority users are described in this chapter. In the first procedure, the method presented in chapters 2 and 3 is adapted to compute priority preserving penalty coefficients for equal priorities. Once the penalty coefficients are computed, each equal priority demand is subdivided into the same number of smaller demands for which penalty coefficients are found and used in the simulation model. This procedure is described in the section Computing Penalty Coefficients for Equal Priorities: Piecewise Procedure.

The second method is more direct, with each equal priority being split into equal number of sub priorities, which are assigned alternately increasing priority values. For example, consider two users (demand A and demand B) sharing the highest priority. Each priority is subdivided into, for instance, five sub priorities with values that alternate between the two demands. So demand A might have sub demands with priorities 1, 3, 5, 7, and 9, and demand B will have five sub demands with priorities 2, 4, 6, 8, and 10. The second and subsequent priorities in the system are then assigned priorities 11 and greater. Although more straightforward than the previous method, the priority values obtained with this method increase very rapidly, even for small systems. This second procedure is described in the section entitled Computing Penalty Coefficients for Equal Priorities: Alternating Priority Method.

The following two sections present these two alternatives to determine penalty coefficients and how to implement those in an LP or NFP driven simulation to ensure that water is properly distributed among equal priority demands.

COMPUTING PENALTY COEFFICIENTS FOR EQUAL PRIORITIES: PIECEWISE PROCEDURE

In this section a method for computing priority preserving unit penalty coefficients for equal priorities is presented. To do so, the procedure described in chapters 2 and 3 is adapted to handle equal priorities. First, the ordinal rule must be modified to allow weights resulting from equal priorities to be equal; second, objective function contribution for preferred paths when satisfying the equal priority demands must be equated; and lastly, equal priority demands must be split into smaller demands with increasing priorities so the LP is able to split available water among the equal priority demands.

Ordinal Rule for Equal Priorities

In a system with equal priorities, the ordinal rule needs to be modified slightly from the standard form $X_1 > X_2 > \ldots > X_n$, to allow, but not bind, users with equal priorities to have equal unit penalty coefficients. This is explained by means of an example.

Consider a system with demand priorities 1, 2, 2, 2, 3, 4, and 5. In this example there is one user's demand will be met before all others (priority = 1). The unit penalty coefficient associated with this demand is decision variable X_1. Three users have the second highest priority and are associated with decision variables X_2, X_3, and X_4. The remaining three users have unequal, decreasing priorities. In keeping with the ordinal rule, these demands are associated with decision variables X_5, X_6, and X_7.

The standard form of the ordinal rule constraints $X_i > X_{i+1} + \varepsilon$ is modified to allow X_2, X_3, and X_4 to be the same. The ordinal rule constraints become:

C1: $X_1 > X_2 + \varepsilon$
C2: $X_2 > X_5 + \varepsilon$
C3: $X_3 > X_5 + \varepsilon$
C4: $X_4 > X_5 + \varepsilon$
C5: $X_5 > X_6 + \varepsilon$
C6: $X_6 > X_7 + \varepsilon$
C7: $X_7 > \varepsilon$

This set of constraints ensures that X_2 is smaller than X_1, but does not guarantee that X_3 and X_4 are also smaller than X_1. Therefore, two additional constraints are required, namely:

C8: $X_1 > X_3 + \varepsilon$
C9: $X_1 > X_4 + \varepsilon$

The addition of C8 and C9 ensure that the ordinal rule is satisfied with X_2, X_3, and X_4 bound by X_1 and X_5.

When two sets of equal priorities are in consecutive order, without a non-repeated priority between the two sets, another set of constraints is required to ensure the ordinal rule is maintained. To illustrate this, consider the priorities 1,2,2,2,3,3,3,4. These demands are associated with the decision variables X_1, X_2, X_3, X_4, X_5, X_6, X_7, and X_8 respectively. As in the previous example, to allow the penalty coefficients to be the same for repeated priorities, the ordinal rule constraints become:

C1: $X_1 > X_2 + \varepsilon$
C2: $X_2 > X_5 + \varepsilon$
C3: $X_3 > X_5 + \varepsilon$
C4: $X_4 > X_5 + \varepsilon$
C5: $X_5 > X_8 + \varepsilon$
C6: $X_6 > X_8 + \varepsilon$
C7: $X_7 > X_8 + \varepsilon$
C8: $X_8 > \varepsilon$

Figure 21 shows the relative placement of the decision variables on the real number line according to constraints C1 – C8. Constraints C1, C2, and C5 place X_5 between X1 and X8. However, constraints C1-C8 do not provide an upper bound for X_3, X_4, X_6 and X_7.

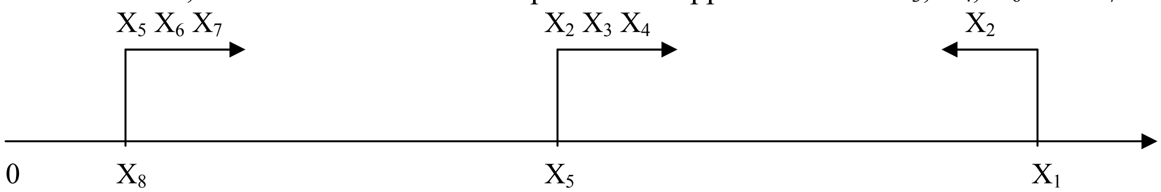

Figure 21. Representation of constraints C1 – C8 on real number line..

To ensure that X_3 and X_4 are smaller than X_1, and that X_6 and X_7 are smaller than X_4, constraints C9 - C12 are included. Constraints C9 and C10 establish an upper bound for X_3 and X_4 and constraints C11 and C12 establish an upper bound for X_6 and X_7. With the inclusion of constraints C9 – C12, the relative location of the decision variables is show on Figure 22. Note that X_2 X_3 X_4 can be located anywhere between X_1 and X_5 and X_5, X_6, and X_7 can be located anywhere between X_1 and X_5.

C9: $X_1 > X_3 + \varepsilon$
C10: $X_1 > X_4 + \varepsilon$
C11: $X_4 > X_6 + \varepsilon$
C12: $X_4 > X_7 + \varepsilon$

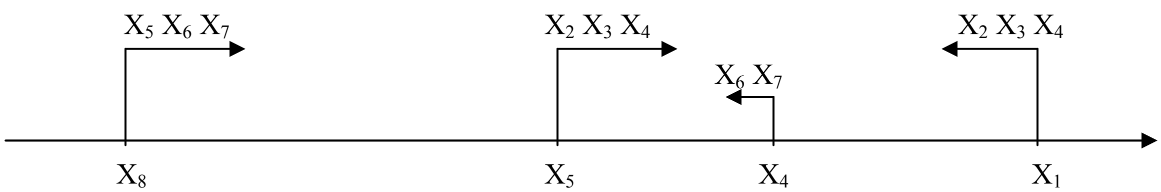

Figure 22. Representation of constraints C1 – C12 on real number line.

Because the two repeated priority groups are consecutive (i.e., second and third priorities), additional constraints are needed. Constraints C13 to C16 ensure that X2 and X3 are greater than X6 and X7. Relative location of decision variables is shown in Figure 23.

The additional ordinal constraints described above ensure the relative position of decision variables associated with unequal priorities without establishing the relative position within the groups of equal priorities. This allows the penalties within each equal priority group to be the same or distinct, depending on other aspects of the physical network.

C13: $X_2 > X_6 + \varepsilon$
C14: $X_3 > X_6 + \varepsilon$
C15: $X_2 > X_7 + \varepsilon$
C16: $X_3 > X_7 + \varepsilon$

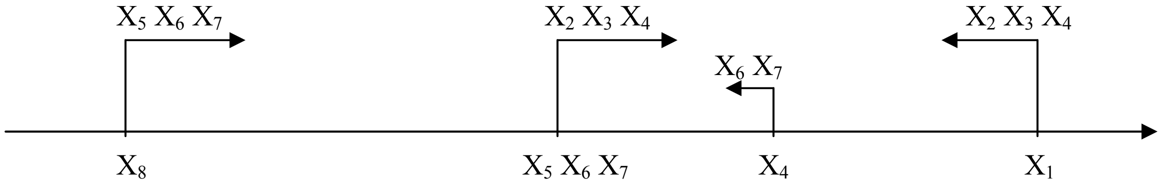

Figure 23. Representation of constraints C1 – C16 on real number line.

Objective Function Contribution

Although necessary, the additional ordinal constraints are not sufficient to ensure priority preserving unit penalty coefficients when priorities are equal. Constraints representing the downstream rule (Equation 2) are formed by including all downstream priorities, without regard to the actual path water might take. Because all downstream demands are combined to form the downstream rule constraints, the computed penalty values are greater than they need to be if only feasible paths are considered. To eliminate this slack and thus ensure priority preserving penalty coefficients for equal priorities, the objective function contribution for water taking particular paths must be equated.

To illustrate this argument, consider the Example 6, based on Network 3 (Figure 8, Chapter 3). The preprocessor input file for example 6 is depicted in Figure 24. In this example, demands D5 and D8 have the same priority, i.e., priority 2.

```
13     | number of columns (n)
500    | dimension of A matrix (rows)  must be >n*n +2n
1.0    | epsilon

                                                                Priority
Rtrn Loc Rtrn Fctr Calsim
S1 D2 D3 S4 D5 D6 S7 J8 D8 D9 D10 S11 D12           rl(i)     r(i)
Nodes
1  0  0  0  0  0  0  0  0  0   0   0   0     8       1         0
S1
1  0  0  0  0  0  0  0  0  0   0   0   0     5       3        .5
D2
1  1  0  0  0  0  0  0  0  0   0   0   0     6       8        .5
D3
0  0  0  1  0  0  0  0  0  0   0   0   0     9       4         0
S4
0  0  0  1  0  0  0  0  0  0   0   0   0     2       7        .5
D5
0  0  0  1  1  0  0  0  0  0   0   0   0     7       7        .5
D6
0  0  0  1  1  1  1  0  0  0   0   0   0     4       7         0
S7
1  1  1  1  1  1  1  0  0  0   0   0   0     0       8         0
J8
1  1  1  1  1  1  1  1  0  0   0   0   0     2       10       .5
D8
1  1  1  1  1  1  1  1  1  0   0   0   0    12       10        0
D9
1  1  1  1  1  1  1  1  0  0   0   0   0     1       12       .5
D10
1  1  1  1  1  1  1  1  0  0   1   1   0    11       12        0
S11
1  1  1  1  1  1  1  1  0  0   1   1   0    10       14       .5
D12
```

Figure 24. Input File for Example 6.

Figure 25 shows the LP for this problem, with added constraint (C170) that equates the objective function contribution for one unit of water diverted at D5 to the objective function contribution of one unit of water diverted at D8. The objective function contribution of one unit of water delivered to D5 is ($X_2 + 0.5\ X_3 + 0.25\ X_{12}$) because the return flow from D5 will go to meet the highest priority demand, D8 in this case, whose return flow, in turn, is diverted by D9. The objective function contribution of one unit of water delivered to D8, on the other hand is (X_3 + 0.5 X_{12}). The result of equating the two objective function contributions is constraint C170.

```
Minimize: X12 – X1

Constraints
C1: X8 - X9 >= 1
C2: X5 - X6 >= 1
C3: X6 - X7 >= 1
C4: X9 - X10 >= 1
C5: X2 - X4 >= 1
C6: X7 - X8 >= 1
C7: X4 - X5 >= 1
C8: X3 - X4 >= 1
C9: X12 >= 1
C10: X1 - X2 >= 1
C11: X11 - X12 >= 1
C12: X10 - X11 >= 1
C13: X1 - X3 >= 1
C14: X8 - X10 - X11 - X12 >= 1
C15: X5 - 0.5 X6 - 0.5 X10 - 0.5 X11 - 0.5 X12 >= 1
C16: X6 - 0.5 X10 - 0.5 X11 - 0.5 X12 >= 1
C17: X9 - X10 - X11 - X12 >= 1
C18: X2 - 0.5 X4 - X7 - 0.5 X10 - 0.5 X11 - 0.5 X12 >= 1
C19: X7 - 0.5 X10 - 0.5 X11 - 0.5 X12 >= 1
C20: X4 - X10 - X11 - X12 >= 1
C21: X3 - 0.5 X12 >= 1
C23: X1 - 0.5 X10 - 0.5 X11 >= 1
C103: X4 - 2 X7 >= 1
C111: X3 - 2 X5 >= 1
C112: X3 - 2 X6 >= 1
C115: X3 - 2 X7 >= 1
C135: X1 - 2 X5 >= 1
C136: X1 - 2 X6 >= 1
C138: X1 - 2 X2 >= 1
C139: X1 - 2 X7 >= 1
C170: X2 - 0.5 X3 - 0.25 X12 = 0
```

Figure 25. LP for Example 6

Preprocessor results are shown on Table 9. From weights on Table 9, the objective function contribution of one unit of water delivered to D5 is $X_2 + 0.5\ X_3 + 0.25\ X_{12} = 45$, and the objective function contribution of one unit of water delivered to D8 is $X_3 + 0.5\ X_{12} = 45$. The equal objective function contribution ensures that the LP or NFP is indifferent to delivery to either D5 or D8.

<u>Test and Results</u>

The system of Example 6 with weights as shown on Table 9 was simulated. Inflows and water allocations are shown on Table 10. In this simulation, all demands are set to 10 taf and storage capacities to 50 taf. Inflow to reservoir S1 is set to zero to simplify the analysis of results. Consequently, simulated deliveries to S1, D2, and D3 are zero and thus omitted from the table. Simulated allocations show a preference to deliver available water to one user, in this case D5, only delivering to D8 (the other user of equal priority) after D5 is fully satisfied, which occurs in the fourth month simulated.

Table 9. Computed Weights for Example 6

Demand	Priority	Weight
S1	8	8
D2	5	11
D3	6	10
S4	9	7
D5	2	22.5
D6	7	9
S7	4	19
D8	2	44.5
D9	12	1
D10	1	46
S11	11	2
D12	10	3

To encourage the LP to distribute water among equal priority users, equal priority demands can each be split into smaller demands. Consider Example 6, in which each demand is 10 taf. Demands D5 and D8 are split into n equal demands, D5/n and D8/n. If, for instance, they are each divided into four (n=4) sub-demands each, say D5a, D5b, D5c, D5d, and D8a, D8b, D8c, D8d, the sub-demand values will each be 2.5 taf. What we would like the LP to do is to satisfy these in turn, D5a, D8a, D5b, D8b, D5c, D8c, D5d, and finally D8d.

Table 10. Water Allocation for Example 6

Month	I4 Inflow TAF	S4 Storage TAF	D5 Delivery TAF	D6 Delivery TAF	S7 Storage TAF	D8 Delivery TAF	D9 Delivery TAF	D10 Delivery TAF	S11 Storage TAF	D12 Delivery TAF
1	10	0	0	0	0	0	0	10	0	5
2	12	0	4	0	0	0	0	10	0	5
3	14	0	8	0	0	0	0	10	0	5
4	16	0	10	0	0	1	0.5	10	0	5
5	18	0	10	0	0	3	1.5	10	0	5
6	20	0	10	0	0	5	5	10	0	5
7	22	0	10	0	0	7	2.5	10	0	5
8	24	0	10	0	0	9	4.5	10	0	5
9	26	0	10	0	1	10	5	10	0	5
10	28	0	10	0	4	10	5	10	0	5
11	30	0	10	0	9	10	5	10	0	5
12	32	0	10	0	16	10	5	10	0	5

In Example 6, the decision variables for D5 and D8 are X_2 and X_3, respectively, and the sub-demands have the subscripts a, b, c, and d. As with the full demands D5 and D8, the objective function contributions for variables X_2 and X_3 are used to ensure that water takes one path rather than another and consequently water is delivered as desired.

The objective function contributions of one unit of water delivered to D5 or D8 are (X_2 + 0.5 X_3 + 0.25 X_{12}) and (X_3 + 0.5 X_{12}), respectively. Therefore, to ensure delivery to D5 before any water is delivered to D8, the objective function contribution of water delivered at X_2 must be greater than the objective function contribution of water delivered at X_3. Therefore, the following inequality must be satisfied:

$$X_{2a} + 0.5\ X_{3a} + 0.25\ X_{12} > X_{3a} + 0.5\ X_{12} \quad (48)$$

If X_{3a} is satisfied next, the objective function contribution of water delivered by D8 must be greater than the objective function contribution of water delivered at D5, and inequality (2) must be satisfied:

$$X_{3a} + 0.5\ X_{12} > X_{2b} + 0.5\ X_{3a} + 0.25\ X_{12} \quad (49)$$

Continuing this logic to satisfy demands D5b, D8b, D5c, D8c, D5d, and D8d in turn, inequalities (3) to (7) must also be satisfied.

$$X_{2b} + 0.5\ X_{3b} + 0.25\ X_{12} > X_{3b} + 0.5\ X_{12} \quad (50)$$

$$X_{3b} + 0.5\ X_{12} > X_{2c} + 0.5\ X_{3b} + 0.25\ X_{12} \quad (51)$$

$$X_{2c} + 0.5\ X_{3c} + 0.25\ X_{12} > X_{3c} + 0.5\ X_{12} \quad (52)$$

$$X_{3c} + 0.5\ X_{12} > X_{2d} + 0.5\ X_{3c} + 0.25\ X_{12} \quad (53)$$

$$X_{2d} + 0.5\ X_{3d} + 0.25\ X_{12} > X_{3d} + 0.5\ X_{12} \quad (54)$$

We also want to ensure that the lowest value of X_{2i} and X_{3i} are those found by the preprocessor, that is:

$$X_{2d} \geq 22.5 \quad (55)$$

$$X_{3d} \geq 44.5 \quad (56)$$

It is also desirable that the values found are not too large; therefore, $X_{3a} + X_{2a}$ should be minimized.

With $X_{12} = 1$, and re-writing the equations in standard LP format:

Minimize: $X_{3a} + X_{2a}$

Subject to:

$$\begin{aligned}
&X_{2a} - 0.5\ X_{3a} - 0.25 \geq \alpha\\
&X_{2b} - 0.5\ X_{3a} - 0.25 \leq \alpha\\
&X_{2b} - 0.5\ X_{3b} - 0.25 \geq \alpha\\
&X_{2c} - 0.5\ X_{3b} - 0.25 \leq \alpha\\
&X_{2c} - 0.5\ X_{3c} - 0.25 \geq \alpha\\
&X_{2d} - 0.5\ X_{3c} - 0.25 \leq \alpha\\
&X_{2d} - 0.5\ X_{3d} - 0.25 \geq \alpha\\
&X_{2d} \geq 22.5\\
&X_{3d} \geq 44.5
\end{aligned}$$

With α=0.05, the solution to this LP is:

$X_{2a} = 22.85$, $X_{2b} = 22.75$, $X_{2c} = 22.65$, $X_{2d} = 22.55$, $X_{3a} = 45.1$, $X_{3b} = 44.9$, $X_{3c} = 44.7$, and $X_{3d} = 44.5$.

If the water simulation model is modified so that demands D5 and D8 are each split into four, with weights as defined by the LP above, the resulting simulated water allocation is as presented on Table 11. The greater the number of steps used to split demands D5 and D8, the more even the distribution of water will be between D5 and D8.

Table 11. Water Allocation for Examples 6a and 7.

Month	I4 Inflow TAF	S4 Storage TAF	D5 Delivery TAF	D6 Delivery TAF	S7 Storage TAF	D8 Delivery TAF	D9 Delivery TAF	D10 Delivery TAF	S11 Storage TAF	D12 Delivery TAF
1	10	0	0	0	0	0.00	0.00	10	0	5
2	12	0	2.5	0	0	0.75	0.38	10	0	5
3	14	0	3.0	0	0	2.50	1.25	10	0	5
4	16	0	5.0	0	0	3.50	1.75	10	0	5
5	18	0	6.0	0	0	5.00	2.50	10	0	5
6	20	0	7.5	0	0	6.25	3.13	10	0	5
7	22	0	9.0	0	0	7.50	3.75	10	0	5
8	24	0	10	0	0	9.00	4.50	10	0	5
9	26	0	10	0	1	10.00	5.00	10	0	5
10	28	0	10	0	4	10.00	5.00	10	0	5
11	30	0	10	0	9	10.00	5.00	10	0	5
12	32	0	10	0	16	10.00	5.00	10	0	5

COMPUTING PENALTY COEFFICIENTS FOR EQUAL PRIORITIES: ALTERNATING PENALTY PROCEDURE.

An alternative method for computing priority preserving unit penalty coefficients for equal priorities is to split each equal priority into an equal number of sub priorities and assign alternating increasing priority values to the sub priorities.

Taking, for instance, Example 6, D5 and D8 are each split into four sub priorities D5a, D5b, D5c, D5d, D8a, D8b, D8c, and D8d. The priorities for D5i are 2, 4, 6, 8, and the priorities for D8i are 3, 5, 7, and 9. The other lower priorities in this example are assigned priorities with priority values starting at 10. That is, S7 is assigned priority 10, D2 is assigned priority 11 and so on to the lowest priority D9 which is assigned priority 18.

The preprocessor input file for this example (Example 7) is presented in Figure 26, and the LP solution is shown on Table 12. The range of weights obtained with the second method is considerably greater than with the first method, and grows very rapidly, as the number of sub priorities considered increases. In this example, by increasing the number of sub priorities from four to five, the highest priority becomes 1,149.

<u>Test and Results</u>

Simulated water allocation for weights shown on Table 12 are the same as those for Example 6 and shown on Table 11.

```
        19      | number of columns (n)
        1000    | dimension of A matrix (rows) must be >n*n +2n
        1.0
 epsilon
                  D5a   D5c    D6    J8   D8b   D8d  D10  D12       RtLoc
Nodes
        S1 D2 D3 S4   D5b   D5d    S7   D8a   D8c   D9    S11      Prty
RtFact
        1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    14     1
0     S1
        1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    10     3
.5     D2
        1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    11     8
.5     D3
        0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    14     4
0     S4
        0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     2    10
.5    D5a
        0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     4    10
.5    D5b
        0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     6    10
.5    D5c
        0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     8    10
.5    D5d
        0  0  0  1  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0    12    10
.5     D6
        0  0  0  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0  0    18    10
0     S7
        1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0  0     0    12
0     J8
        1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0     3    16
.5    D8a
        1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0     5    16
.5    D8b
        1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0     7    16
.5    D8c
        1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0     9    16
.5    D8d
        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0    17    20
0     D9
        1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0     1    18
.5    D10
        1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  1  1  0    16    18
0    S11
        1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  1  1  0    15    20
.5    D12
```

Figure 26. Input File for Example 7.

Table 12. Computed Weights for Example 7

Demand	Priority	Weight
S1	8	8
D2	5	11
D3	6	10
S4	9	7
D5a	2	286
D5b	4	142
D5c	6	70
D5d	8	34
D6	7	9
S7	4	10
D8a	3	285
D8b	5	141
D8c	7	69
D8d	9	23
D9	12	1
D10	1	573
S11	11	2
D12	10	3

CONCLUSIONS

Equal priorities can be readily represented and their unit penalties rigorously determined. This chapter presented two methods to compute weights for equal priority demands. Each method has advantages and disadvantages. While the first method is more complex, it results in a range of weights that is fairly small. The second method is considerably more straightforward. However, it produces a much greater range of weights, something that might be problematic for large networks.

CHAPTER 6: NEGATIVE WEIGHTS

INTRODUCTION

Negative weights are used in LP and NFP driven simulations when flow through a particular network arc is not desired. Negative weights are often used to avoid spilling water from a system, minimizing flows above flood stage in natural channels, and to avoid encroachment into the flood control pool of a reservoir.

Negative weights also are used to avoid the use of hard constraints that might cause infeasibilities. For instance, rather than setting a maximum capacity on the flow in a channel to avoid flooding, the channel is split into multiple arcs. The maximum flow, or flood stage, for the main channel arc is set as a hard constraint, with no associated weight, and the arc(s) that carry water in excess of the maximum channel capacity are given negative weight(s). The negative weights send flood flows elsewhere in the system, if possible. However, unlike a hard constraint, the negative weight allows flood flows to occur on the excess flow arc when other flow options are unavailable, or to prioritize flooding outcomes.

The flood control pool of a reservoir is often given a negative weight smaller than the weight associated with the flood channel(s) immediately downstream, reflecting the priorities where excess flows should be avoided first, second, etc. Consequently, water tends to be temporarily stored in the flood control pool and released as soon as total release from the reservoir can be kept below the critical flood stage capacity of the downstream channel.

While positive weights allocate limited supplies according to priority, negative weights are usually used in surplus conditions, to allocate water to where excess water is least damaging in an ordered ranking or priority system with regard to excess water. Thus, both positive and negative weights can apply to different flow ranges. This is common with NFP driven simulation models.

For LP driven simulation models with more general (non-NFP) linear constraints, negative weights can be used in contexts other than surplus flow conditions. In LP driven simulations, negative weights can be used to indicate the degree to which a desirable operational outcome is achieved, and also to ensure that non-NFP constraints truly reflect operating criteria. For instance, negative weights can be used in soft constraints that set a minimum or maximum desired flow (allocation). Consider equation 57.

$$X + slack - surplus = 100 \tag{57}$$

For maximization, if the variable *slack* is assigned a negative weight and the variable *surplus* is not assigned a weight, the LP will be less likely to allocate water to the slack variable, and the LP will prefer $X>100$ to $X<100$. In this instance, the combination of constraint and negative weight is used to simulate a minimum desired flow without imposing it rigidly with no exception. Conversely, by assigning a negative weight to the variable *surplus* and no weight to the variable *slack*, the LP is more likely to allocate less than 100 to the decision variable *X*, and rendering the constraint a maximum desired flow. The degree to which the desired operation criterion is attained is a function of the objective function coefficient (“weight”) of the surplus or slack variables.

In LP driven simulations, negative weights also can help balance decision variables within a constraint or constraint set to ensure that constraints are simulated as intended. Such weights are generally associated with non-arc flow variables. Equations (44)-(47) of Chapter 4 and non-arc flow decision variables UNUSED_FS and UNUSED_SS provide an example of this type of constraint set and this particular type of negative weighted variables.

The three types of negative weights described above play different roles in the LP. The distinct contexts in which the negative weights appear require distinct approaches in their computation.

COMPUTATION OF NEGATIVE WEIGHTS

In its present form, the algorithm presented in Chapter 2 is not suitable to compute a priority preserving set of positive and negative weights simultaneously. Consider the network depicted in Figure 27, where the reservoir S1 is split into four pools, namely, S1_1, S1_2, S1_3, and S1_4. The priorities for this example are shown on Table 13.

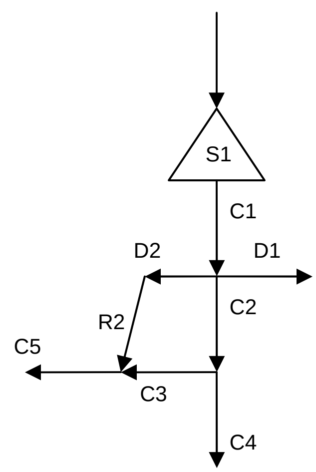

Figure 27. Example Network.

Setting the return flow fraction for D2 to 0.5, the lowest positive priority, S1_2, to at least 1 and the highest negative priority, S1_4, to at most -1, and ε to 1, the automated weight generator creates the LP presented in Figure 28, where X_i represents the ith priority.

Table 13. Example Priorities and Computed Weights

Node or Arc	Priority	LP Variable	Sign	Computed Weight
S1_1	2	X2	Positive	3
S1_2	4	X4	Positive	1
S1_3	5	X5	Negative	-1
S1_4	8	X8	Negative	-4
D1	3	X3	Positive	2
D2	1	X1	Positive	4

C4	7	X7	Negative	-3
C5	6	X6	Negative	-2

The computed weights are also shown in Table 13. While properly ranked, the weights in Table 13 are not priority preserving. This can be seen in Table 14, where the objective function values for a unit of water for each feasible path are shown. The highest objective function value for one unit of water is 3, which can be achieved by allocating the water to either S1_1 or D2. Because water can be allocated to S1_1 before being allocated to D2, the set of weights presented in Table 13 is not priority preserving.

```
Min X1-X8
Subject to:
C1:  X1-X2>=1
C2:  X2-X3>=1
C3:  X3-X4>=1
C4:  X4-X5>=1
C5:  X5-X6>=1
C6:  X6-X7>=1
C7:  X7-X8>=1
C8:  X4>=1
C9:  X5<=-1
C10:  X1-X3-0.5X6-X7>=1
C11:  X2-X3-X4-X5-X6-X7-X8>=1
C12:  X3-X6-X7>=1
C13:  X4-X5-X6-X7-X8>=1
C14:  X5-X6-X7-X8>=1
```

Figure 28. LP for Example in Figure 27.

Table 14. Objective Function Value for Feasible Paths for One Unit of Water (1)

Network Path	Objective Function Value
S1_1	3
C1 D2 C5	3
C1 D1	2
S1_2	1
S1_3	-1
C1 C2 C3 C5	-2
C1 C2 C4	-3
S1_4	-4

The algorithm fails in simultaneously computing positive and negative weights because the downstream rule (Equation 37, Chapter 2) is a linear combination of all junior priorities downstream. This is appropriate when all weights are positive, but not when negative weights are included. Consider the example presented above, where constraints C10 and C11 are the downstream rule for D2 (X1) and S1_1 (X2). Variables X5-X8 represent negative weights and are therefore negative. When the computed weights are substituted into C10 and C11, these constraints become non-binding.

C10: $X1-X3-0.5X6-X7>=1$

$$X1>= 1 + X3 + 0.5X6 + X7$$

$$X1>= 1 + 3 + 0.5(-2) + (-3) = 0$$

C11: $X2-X3-X4-X5-X6-X7-X8>=1$

$$X2>= 1 + X3 + X4 + X5 + X6 + X7 + X8$$

$$X2>= 1 + 3 + 1 + (-1) + (-2) + (-3) + (-4) = -5$$

If a senior demand is located upstream of negative weighted priorities, the resulting senior weight is reduced by the negative weights downstream from it. Depending on the priorities and the network configuration the downstream rule may be rendered non-binding where it would otherwise have been. The value of the senior weights (D2 and S1_1) will be determined by the other constraints in the algorithm. In the example above the binding constraints are the ordinal constraints ($X_i >= X_i + \varepsilon$). The algorithm fails when positive and negative weights are computed simultaneously.

SURPLUS FLOW CONDITIONS

As discussed in the introduction to this chapter, negative weights often are used when it is desired to avoid allocation of water to some network arcs or nodes under excess flow conditions. In this case, the priorities for allocating excess water among the arcs are usually well defined. Consider, for instance, typical locations where excess flow is routed: outflow from a system, reservoir flood control pools, and arcs representing flows exceeding channel carrying capacity. It is undesirable to have water allocated to any of those variables; however, the priority is usually clear: the highest priority is to avoid water above dam safety levels in the dam, then flood stage of a river reach, then to avoid encroachment of the flood control pool, and then to avoid spilling water out of the system. Also, because these decision variables represent physical flows, they are associated with a specific location within the network.

Assigning weights to these variables, therefore, is similar to assigning positive weights to demand or delivery variables where the allocation of water is desired at specific locations within the network. The weight generator pre-processor can be reapplied separately for the negative weighted variables in the same way it is used for the positive weights, and the resulting weights multiplied by -1.

The separate computation of these weights is appropriate as long as the positive and negative weight sets represent mutually exclusive events. That is, the positive weight set is required when water is not available to meet all demands, while the negative weight set is used to allocate excess water to where it causes the least damage. These conditions are usually mutually exclusive.

However, there are circumstances in which an arc to which water allocation is not desired (negative weight) receives water because of water allocation to a positive weighted priority demand. In this case, this arc should be removed from the priority computation and set separately. Setting weights for this type of arc-flow variable is presented in the next section.

Calibrated Coefficients

The network presented earlier in this chapter and the Two-River System model both provide examples of situations in which an arc where flow is not desired receives water because of water allocation to a positive weighted priority demand. In the network example in this chapter, flow is not desired on arc C5. However, return flow from D2, the highest priority for flow allocation, necessarily flows into C5. In the case of the Two-River System model, the negative weighted surplus Delta outflow variables (C34B_SWP and C34B_CVP) are assigned water due to an inflow-export ratio restriction (see Chapter 4). For these situations, the negative weights for those arcs require manual calibration in relation to the remaining prioritized weight set, rather than a prioritized weight itself. Whenever water will necessarily flow out of the system, either because of return flows, minimum instream flow requirements upstream, or rules such as the inflow-export limits, the negative weight given to the outflow (or sink) arc, should be manually calibrated.

Consider the network presented in Figure 27. Table 15 (case A) presents positive and negative weights computed separately as described in the previous section, and computed to ensure that the weights are priority preserving (case B). The difference in weight computation for cases (A) and (B) is in the value of ε. In case (A), ε=1 for positive and negative weights, and in case (B) ε=1 for positive weights and ε=0.5 for negative weights. In the case of this simple network, where water paths are easily enumerated and non-NFP constraints are absent, the weight generator parameter ε can be calibrated to obtain a priority preserving set of positive and negative weights.

Table 16 presents the objective function contribution for each possible path for one unit of water for case (A) and case (B). If the weight computation is performed as described in the previous section, the objective function values for one unit of water being delivered to S1_1 is the same as for one unit of water delivered to D2 (C1 D2 C5) in case (A), but not in case (B), where objective function values preserve the priorities shown in Table 16.

Table 15. Example Priorities and Computed Weights

Sign	Node or Arc	Priority	Computed Weight (A)	Computed Weight (B)
Positive	D2	1	5	5
	S1_1	2	3	3
	D1	3	2	2
	S1_2	4	1	1
Negative	S1_4	1	-6	-4
	C4	2	-3	-2
	C5	3	-2	-1.5
	S1_3	4	-1	-1

Priority preserving weight sets could also have been obtained by increasing the value of ε for the positive weights or increasing the value of the lowest positive weight and leaving ε=1 and the baseline (lowest) weight as in case (A) for the negative weights (=-1).

Table 16. Objective Function Value for Feasible Paths for One Unit of Water (2)

Network Path	Objective Function Value (A)	Objective Function Value (B)
S1_1	4	4
C1 D2 C5	4	4.25
C1 D1	2	2
S1_2	1	1
S1_3	-1	-1
C1 C2 C3 C5	-2	-1.5
C1 C2 C4	-3	-2
S1_4	-6	-4

For large networks, however, the enumeration of paths and their objective function contribution can be complex and time consuming. Furthermore, when non-NFP constraints are involved, this task may become daunting, if not impossible. Therefore, a more traditional calibration procedure is preferable for these priorities (described in chapters 7 and 8). In the case of the Two-River System model, there is a trade-off between exporting water to meet south-of-Delta priorities while allowing water to flow out of the system as surplus Delta outflow and keeping water in storage north of the Delta.

The most practical way to obtain the objective function coefficient for the Delta surplus outflow variables (*C34B_SWP* and *C34B_CVP*) is to manually calibrate these parameters given a priority preserving set of positive weights and the objective function coefficient for the non-NFP variables *UNUSED_SS* and *UNUSED_FS*. At this point, values of different orders of magnitude may be given to negative weights representing arcs where flow is to be avoided. The model user can select values for *UNUSED_SS* and *UNUSED_FS* that result in simulations best reflecting real or desired operations. Once the coefficients of *C34B_SWP* and *C34B_CVP* have been selected, the algorithm may be applied for the negative weights using the coefficients for *C34B_SWP* and *C34B_CVP* as the baseline values (i.e., lowest weight value).

BALANCE WEIGHTS

Negative weights used to balance decision variables within a constraint or constraint set, must, be chosen according to the constraints and the decision variables they must balance. Because of the individual nature of such constraints and their non-NFP nature, the computation of balancing weights cannot be generalized using the approach presented in Chapter 2.

In the example of the Two River System model presented in Chapter 4, the variables *UNUSED_SS* and *UNUSED_FS* fall into this category. The weights associated with these variables must balance the weights associated with demands south of the Delta to ensure accurate representation of the COA constraints.

The following COA constraints were presented in Chapter 4. They represent the SWP's portion of the COA.

$$D34A - SWPDS + D34D\_EXP1 + C34B\_SWP + UNUSED\_SS + 0.25\ IBU - 0.45\ UWFE = 0 \quad (58)$$

$$D34C\_EXP2 - UNUSED\_SS <= 0 \quad (59)$$

The COA constraint in equation (49) can be considered a mass balance of SWP water in the system. The sources of available water are given by the terms with negative coefficients, while the allocations are the terms with positive coefficients. The variable *IBU* represents in-basin water use to be met by the projects (CVP and SWP) when the Delta is in balance. *UWFE* represents the surplus water present in the Delta available to be split between the projects. *IBU* and *UWFE* are linked to integer variables so that if *IBU* exceeds zero, *UWFE* is zero and vice-versa.

The decision variable *D34A* has a greater priority than demands south of the Delta, and therefore will be met before water is exported from the Delta. The variables *D34D_EXP1* (SWP) and *D34C_EXP2* (CVP) are implicitly represented in the objective function by decision variables representing delivery and storage demands south of the Delta. The weight assigned to *UNUSED_SS* must ensure that the SWP exports, under *D34D_EXP1*, all the water it can within physical and operating limits such as the inflow-export limits and to which it is entitled under the COA. The negative weight associated with *UNUSED_SS*, therefore, must exceed (in magnitude) any positive weight on CVP demands south of the Delta (and implicitly *D34C_EXP2*). When the negative weight on *UNUSED_SS* exceeds weights on all demands south of the Delta, the objective function will accrue a positive value by allocating water to *D34D_EXP1*, and a negative value if allocated to *UNUSED_SS* (and in turn to *D34C_EXP2*). The proper value of *UNUSED_SS* will guarantee that water that belongs to the State will be pumped by the State (up to pumping capacity). By properly selecting the weights for variables that balance other weights within the model constraints, operating and regulatory criteria which do not fall under a pure priority based allocation of water are simulated as intended.

SOFT CONSTRAINTS

Soft constraints are used when a target is desired and deviation from the target is penalized with a negative weight (for a maximization formulation). Consider equations (60) and (61) where *D34C* represents Tracy Pumping Plant and *D34D* represents Banks Pumping Plant. The variables *SURPL0120* and *SURPL0121* are not weighted, but *SLACK0120* and *SLACK0121* appear in the objective function with a coefficient of -2000. This constraint is to set the minimum *desired* pumping at Tracy and Banks at 800 cfs and 300 cfs, respectively. Rather than setting the minimum pumping as a hard constraint ($D34C \geq 800$ and $D34D \geq 300$), soft constraints avoid infeasibilities, but make policies happen "most" of the time. The negative weights on the slack variables discourage the LP from assigning non-zero values to those variables.

$$D34C - SURPL0120 + SLACK0120 = 800 \quad (60)$$

$$D34D - SURPL0121 + SLACK0121 = 300 \quad (61)$$

Soft constraints with slack and surplus variables must also be computed taking account other decision variables. In times of short water supplies, the minimum desired pumping at Tracy and Banks compete for water with the variables *UNUSED_FS* and *UNUSED_SS*. Examining the weights in Table 1 of Chapter 4, we see that DWR has chosen to give higher priority to meeting the minimum desired pumping over water ownership according to the COA. That is, because the weight for *UNUSED_FS* and *UNUSED_SS* is -1285 and the weight for *SLACK0120* and *SLACK0121* is -2000, in times of short water supply, the LP will favor meeting the minimum desired pumping at the expense of correctly assigning SWP's water to SWP export and CVP's water to CVP export. If the weight on the slack variables is less than the weight for *UNUSED_FS* and *UNUSED_SS*, the LP will favor assigning water to the rightful owner according to the COA over meeting each project's minimum desired pumping. This is another case in which the weights must be calibrated individually after the priority based demand weights have been assigned, at magnitudes that balance other competing demands, to ensure accurate representation of the real system.

CONCLUSIONS

In this chapter I discussed the use of negative weights in LP driven simulations, and methods for computing them. A simple network example was used to illustrate how the simultaneous computation of positive and negative weights can not be done with the weight generator described in Chapter 2 without violating priorities.

For the purposes of computing negative weights, the variables having negative weights were split into two categories: (i) weights used in surplus conditions, to allocate water to where excess water is least damaging, and (ii) weights used to balance decision variables within a constraint or constraint set in a way that ensures that the constraints are simulated as intended.

For variables associated with surplus flow conditions, negative weights can be computed separately using the method described in Chapter 3, in the same way as the computation of positive weights. The method performs just as well for arc-flow variables were flow is desired as for arc-flow variables were flow is not desired as long as the computation of negative weights is done separately from that of positive weights.

In the case of variables that balance decision variables within a constraint or constraint set to ensure that the constraints are simulated as intended, the weights must be obtained individually and manually, given the trade-offs determined by the constraints.

CHAPTER 7: PRIORITY PRESERVING WEIGHTS FOR LINEAR PROGRAMMING DRIVEN SIMULATIONS

The simulation test examples presented in Chapter 3 all fall into the classical generalized NFP formulation. The classical generalized NFP consists of a linear objective function with only arc flow variables and linear constraints limited to mass balances with linear gains or losses at each node and upper and lower bounds directly on arc flow decision variables (Chapter 2, equations (2) and (3), respectively). The constraints in a generalized NFP simply set the spatial connections, or "plumbing", and the capacity on each connection (or arc). Each decision variable in a NFP driven simulation represents flow in one specific network arc. Examples in Chapter 3, therefore, illustrate the computation and use of priority preserving weights for NFP.

Weights computed by the method described in Chapter 2 also can be priority preserving when applied to more general linear programming (LP). Brown (2005) successfully applied this method for linear program representation of flood control operations in Iowa. This chapter demonstrates that as long as the objective function is linear, the inclusion of linear non-NFP constraints does not affect the priority preserving quality of the weights generated with the automated procedure.

In this chapter I discuss how the weight set generated by the method described in Chapter 2 is priority preserving for NFP driven simulations. I further extend this to discuss how different types of non-NFP linear constraints can affect the allocation of water in an LP driven simulation and how the allocation remains priority preserving. I conclude this chapter with a step-by-step procedure to compute weights for LP driven simulations.

PRIORITY PRESERVING WEIGHTS FOR NFP DRIVEN SIMULATIONS

Consider how a unit of water is allocated by a NFP solver. A unit of water will take a network path that optimizes the objective function (minimization or maximization). Within this path, a unit of water can be allocated to storage or flow demands. Flow demands may or may not generate return flows. If a unit of water is allocated to a non-consumptive flow demand (e.g., minimum instream flow), or to a demand that returns a fraction of its allocation, the return flow can continue to travel through the network to the next highest priority downstream of the return flow location. This water allocation defines the path that generates the greatest contribution to the objective function. The network path of one unit of water ends when it is either allocated to storage or exits the system. Because NFP has a linear objective function of arc flows, the objective function contribution of a unit of water is a linear function of the decision variables along the path the water takes.

Now consider the algebraic form of the weight generating LP presented in Chapter 2, equations 31 to 35. These equations are repeated here as equations (62) to (66), with X_i denoting the weights, X_1 representing the highest weight and X_N the lowest weight. The first constraint (equation 63) of this LP establishes ranking among weights; the higher the priority, the greater the weight must be. The second constraint (equation 64) relates the weight representing a senior priority, X_p, to the sum of weights representing all downstream junior priorities. The third constraint (equation 65) ensures that the weight representing a senior priority exceeds the objective function accrual of a unit of water being delivered to a junior upstream and its return being then allocated to the downstream senior.

Minimize: $Z = X_1 - X_N$ (62)

Subject to:

$$X_p \geq X_{p+1} + \varepsilon \qquad \forall p = 1,\ldots,N-1 \tag{63}$$

$$X_p \geq (1-a_p)\sum_{j>p}^{K} X_j^{'} + \sum_{j>p}^{L} X_j^{''} + \varepsilon \qquad \forall p = 1,\ldots,N \tag{64}$$

$$X_p \geq \left(\frac{1}{1-a_j}\right) X_j + \varepsilon, \quad \text{for all upstream juniors j; p=1,…,N} \tag{65}$$

$$X_N = \text{Base} \tag{66}$$

As stated in Chapter 2, an infinite number of weight sets (X_1 to X_N) preserve priority. One of these sets can be computed by assigning the lowest priority in the system a baseline weight X_N and choosing a value for ε. The next higher priority, X_{N-1}, can then be found so that equations 63, 64, and 65 are satisfied, moving to the next higher priority until X_1 is computed.

Because a NFP objective function is linear, for each computed weight X_i, equations (63) to (65) guarantee that priority is preserved by ensuring (i) a higher priority weight exceeds weights associated with lower priorities (equation 63), (ii) X_i is priority preserving in relation to all paths a unit of water might take downstream (equation 64) and, (iii) X_i is priority preserving in relation to all junior priorities paths upstream (equation 65). Furthermore, the constraints of a NFP simply route the water in the network and set lower and upper bounds on the decision variables. Therefore, any set of X_i values that satisfy the inequalities provides a priority-preserving set of weights.

If the objective function of the solver used to drive a simulation model includes non-linear terms, the weight set computed with equations (62) to (66) is less likely to be priority preserving, as the objective function contribution of an additional unit of water allocated to a particular decision variable may be disproportional (non-linear) to its coefficient (weight). Consequently, what makes the weight generator algorithm work for an NFP driven simulation is the linearity of the objective function which includes only arc flow decision variables.

PRIORITY PRESERVING WEIGHTS FOR LP DRIVEN SIMULATIONS

In an NFP driven simulation, if the weight set is priority preserving, the allocation of water will occur in order of priority as long as the physical system (continuity and capacity constraints) permits. Examples in Chapter 3 illustrate allocations driven solely by priorities.

NFP problems are a subset of the more general LP problem. While a NFP only has continuity and capacity constraints (Chapter 2, equations (2) and (3), respectively), a LP may include many other linear constraints. These non-NFP constraints are included to simulate some desired change in the way water is allocated. Additional constraints, therefore, constitute a higher layer of priorities that would supersede the priorities represented by weights in the objective function. Consequently, unlike a NFP-driven simulation, a LP may skip allocation to one or more demands in priority according to the priority list as a result of non-NFP constraints. In this section we discuss how additional LP constraints may affect the order of allocation, but, *subject to the linear constraints of the physical system being simulated*, the weight set remains priority preserving.

Types of Variables and Non-NFP Constraints

In a NFP, all decision variables represent physical water flows (arc flow), i.e., actual allocation of water at a particular physical location in space or time (storage). These decision variables appear in the continuity constraints and represent flow in and out of a node and storage. However, a LP may introduce variables that do not represent physical water flows at one location. These variables may be a function of the standard NFP decision variables or a function of state variables or any other variables being simulated. Like the standard NFP decision variables, these LP variables may or may not be weighted, that is, they may or may not appear in the objective function. To simplify the discussion that follows, we will call decision variables representing water flows at one specific network arc, NFP variables (or arc flow) and all other decision variables non-NFP (or non- arc flow) varaibles.

Furthermore, LP constraints are categorized into three types:

> Type A constraints are standard NFP continuity and capacity constraints.
>
> Type B constraints relate two or more decision variables in ways that differ from continuity or bound constraints in NFP. This type of constraint may introduce non-NFP decision variables (non-arc flow variables), but these decision variables have zero weights and so do not appear on the objective function.
>
> Type C constraints also relate two or more decision variables, but introduce non-NFP decision variables that have non-zero weights in the objective function. Type C constraints are required for non-arc flow decision variables to appear in the objective function.

Non-NFP type constraints (types B and C) are included in LP driven models when a NFP priority and mass balance allocation is insufficient to accurately simulate the system. Therefore, by design, these constraints likely will affect the allocation of water so that a junior priority may be allocated water before a senior priority.

In the following section I argue that a weight set computed with the weight generator for arc-flow variables is also priority preserving for LP driven simulations including both types B and C non-NFP constraints, for all the priorities based on the regular NFP variables. I also argue that the objective function coefficient of non-NFP variable should be regarded as a parameter that requires calibration and not a prioritized weight similar to the weights given to the arc-flow variables representing demands or arcs in which flow is to be avoided.

Non-NFP Constraints and Water Allocation

Consider a type B constraint, one which does not include weighted decision variables that are not arc flow (non-NFP type). Consequently, the objective function contains only arc flow decision variables. Type B constraints that prevent allocation to a water user in priority are no different than, for example, capacity constraints in a NFP problem. Once the capacity has been reached the next unit of water will be allocated to the next priority demand. For a type B constraint, the linear objective function is unchanged, and consequently, *subject to the LP constraints*, water is still allocated in order of priority.

Type C constraints introduce weighted, non-arc flow (non-NFP), decision variables. Because these variables are weighted they appear in the objective function and may, therefore, upset the allocation of water. In the discussion that follows I argue that the inclusion of type C constraints does not affect the priority allocation of water among arc flow priorities. As with type B constraints, *subject to the LP constraints*, water is allocated in priority order among arc flow priorities when type C constraints are introduced.

The objective function coefficients of non-NFP decision variables are treated as parameters to be calibrated once weights on arc flow variables have been computed. A step-by-step method for computing coefficients for all decision variables (NFP and non-NFP) is presented later in this chapter in the section “Procedure to Generate Priority Preserving Weights for LP Driven Simulations”.

Consider any optimization problem. The problem constraints define its feasible solution space. Therefore, the constraints are the highest priority of any optimization problem. As constraints are added to the problem the solution space may be reduced. The role of the objective function is to determine, within the solution space defined by the constraints, the optimal solution.

Consider a LP consisting of an objective function (equation 67), NFP type constraints (equations 68 and 69), and one non-NFP linear constraint (equation 70). Assume the weights c_k are priority preserving for a NFP of equations (67) to (69) and c_N is the weight required to ensure that the non-NFP constraint (equation 70) works as desired. The single non-NFP constraint is a linear combination of arc-flow variables.

Maximize: $$Z = \sum_{k=1}^{K} c_k X_k + c_N Y_N \qquad (67)$$

Subject to:
i. mass balance at each node

$$\sum_{k \in Kin} a_k X_k = \sum_{k \in Kon} X_k \qquad \text{for all nodes n = 1, 2, … , N} \qquad (68)$$

ii. upper and lower capacity constraints for each arc

$$0 \le l_k \le X_k \le u_k \qquad \text{for all arcs k = 1, 2, … , K} \qquad (69)$$

iii. the non-NFP constraint

$$Y_N = \sum_{k=1}^{K} d_k X_k \qquad (70)$$

where:

Z = total system penalty
N = number of nodes
K= number of arcs
X_k = flow entering arc k
c_k = weight per unit flow in arc k
a_k = flow multiplier for arc k
K_{in} = arcs flow into node n
K_{on} = arcs flow out of node n
l_k = lower bound flow for arc k
u_k = upper bound flow for arc k
Y_N = non-NFP decision variable
c_N = objective function coefficient for Y_N

Substituting for Y_N in the objective function, the objective function becomes

$$Z = \sum_{k=3}^{K} (c_k + c_N d_k) X_k \quad . \qquad (71)$$

The coefficients of variables x_1 and x_2 have changed, potentially affecting the optimal solution (simulated water allocation). However, the cause of the change in coefficients is the inclusion of the non-NFP constraint. The variable Y_N and its objective function coefficient c_k only exist to make the constraint work. Therefore, Y_N and c_N are integral parts of the constraint (equation 70), and thus part of the definition of the solution space. Consequently, the coefficient of the non-NFP variable Y_N should be regarded as a parameter that requires calibration, and not a prioritized weight similar to the weights given to the arc-flow variables representing demands or arcs in which flow is to be avoided. Hence, the inclusion of any linear constraint to the NFP constraint set does not affect the priority preserving quality of the weight set computed for the arc flow variables, that is, *subject to the problem constraints*, the weight set is priority preserving.

The argument presented above could be generalized for any linear non-NFP constraint, as all non-NFP decision variables can, in theory, be written explicitly as a linear function of the arc-flow (NFP) decision variables. However, it is essential that the non-NFP constraints introduced be linear to ensure the linearity of the resulting objective function.

The following discussion is based on the insights gained with the Two-River System model presented in Chapter 4, and further supported by additional examples in which type B and C constraints are added to a NFP driven simulation and the resulting allocation is examined and found to be priority preserving. It illustrates the theory described above.

In Chapter 4 we discussed using the NFP priority preserving unit cost coefficient generator for a LP driven simulation, the Two-River System model. The Two-River model LP contains several non-NFP type constraints. Most non-NFP constraints in the Two-River System model are type B constraints and, as discussed above, do not affect the priority preserving quality of the weight set.

The Two-River System model also has non-NFP constraints that introduce new non-NFP decision variables into the objective function. These type C non-NFP constraints are mostly associated with Delta operations. Among these constraints are the soft goal constraints used when a target is desired and deviation from the target is penalized with a negative weight. Consider the constraint $\text{D34C} - \text{SURPL0120} + \text{SLACK0120} = 800$, where D34C represents Tracy Pumping Plant (CVP). The variable SURPL0120 is not weighted, but SLACK0120 appears in the objective function with a coefficient of -2000. The purpose of this constraint is to set the minimum desired pumping at the Tracy Pumping Plant to 800 cfs. Rather than setting the minimum pumping as a hard constraint ($\text{D34C} \geq 800$), a soft constraint is used to avoid infeasibilities. The negative weight on the variable SLACK0120 discourages the LP from assigning a value to SLACK0120.

To understand how this constraint and the weighted decision variable may affect water allocation, consider two extreme cases, one in which the magnitude of the weight on SLACK0120 is very large and, when this weight is very small. When the weight's magnitude is very large, the behavior approximates the hard constraint $\text{D34C} \geq 800$, a type B constraint. As discussed above, including type B constraints does not affect the priority preserving quality of a weight set, although, as a constraint, it has the highest priority. At the other extreme, as the weight approaches zero it has minimal effect on the value of the objective function and the allocation of water. Reducing the magnitude of the weight associated with the slack variable approaches the case of not having the criterion at all. In either case, *subject to the constraints*, the LP will allocate water in order of priority according to the generated weight set.

The most complex type C constraints in the Two-River System model represent the Coordinated Operations Agreement (COA). The COA divides, between the CVP and SWP, both the responsibilities for in-basin-use (IBU) of water from storage and the excess water in the system, the unstored-water-for-export (UWFE). If one project cannot export all the water to which it is entitled under the COA, the other project is allowed to take any unused portion (UNUSED_FS and UNUSED_SS). Consider the COA constraints shown in equations (72) to (75).

$$\text{D34B} + \text{C34B\_CVP} + \text{UNUSED\_FS} - \text{CVPDS} + \text{D34C\_EXP1} + 0.75\ \text{IBU} - 0.55\ \text{UWFE} = 0 \qquad (72)$$

$$\text{D34A} + \text{C34B\_SWP} + \text{UNUSED\_SS} - \text{SWPDS} + \text{D34D\_EXP1} + 0.25\ \text{IBU} - 0.45\ \text{UWFE} = 0 \qquad (73)$$

$$-\ \text{UNUSED\_FS} + \text{D34D\_EXP2} <= 0 \qquad (74)$$

$$-\ \text{UNUSED\_SS} + \text{D34C\_EXP2} <= 0 \qquad (75)$$

where:

D34B = CVP demand in Delta
D34A = SWP demand in Delta
C34B_CVP = CVP portion of surplus Delta outflow
C34B_SWP = SWP portion of surplus Delta outflow
UNUSED_FS = Unused Federal share of Delta surplus
UNUSED_SS = Unused State share of Delta surplus
CVPDS = CVP change in storage
SWPDS = SWP change in storage
D34C_EXP1 = CVP export
D34C_EXP2 = CVP export of UNUSED_SS
D34D_EXP1 = SWP export

D34D_EXP2 = SWP export of UNUSED_FS
IBU = Total In-Basin-Uses met with storage withdrawals
UWFE = Total Unstored-Water-For-Export

Equations (72) and (73) are the accounting equations for the various COA components for the CVP and SWP, respectively. Constraint (74) enables the SWP to export the unused Federal share of Delta water and constraint (75) allows the CVP to export the unused State share of water under the COA. These four constraints introduce UNUSED_FS and UNUSED_SS, the weighted non-NFP decision variables representing the Federal and State share of Delta surplus, respectively. These variables and corresponding negative weights are included in the LP to ensure that each project pumps as much as possible under its own COA allowance. The COA constraints are, therefore, type C constraints.

Under the COA, each project can export the other project's unused share of Delta water. Consider equation (6). If there were no competing projects south of the Delta, *UNUSED_FS* would not need to be weighted, as the positive weights on Federal storage and delivery south of the Delta would be enough to ensure the greatest allowable CVP pumping from the Delta. However, without a negative weight on *UNUSED_FS*, the LP will assign water to whichever variable will result in the highest objective function value in a strictly priority preserving fashion. Consider, the situation in which the State portion of San Luis Reservoir (S4) is at its minimum value (i.e., its dead pool is full but all other pools are empty). At this point, if one unit of water is allocated to S4 it will add to the objective function the weight on S4_2. Assume that water allocated to Federal San Luis Reservoir (S3) will be stored in S3_4, which has a lower priority and, therefore, weight than S4_2. Referring to Table 8 in Chapter 4, the DWR weights on S4_2 and S3_4 are 1235 and 65, respectively. If *UNUSED_FS* does not have a negative value sufficient to counter the weights on, in this case, S4_2 and S4_3, a portion (or all) of the CVP share of Delta water will be allocated to the higher priority pools of the State San Luis Reservoir. The weight on *UNUSED_FS* is -1285, which is higher than the highest weight on any priority south of the Delta. Therefore, allowing one project to pump one unit of the other project's water results in a negative accrual to the objective function of -1285+1235=-50, while allocating to the "rightful" owner under the COA increases the objective function by 65. So, while allocating water to the highest priority south of the Delta, S4_2 in this example, would appear to be "priority preserving" it would have "violated" the water sharing criteria of the COA.

The COA constraints, therefore, act as another layer of priority that is higher than the priorities on the various demands within the system. To satisfy the COA constraints the allocation of water is shifted slightly. However, *subject to the model constraints*, the allocation remains priority preserving.

Several test cases are presented below. These test cases illustrate the interplay between the different types of constraints and variables in a LP driven simulation. The test cases demonstrate how, as each type of constraint is introduced, the allocation of water is shifted slightly to accommodate the constraint, but remains priority preserving with respect to the NFP (arc flow) variables.

Test Cases

In the previous section I argue that model constraints constitute a higher priority than those on arc flow. I also make several assertions regarding LP driven simulations, non-NFP constraints, and priority preserving simulations. To examine and test these assertions, test cases consisting of six simulations based on the Two-River System model were designed. The results from these test cases are presented here.

Approach

In Chapter 4 a NFP version of the Two-River System model was created to confirm that the weight sets presented (UCD and DWR) were priority preserving. By stripping the simulation of all non-NFP constraints (constraint types B and C) and considerably simplifying the hydrology we were able to verify that the weight sets used resulted in priority preserving water allocation. In the discussion that follows, refer to Chapter 4, Figure 14.

In this chapter, we build on the Two-River System NFP driven model to test the effects of LP constraints on water allocation. Starting with the NFP driven model of Chapter 4, we progressively add type B and then type C constraints and examine how the allocation of water changes after those constraints are added.

Six runs were created. Each consecutive run builds on the preceding simulation by adding new constraints and decision variables.

- Run I: NFP driven simulation (type A constraints only).
- Run II: Reservoir evaporation is added to Run I (type B constraint).
- Run III: Inflow/export limit on Delta exports is added to Run II (type B constraint).
- Run IVa: Slack/surplus variables to induce a minimum desired pumping at Tracy (D34C) and Banks (D34D) pumping plants is added to Run III (type C constraint, soft constraint).
- Run IVb: Increases the penalty on the slack variables of run IVa to a very large number (type C constraint).
- Run V: Builds upon Run IVa by adding the COA constraints. Weights on slack variables on minimum desired pumping are set to original value as in Run IVa (type C constraint).

Sample LP listings for the first simulated month for runs I to V appear in Appendices VII to XII.

For ease of interpreting simulation results, several Two River System state variables and capacities were simplified, as follows:

1. Initial storage for all reservoirs was set to the full dead pool (S1 = 550 taf, S2 = 29.6 taf, S3 = 45 taf, and S4 = 55 taf)
2. Inflows to the system were set to a regular pattern with I1(t)=1000+500*(t-1) and I2(t)=250+125*(t-1), were t represents the period of simulation. This steadily increases the inflows so prioritization of water allocation and use can be examined.

3. Evaporation was removed to create Run I and reinstated in Run II and subsequent runs.
4. All diversion demands, including south of the Delta (D3 and D4) were set to a constant value of 1000 cfs. As with the original Two-River System model, D33 is set to zero.
5. San Luis Reservoir pool sizes were set to constant values (S3_1 = 45, S3_2 = 0, S3_3 = 455, S3_4 = 450, S3_5 = 22, S4_1 = 55, S4_2 = 0, S4_3 = 445, S4_4 = 500, and S4_5 = 67, for total capacity of S3 = 972 taf and S4 = 1067 taf)

Results

Table 17 presents the timing of water allocation for the six runs. Because several demands have the same priority, it is best to analyze the order of allocation by looking at the timing of allocation for each group of demands that have the same priority. Table 17 shows the month in which each demand starts being met and the month in which the demand is fully met.

To interpret the results in Table 17, consider the Feather River demands in run I, the NFP driven simulation. In run I, C2_MIF starts being met in the first month and is fully met in month 7. D2 starts being met in month 8 and is fully met in month 15. Between months 15 and 22, the incremental inflow into the Feather River is used to meet higher priority demands in the Delta and south of the Delta. Once S3_3 and S4_3 (priority 5) are full in month 19, S1_2 (priority 6) starts filling, after which S2_2 (priority 7) starts accruing water. Once S2_2 is full the other pools of S2 start filling in turn.

A similar analysis can be made on the Sacramento River. The minimum instream flow requirement at C30 (C30_MIF) is fully met in the first month, demands with priority 3 are met between the first and sixth month, after which priority 4 demands D3 and D4 are met by month 11. Priority 5 S3_3 and S4_3 starts being filled in the twelfth month and are fully met by month 19. After that, all other demands are filled in order of priority.

Table 17. Month in which demand starts being met and is fully met.

		TEST CASE RUNS											
	Run:	**I - NFP**		**II - Evaporation**		**III - Export Ratio**		**IVa - Soft constraints**		**IVb - Soft constraints**		**V - COA**	
	Constraint:	**Type A**		**Type B**		**Type B**		**Type C**		**Type C**		**Type C**	
Priority	**Demand**	Begin	Full	Begin	Full	Begin	Full	Begin	Full	Begin	Full	Begin	Full
1	S1_1	1	1	1	1	1	1	1	1	1	1	1	1
1	S2_1	1	1	1	1	1	1	1	1	1	1	1	1
1	S3_1	1	1	1	1	1	1	1	1	1	1	1	1
1	S4_1	1	1	1	1	1	1	1	1	1	1	1	1
2	C2_MIF	1	7	1	8	1	8	1	8	1	8	1	8
2	C30_MIF	1	1	1	2	1	2	1	2	1	2	1	2
3	D2	8	15	8	15	8	15	8	15	8	15	8	15
3	D30	2	3	2	4	2	4	2	8	3	4	2	8
3	D31	1	1	1	2	1	2	1	2	4	7	6	7
3	D34A	1	5	4	6	1	4	1	4	3	8	1	7
3	D34B	2	2	1	4	4	7	4	6	1	7	1	4
3	C34A	5	7	6	7	6	6	6	7	4	10	4	6
4	D3	10	11	9	10	8	11	8	11	1	10	7	11
4	D4	7	9	7	12	7	12	7	12	1	12	7	13
5	S3_3	12	19	16	18	16	22	16	23	16	23	11	24
5	S4_3	13	19	12	19	12	24	12	24	12	24	13	24
6	S1_2	19	22	19	22	17	21	17	21	17	21	15	22
7	S2_2	22	31	22	31	21	31	21	31	21	31	17	30
8	S1_3	22	25	23	25	21	25	21	25	21	25	22	26
9	S2_3	31	41	31	41	31	41	31	41	31	41	30	41
10	S1_4	25	27	25	27	25	27	25	27	25	27	26	29
11	S2_4	41	44	41	44	41	44	41	44	41	44	41	43
12	S3_4	24	29	24	29	27	29	27	29	27	29	28	31
13	S1_5	27	28	28	29	28	29	28	29	28	29	29	30
14	S4_4	28	30	29	30	29	32	29	32	29	32	24	28
15	S2_5	44	45	44	45	44	45	44	45	44	45	43	44

As the non-NFP constraints are incrementally introduced to each run, a shift occurs in the timing of fulfillment of each priority, as each new constraint becomes a priority higher than those assigned to the various demands.

Upstream of node 32 (see Figure 14, Chapter 4), demands can only be met by water from one source. Demands on the Sacramento River upstream of node 32 can only be met by inflow into Shasta Reservoir (S1) and demands on the Feather River can only be met by inflow into Oroville Reservoir (S2). Therefore, to ease interpreting results, therefore, demands that can only be met from inflow into S2 are highlighted in the table. Once the various LP constraints are introduced in runs II to V, they become, together with existing constraints, the highest priority. Each new constraint shifts slightly the timing of filling demands. A full listing of constraints for run I appears in Appendix B-1.

Constraint Type B: Evaporation

Run II introduces reservoir evaporation (at all reservoirs) through LP constraints such as those listed as equations (76) and (77) and the term E1 in the continuity equation for the reservoir S1 (equation 78). No weights are associated with the non-NFP variables A1 (reservoir S1 surface area) and E1 (evaporation from S1) that constraints (76) and (77) introduce. Therefore, constraints (76) and (77) are type B constraints. A full listing of constraints for run II is shown in Appendix B-2.

$$- 8.91348\ S1 + A1 = 2099.39 \quad (67)$$

$$61.4876\ E1 - 0.220781\ A1 = 1545.86 \quad (68)$$

$$- F1 - C1 - 16.2634\ S1 - E1 = -9944.89 \quad (69)$$

In run II, when evaporation is introduced, the highest priority associated with the dead pools is tested. While in run I, the entire inflow into S1 can be released to meet the minimum instream flow at C3 (C30_MIF), in all other runs, the release from S1 is reduced by the volume evaporated from S1. Results presented in Table 18 reflect this. The dead pools of all reservoirs remain full in all runs, which means water loss through evaporation from the dead pools is immediately replaced. In the first six months of run I, Delta exports D34C and D34D are zero. In all other runs the exact volumes that evaporate from S3 and S4 are pumped at D34C and D34D, respectively, (Table 18) in the first six months.

The evaporation constraints shift the timing of allocation slightly. However, including the non-NFP constraints to compute evaporation did not affect the priority preserving allocation, as water is allocated to always to keep higher priority reservoir pools full.

Constraint Type B: Export/Inflow Ratio

Run III introduces the inflow export ratio constraints shown in equations (79) to (83). A full listing of constraints for run III is shown in Appendix B-3. These constraints introduce new decision variables defined by the constraints. Because these new decision variables do not appear in the objective function, the export/inflow constraints fall in the type B constraint category.

$$- D34C - D34D + EXPORTACTUAL = 0 \quad (79)$$

$$- C33 + INFLOW = 0 \quad (80)$$

$$EXPRATIO\_ = 0.65 \quad (81)$$

- 0.65 INFLOW + EIEXPCTRL = 0 (82)

EXPORTACTUAL - EIEXPCTRL <= 0 (83)

To understand how the export/inflow constraints affect the allocation of water, consider the simulated allocations and export/inflow ratio (EI) for months 13 to 27 presented on Table 19. The export/inflow constraints limit the Delta exports (D34C+D34D) to *EI * inflow*. The remaining *(1-EI) * inflow* can either meet other Delta demands (D34A, C34A and D34B) or be assigned to surplus Delta outflow (C34B). While the total demand within the Delta is at least *(1-EI) * inflow*, no surplus outflow will occur. That is, as long as the Delta inflow (C33) is less than the combined Delta demand divided by *(1-EI)*, there will be no surplus Delta outflow.

The EI ratio is set to 0.65 for August through January, 0.35 for March through July, and varies between 0.35 and 0.45 in February. Therefore, the inflow into the Delta needs to be less than 3000/(1-0.65) = 8,571 cfs August through January, and 3000/(1-0.35)=4,615 cfs March through July to avoid water being allocated to surplus outflow (C34B). As presented in Table 3, in run II, the inflow to the Delta exceeds these values in months 17 through 21. Because of the export/inflow constraints in run III, the inflow to the Delta, C33, is capped at 4,615 cfs, which results in earlier allocation of water to S1_2 (tables 17 and 19) than in run II. Consequently, to satisfy the export/inflow constraint, allocation to higher priority demands south of the Delta is delayed, and water is allocated to a lower priority demand (S1_2). Once again, there is a slight shift in the timing of allocation of water, but, subject to the constraints, the available water in the system is still allocated according to priority.

Constraint Type C: Soft Constraint, Minimum Desired Pumping

Runs IVa and IVb introduce the soft constraints (constraint type C) listed as equations (84) and (85). The slack variables are given weights -2000 and -2,000,000 in runs IVa and IVb, respectively. LP listings for runs IVa and IVb are presented in appendices B-4 and B-5, respectively.

D34C - SURPL0126 + SLACK0126 = 800 (84)

D34D - SURPL0127 + SLACK0127 = 300 (85)

Table 18 presents the simulated water allocation for priorities 2 and 3. While the combined monthly pumping at C34C and C34D is the same in runs III and IVa, the distribution of water is affected by the minimum desired pumping constraints (equations 84 and 85). Before the seventh month the pumping at D34C and D34D match the volume of water evaporated from the dead pool at S3 and S4. Between the seventh and ninth month water becomes available for export from the Delta, but is insufficient to meet both minimum desired pumping values. In run III, without the minimum desired pumping constraint, no attempt is made to meet the minimum desired pumping. However, in run IVa, as soon as the water available for pumping exceeds 300 cfs (month 8), the pumping shifts to C34D, where it meets the minimum desired flow of 300 cfs. In run IVb, where, the weight on the slack variables is very large, the soft constraint effectively becomes a hard constraint and the minimum desired pumping rates are met in all months.

Differences in water allocation in runs III, IVa and IVb support the assertions that, depending on the weight associated with the slack/surplus variable, this type of soft constraint acts as a hard constraint when the weight is high (run IVb), has minimal effect on allocation when the weight is low, and satisfies the minimum pumping to varying degree when the weight is somewhere in between (run IVa).

Constraint Type C: COA

As described in the section titled "LP Constraints and Water Allocation", the COA constraints are designed to simulate the sharing of Delta water between the State (SWP) and Federal (CVP) projects. The COA introduces 16 constraints to run V. The COA constraints are the last 16 constraints listed in Appendix B-6. Of relevance here are the type C constraints that introduce the non-arc flow decision variables UNUSED_SS and UNUSED_FS. These constraints are repeated here as equations (86) to (90).

$$- D34A - D34B - C34B\_CVP - C34B\_SWP - UNUSED\_FS - UNUSED\_SS + SWPDS + CVPDS - D34C\_EXP1 - D34D\_EXP1 - IBU + UWFE = 0 \quad (86)$$

$$D34B + C34B\_CVP + UNUSED\_FS - CVPDS + D34C\_EXP1 + 0.75\ IBU - 0.55\ UWFE = 0 \quad (87)$$

$$D34A + C34B\_SWP + UNUSED\_SS - SWPDS + D34D\_EXP1 + 0.25\ IBU - 0.45\ UWFE = 0 \quad (88)$$

$$- UNUSED\_FS + D34D\_EXP2 <= 0 \quad (89)$$

$$- UNUSED\_SS + D34C\_EXP2 <= 0 \quad (90)$$

Without the COA in the simulation, water in the Delta is distributed between the CVP and SWP pumps according to the weights on demands south of the Delta and the minimum desired pumping constraints. Once the COA is introduced, water is distributed according to the COA, minimum desired pumping and weights. This more even distribution of water pumped between the two projects can be seen by comparing the pumping rates (D34C and D34D) between runs IVa and V shown in Table 18. The same results are also shown in Table 17, where the filling of the south of the Delta reservoir pools S3_3 and S4_3 start at approximately the same time (months 11 and 13, respectively). In run IVa, with a less even distribution of pumping, the filling of S3_3 and S4_3 start at months 12 and 16, respectively.

PROCEDURE TO GENERATE PRIORITY PRESERVING WEIGHTS FOR LP DRIVEN SIMULATIONS

As discussed in chapters 6 and 7 the automated weight generator pre-processor described in Chapter 2 cannot compute all the weights required for general LP driven simulations. However, the automated weight generator can compute priority based weights on all arc-flow (NFP) variables where flow is either desired (positive weights), or where flow is to be avoided (negative weights). The only exception is for arcs (usually, but not exclusively, an outflow or sink) for which water allocation is not desired (negative weight) but receives water because of water allocation to a positive weighted priority or because of non-NFP constraints (see Chapter 6). Usually, objective function variables are NFP variables. Weights for non-NFP variables that appear in the objective function typically must be determined by manual calibration. The remainder of this section describes the procedure for computing weights for LP driven simulations. A summary of these steps is presented in Table 18.

Table 18. Weight Generating Procedure Steps.

STEP	ACTION
1	Define network connectivity and return flow factors
2	Sort weighted decision variables by type
3	Prioritize arc-flow decision variables
4	Compute positive weights on arc-flow decision variables
5	Temporarily set negative weights on arc-flow decision variables
6	Manually calibrate negative weights for non-NFP decision variables
7	Compute final set of negative weights on arc-flow decision variables
8	Test final weight set

<u>Step 1: Define Network Connectivity and Return Flow Factors</u>

To compute weights for a LP driven simulation it is necessary to generate, from the model database, the location connectivity matrix **M,** representing the networks connectivity (Chapter 2). Each row or column of **M** is related to a node within the network. To define the spatial connections between priorities and the location of return flows, it is sometimes necessary to include nodes in **M** that are not linked to any priority for water allocation. Connections include: (i) arcs of return flows and (ii) return flow factor for all connections included in **M**.

<u>Step 2: Sort Weighted Decision Variables by Type.</u>

Weighted decision variables should be sorted into distinct classes: (a) those representing arc-flow (or NFP variables) where water is desired, (b) arc-flow where water is not desired, and (c) non-NFP decision variables. The arc-flow variables where water is desired will be given positive weights and the arc-flow variables where water is not desired will be given negative weights (for a maximization problem in both cases).

Within a network, arcs may exist where flow is not desired but necessarily receive water from other locations where flow is desired. These are often sink arcs and the "conflict" arises either from return flows or non-NFP constraints. These "gray area" arc-flow variables must be identified and will be handled individually.

Weighted non-NFP decision variables can be broadly categorized into two classes: (i) those that make a constraint or constraint set work as desired to reflect legal or operational constraints (e.g., COA) and (ii) those that reflect the degree of hardness of a soft constraint (slack or surplus variables).

<u>Step 3: Prioritize Arc-Flow Decision Variables in two Vectors.</u>

Prioritize arc-flow variables separately for scarce and excess flow conditions. To compute weights on arc-flow variables using the automated procedure (Chapter 2), a priority vector corresponding to matrix **M** should be created. The ith entry on each priority vector contain the priority corresponding ith row (or column) of matrix **M**. One vector should contain the priorities of arc-flow variables where water is desired, with all other entries being zero. The second priority vector should contain the priorities of arc-flow variables where water is not desired.

Step 4: Compute Positive Weights on Arc-Flow Variables.

Use the automated procedure described in Chapters 2 and 3 to compute positive weights. To avoid potential scaling problems between positive and negative weights, the lowest difference between consecutive weights (ε) should be set to at least 10.

Scaling between positive and negative weights is not necessary if there is no interaction between positive and negative weights. Without interaction between positive and negative weights, negative weights can be computed in the same way as the positive weights, using the same matrix **M** and the vector containing priorities for avoidance of water. Interaction between positive and negative weights is likely to occur when the LP includes weighted non-NFP variables or arcs to which water allocation is not desired (negative weight) but receive water because of water allocation to a positive weighted priority (see Chapter 6). If scaling is needed, additional steps are required (steps 6 and 7).

Step 5: Temporarily Set Negative Weights on Arc-Flow Variables.

To scale positive and negative weight sets, temporarily assume a set of negative weights for the arc flow variables, including arcs with conflicting priorities as described in Chapter 6. At this point, the traditional way of choosing weights that are one order of magnitude different for the distinct priorities is usually adequate. Alternatively, the automated procedure could be used with large ε and large baseline negative weight.

Step 6: Manually Calibrate Negative Weights for non-NFP Decision Variables.

The next step is to select the non-NFP variable weights that result in simulations best reflecting the legal or operational criteria the non-NFP constraints represent. For the Two-River System model discussed in Chapter 4, the variables UNUSED_SS and UNUSED_FS would fall in this category. These variables must balance the positive weights on demands south of the Delta to ensure that the COA constraints truly reflect the COA. As the non-NFP constraints which introduce weighted non-NFP variables may be complex, the relationship between the various variables and how they play out in the simulation may not be apparent, and manual calibration may be required. Careful manual calibration should clarify any existing relationships.

Non-NFP constraints that reflect desired policies or operational criteria (soft constraints) usually include weighted non-NFP variables. The degree of hardness of these constraints will be given by the weight given to those variables. These weights should also be determined by manual calibration.

Step 7: Compute Final Set of Negative Weights on Arc-Flow Variables.

Using the automated procedure (Chapter 2), compute the final set of negative weights on arc-flow variables. The values of ε and the lowest priority negative weight (baseline weight) should be set so negative weights obtained through manual calibration for arc-flow variables fit within the final computed set of negative weights. In Two-River System model, this applies to the sink arc-flow variables C34B_SWP and C34B_CVP (see chapter 4), which are the lowest priority on arc-flow variables where allocation of water is not desired.

Step 8: Test Final Weight Set.

Weights obtained in steps 1-7 should be tested in a variety of water abundant and water scarce conditions. Tests similar to those described in Chapter 7 can help ascertain that the weight set is priority preserving. By setting demands to constant values and progressively increasing inflow into the system, it is possible to verify that demands are met in order of priority. In addition, the procedure of adding non-NFP constraints to a NFP-driven version of the model is helpful in teasing out the effects of non-NFP constraints and their weights. This process is illustrated in Chapter 7.

CONCLUSIONS

In this chapter I argue that the introduction of non-NFP constraints to the constraint set and non-arc flow variables to the objective function does not affect the priority preserving quality of the weight set. This claim is supported by a qualitative discussion of the structure of the objective function and constraints of both a NFP and a LP, and that, the constraints are nothing more than a higher level of priorities. This is further supported by test case results for a simplified California system here and success applying the priority weighting method by Brown (2005) to a LP flood control problem in Iowa.

In this chapter I also argued that the weight generator procedure presented in Chapter 2 is priority preserving for NFP driven simulations due to the linearity of the objective function. LP constraints were split into three types (A, B, and C), and analyzed separately regarding their possible effects on water allocation and preservation of operational priorities.

Type A constraints are the typical NFP continuity and capacity constraints. Type A constraints only include variables that represent arc flow. Type B constraints may or may not introduce non-NFP variables (non-arc flow variables). These non-NFP variables are not weighted and therefore do not appear in the objective function. Type C constraints include weighted non-NFP variables. This type of constraint, therefore, introduces new variables to the objective function. However, these non-NFP variables should be considered parameters that require calibration, and not a prioritized weight.

The test cases presented in this chapter support the conjecture that a priority preserving set of weights for a NFP driven simulation model remains priority preserving once more general LP constraints are included. Including non-NFP constraints affects the order of allocation where those particular constraints interact with the system being modeled. Non-NFP constraints affect the allocation of water; that is why they are included in the constraint set. However, *subject to the LP constraints*, the allocation of water appears to be priority preserving for a more general LP if the weight set is priority preserving for a NFP driven model.

Model constraints act as an additional layer of priorities which, by design, may override the priorities assigned to the various demands within the system. In other words, two sets of conditions ensure that a simulation is accurate and thus priority preserving. First and foremost, all physical, regulatory, and institutional constraints embodied in the LP constraint set must be satisfied. These constraints are the highest priority to be met. For every problem an infinite number of solutions (water allocations) satisfy the physical, regulatory, and institutional constraints. Within this solution space, a subspace (or one single solution in the absence of multiple optima) exists in which water is allocated according to the priorities associated with every demand. Priorities associated with each water demand in the system are, therefore, a lower rank priority than the LP constraints. Consequently, as long as the objective function is linear, neither the inclusion of non-NFP constraints nor the introduction of non-arc flow decision variables in the objective function upset the priority preserving quality of a weight set that is priority preserving under NFP constraints for arc flow-based priorities.

Finally a generalized overall approach to establishing priority preserving weights is presented. This overall approach addresses the range of issues examined in this dissertation.

Table 19. Simulated Allocation for Priorities 2 to 4.

Month	RUN I											
	D30 CFS	C30 min CFS	D31 CFS	C2 min CFS	D2 CFS	D34A CFS	C34A CFS	D34B CFS	D34C CFS	D3 CFS	D34D CFS	D4 CFS
1	0	1,000	1,000	257	0	257	0	0	0	0	0	0
2	500	1,000	375	375	0	0	0	1,000	0	0	0	0
3	1,000	1,000	500	500	0	0	0	1,000	0	0	0	0
4	1,000	1,000	1,000	625	0	125	0	1,000	0	0	0	0
5	1,000	1,000	0	750	0	1,000	750	1,000	0	0	0	0
6	1,000	1,000	1,000	875	0	1,000	375	1,000	0	0	0	0
7	750	3,250	1,000	1,000	0	1,000	1,000	1,000	0	0	250	250
8	1,000	3,250	1,000	1,000	125	1,000	1,000	1,000	0	0	500	500
9	1,000	3,250	1,000	1,000	250	1,000	1,000	1,000	0	0	1,000	1,000
10	1,000	3,250	1,000	1,000	375	1,000	1,000	1,000	500	500	1,000	1,000
11	1,000	3,250	1,000	1,000	500	1,000	1,000	1,000	1,000	1,000	1,000	1,000
12	1,000	4,500	1,000	1,000	625	1,000	1,000	1,000	1,500	1,000	1,000	1,000
13	1,000	2,750	1,000	1,000	750	1,000	1,000	1,000	516	1,000	2,484	1,000
14	1,000	1,000	1,000	1,000	875	1,000	1,000	1,000	1,000	1,000	2,500	1,000
15	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	3,000	1,000

Month	RUN II											
	D30 CFS	C30 min CFS	D31 CFS	C2 min CFS	D2 CFS	D34A CFS	C34A CFS	D34B CFS	D34C CFS	D3 CFS	D34D CFS	D4 CFS
1	0	950	950	254	0	0	0	217	17	0	20	0
2	479	1,000	1,000	377	0	0	0	368	4	0	5	0
3	984	1,000	1,000	503	0	0	0	502	0	0	0	0
4	1,000	1,000	1,000	629	0	116	0	1,000	0	2	0	3
5	1,000	1,000	1,000	753	0	735	0	1,000	1	0	1	0
6	1,000	1,000	1,000	876	0	1,000	336	1,000	7	0	8	0
7	709	3,250	1,000	999	0	1,000	1,000	1,000	17	0	232	212
8	1,000	3,250	1,000	1,000	121	1,000	1,000	1,000	31	0	398	362
9	1,000	3,250	1,000	1,000	244	1,000	1,000	1,000	864	824	46	0
10	1,000	3,250	1,000	1,000	368	1,000	1,000	1,000	1,046	1,000	341	289
11	1,000	3,250	1,000	1,000	493	1,000	1,000	1,000	1,040	1,000	862	817
12	1,000	4,500	1,000	1,000	619	1,000	1,000	1,000	1,032	1,000	1,384	1,000
13	1,000	2,750	1,000	1,000	748	1,000	1,000	1,000	1,016	1,000	1,945	1,000
14	1,000	1,000	1,000	1,000	877	1,000	1,000	1,000	1,003	1,000	2,480	1,000
15	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	2,987	1,000

Month	RUN III											
	D30 CFS	C30 min CFS	D31 CFS	C2 min CFS	D2 CFS	D34A CFS	C34A CFS	D34B CFS	D34C CFS	D3 CFS	D34D CFS	D4 CFS
1	0	950	950	254	0	217	0	0	17	0	20	0
2	479	1,000	1,000	377	0	368	0	0	4	0	5	0
3	984	1,000	1,000	503	0	502	0	0	0	0	0	0
4	1,000	1,000	1,000	629	0	1,000	0	116	0	2	0	3
5	1,000	1,000	1,000	753	0	1,000	0	735	1	0	1	0
6	1,000	1,000	1,000	876	0	1,000	1,000	336	7	0	8	0
7	709	3,250	1,000	999	0	1,000	1,000	1,000	17	0	232	212
8	1,000	3,250	1,000	1,000	121	1,000	1,000	1,000	393	362	35	0
9	1,000	3,250	1,000	1,000	244	1,000	1,000	1,000	864	824	46	0
10	1,000	3,250	1,000	1,000	368	1,000	1,000	1,000	335	289	1,052	1,000
11	1,000	3,250	1,000	1,000	493	1,000	1,000	1,000	1,040	1,000	862	817
12	1,000	4,500	1,000	1,000	619	1,000	1,000	1,000	1,032	1,000	1,384	1,000
13	1,000	2,750	1,000	1,000	748	1,000	1,000	1,000	1,016	1,000	1,945	1,000
14	1,000	1,000	1,000	1,000	877	1,000	1,000	1,000	1,003	1,000	2,480	1,000
15	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	2,987	1,000

Table 19. Simulated Allocation for Priorities 2 to 4 (contd).

Month	RUN IVa											
	D30 CFS	C30 min CFS	D31 CFS	C2 min CFS	D2 CFS	D34A CFS	C34A CFS	D34B CFS	D34C CFS	D3 CFS	D34D CFS	D4 CFS
1	0	950	950	254	0	217	0	0	17	0	20	0
2	479	1,000	1,000	377	0	368	0	0	4	0	5	0
3	984	1,000	1,000	503	0	502	0	0	0	0	0	0
4	1,000	1,000	1,000	629	0	1,000	0	116	0	2	0	3
5	1,000	1,000	1,000	753	0	1,000	0	735	1	0	1	0
6	1,000	1,000	1,000	876	0	1,000	336	1,000	7	0	8	0
7	709	3,250	1,000	999	0	1,000	1,000	1,000	17	0	232	212
8	1,000	3,250	1,000	1,000	121	1,000	1,000	1,000	129	98	300	265
9	1,000	3,250	1,000	1,000	244	1,000	1,000	1,000	610	570	300	254
10	1,000	3,250	1,000	1,000	368	1,000	1,000	1,000	800	754	587	535
11	1,000	3,250	1,000	1,000	493	1,000	1,000	1,000	1,040	1,000	862	817
12	1,000	4,500	1,000	1,000	619	1,000	1,000	1,000	1,032	1,000	1,384	1,000
13	1,000	2,750	1,000	1,000	748	1,000	1,000	1,000	1,016	1,000	1,945	1,000
14	1,000	1,000	1,000	1,000	877	1,000	1,000	1,000	1,003	1,000	2,480	1,000
15	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	2,987	1,000

Month	RUN Ivb											
	D30 CFS	C30 min CFS	D31 CFS	C2 min CFS	D2 CFS	D34A CFS	C34A CFS	D34B CFS	D34C CFS	D3 CFS	D34D CFS	D4 CFS
1	0	956	0	736	0	0	0	592	800	783	300	280
2	0	1,000	0	220	0	0	0	592	800	796	300	295
3	463	1,000	0	171	0	592	0	0	800	800	300	300
4	1,000	1,000	424	629	0	0	592	0	800	802	300	303
5	1,000	1,000	237	753	0	1,000	0	400	800	799	300	299
6	1,000	1,000	208	876	0	1,000	1,000	43	800	793	300	292
7	709	3,250	1,000	999	0	149	1,000	1,000	800	783	300	281
8	1,000	3,250	1,000	1,000	121	1,000	329	1,000	800	769	300	265
9	1,000	3,250	1,000	1,000	244	1,000	810	1,000	800	760	300	254
10	1,000	3,250	1,000	1,000	368	1,000	1,000	1,000	1,046	1,000	341	289
11	1,000	3,250	1,000	1,000	493	1,000	1,000	1,000	1,040	1,000	862	817
12	1,000	4,500	1,000	1,000	619	1,000	1,000	1,000	1,032	1,000	1,384	1,000
13	1,000	2,750	1,000	1,000	748	1,000	1,000	1,000	1,016	1,000	1,945	1,000
14	1,000	1,000	1,000	1,000	877	1,000	1,000	1,000	1,003	1,000	2,480	1,000
15	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	2,987	1,000

Month	RUN V											
	D30 CFS	C30 min CFS	D31 CFS	C2 min CFS	D2 CFS	D34A CFS	C34A CFS	D34B CFS	D34C CFS	D3 CFS	D34D CFS	D4 CFS
1	0	950	0	254	0	547	0	620	17	0	20	0
2	479	1,000	0	377	0	625	0	743	4	0	5	0
3	984	1,000	0	503	0	685	0	818	0	0	0	0
4	1,000	1,000	0	629	0	833	283	1,000	0	2	0	3
5	1,000	1,000	0	753	0	833	902	1,000	1	0	1	0
6	1,000	1,000	498	876	0	838	1,000	1,000	7	0	8	0
7	709	3,250	1,000	999	0	1,000	1,000	1,000	219	202	30	10
8	1,000	3,250	1,000	1,000	121	1,000	1,000	1,000	306	275	123	87
9	1,000	3,250	1,000	1,000	244	1,000	1,000	1,000	610	570	300	254
10	1,000	3,250	1,000	1,000	368	1,000	1,000	1,000	816	770	571	519
11	1,000	3,250	1,000	1,000	493	1,000	1,000	1,000	1,105	1,000	796	751
12	1,000	4,500	1,000	1,000	619	1,000	1,000	1,000	1,394	1,000	1,022	985
13	1,000	2,750	1,000	1,000	748	1,000	1,000	1,000	1,711	1,000	1,249	1,000
14	1,000	1,000	1,000	1,000	877	1,000	1,000	1,000	2,008	1,000	1,476	1,000
15	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	2,264	1,000	1,703	1,000

Table 20. Simulated Allocation for Priorities 5 and greater.

Month	RUN I											
	S1_2 TAF	S1_3 TAF	S1_4 TAF	S1_5 TAF	S2_2 TAF	C33 CFS	C34B CFS	EI	S3_3 TAF	S3_4 TAF	S4_3 TAF	S4_4 TAF
13	0	0	0	0	0	6,000	0	0.65	0	0	91	0
14	0	0	0	0	0	6,500	0	0.65	0	0	180	0
15	0	0	0	0	0	7,000	0	0.65	0	0	303	0
16	0	0	0	0	0	7,625	0	0.65	20	0	445	0
17	0	0	0	0	0	8,250	0	0.35	220	0	426	0
18	0	0	0	0	0	8,875	0	0.35	441	0	442	0
19	251	0	0	0	0	5,275	0	0.35	455	0	445	0
20	567	0	0	0	0	5,000	0	0.35	455	0	445	0
21	909	0	0	0	0	5,000	0	0.35	455	0	445	0
22	1,165	82	0	0	54	5,000	0	0.65	455	0	445	0
23	1,165	451	0	0	115	5,000	0	0.65	455	0	445	0
24	1,165	778	0	0	182	6,000	0	0.65	455	60	445	0
25	1,165	785	483	0	259	4,032	0	0.65	455	0	445	0
26	1,165	785	929	0	341	5,000	0	0.65	455	0	445	0
27	1,165	785	1,100	100	433	8,600	0	0.65	455	221	445	0

Month	RUN II											
	S1_2 TAF	S1_3 TAF	S1_4 TAF	S1_5 TAF	S2_2 TAF	C33 CFS	C34B CFS	EI	S3_3 TAF	S3_4 TAF	S4_3 TAF	S4_4 TAF
13	0	0	0	0	0	5,960	0	0.65	0	0	77	0
14	0	0	0	0	0	6,484	0	0.65	0	0	165	0
15	0	0	0	0	0	6,987	0	0.65	0	0	287	0
16	0	0	0	0	0	7,615	0	0.65	222	0	227	0
17	0	0	0	0	0	8,233	0	0.35	421	0	206	0
18	0	0	0	0	0	8,844	0	0.35	455	0	406	0
19	220	0	0	0	0	5,757	0	0.35	455	0	445	0
20	519	0	0	0	0	5,164	0	0.35	455	0	445	0
21	841	0	0	0	0	5,192	0	0.35	455	0	445	0
22	1,165	0	0	0	38	5,245	0	0.65	455	0	445	0
23	1,165	342	0	0	99	5,219	0	0.65	455	0	445	0
24	1,165	657	0	0	165	6,000	0	0.65	455	50	445	0
25	1,165	785	340	0	241	4,279	0	0.65	455	0	445	0
26	1,165	785	779	0	323	5,030	0	0.65	455	0	445	0
27	1,165	785	1,100	0	417	7,733	0	0.65	455	168	445	0

Month	RUN III											
	S1_2 TAF	S1_3 TAF	S1_4 TAF	S1_5 TAF	S2_2 TAF	C33 CFS	C34B CFS	EI	S3_3 TAF	S3_4 TAF	S4_3 TAF	S4_4 TAF
13	0	0	0	0	0	5,960	0	0.65	0	0	77	0
14	0	0	0	0	0	6,484	0	0.65	0	0	165	0
15	0	0	0	0	0	6,987	0	0.65	0	0	287	0
16	0	0	0	0	0	7,615	0	0.65	222	0	227	0
17	201	0	0	0	0	4,615	0	0.35	256	0	171	0
18	460	0	0	0	0	4,615	0	0.35	292	0	108	0
19	746	0	0	0	0	4,615	0	0.35	326	0	47	0
20	1,077	0	0	0	0	4,615	0	0.35	343	0	0	0
21	1,165	222	0	0	44	4,615	0	0.35	312	0	0	0
22	1,165	324	0	0	97	8,571	0	0.65	455	0	66	0
23	1,165	458	0	0	157	8,571	0	0.65	455	0	274	0
24	1,165	652	0	0	223	8,039	0	0.65	455	0	445	0
25	1,165	785	286	0	299	5,085	0	0.65	455	0	445	0
26	1,165	785	725	0	382	5,030	0	0.65	455	0	445	0
27	1,165	785	1,100	0	475	6,852	0	0.65	455	114	445	0

Table 20. Simulated Allocation for Priorities 5 and greater (contd).

Month	RUN IVa											
	S1_2 TAF	S1_3 TAF	S1_4 TAF	S1_5 TAF	S2_2 TAF	C33 CFS	C34B CFS	EI	S3_3 TAF	S3_4 TAF	S4_3 TAF	S4_4 TAF
13	0	0	0	0	0	5,960	0	0.65	0	0	77	0
14	0	0	0	0	0	6,484	0	0.65	0	0	165	0
15	0	0	0	0	0	6,987	0	0.65	0	0	287	0
16	0	0	0	0	0	7,615	0	0.65	204	0	244	0
17	201	0	0	0	0	4,615	0	0.35	193	0	234	0
18	460	0	0	0	0	4,615	0	0.35	179	0	221	0
19	746	0	0	0	0	4,615	0	0.35	165	0	208	0
20	1,077	0	0	0	0	4,615	0	0.35	150	0	192	0
21	1,165	222	0	0	44	4,615	0	0.35	164	0	146	0
22	1,165	324	0	0	97	8,571	0	0.65	379	0	139	0
23	1,165	458	0	0	157	8,571	0	0.65	455	0	271	0
24	1,165	649	0	0	223	8,085	0	0.65	455	0	445	0
25	1,165	785	283	0	299	5,085	0	0.65	455	0	445	0
26	1,165	785	722	0	382	5,030	0	0.65	455	0	445	0
27	1,165	785	1,100	0	475	6,809	0	0.65	455	111	445	0

Month	RUN IVb											
	S1_2 TAF	S1_3 TAF	S1_4 TAF	S1_5 TAF	S2_2 TAF	C33 CFS	C34B CFS	EI	S3_3 TAF	S3_4 TAF	S4_3 TAF	S4_4 TAF
13	0	0	0	0	0	5,960	0	0.65	0	0	77	0
14	0	0	0	0	0	6,484	0	0.65	0	0	165	0
15	0	0	0	0	0	6,987	0	0.65	0	0	287	0
16	0	0	0	0	0	7,615	0	0.65	204	0	244	0
17	201	0	0	0	0	4,615	0	0.35	193	0	234	0
18	460	0	0	0	0	4,615	0	0.35	179	0	221	0
19	746	0	0	0	0	4,615	0	0.35	165	0	208	0
20	1,077	0	0	0	0	4,615	0	0.35	150	0	192	0
21	1,165	222	0	0	44	4,615	0	0.35	164	0	146	0
22	1,165	324	0	0	97	8,571	0	0.65	379	0	139	0
23	1,165	458	0	0	157	8,571	0	0.65	455	0	271	0
24	1,165	649	0	0	223	8,085	0	0.65	455	0	445	0
25	1,165	785	283	0	299	5,085	0	0.65	455	0	445	0
26	1,165	785	722	0	382	5,030	0	0.65	455	0	445	0
27	1,165	785	1,100	0	475	6,809	0	0.65	455	111	445	0

Month	RUN V											
	S1_2 TAF	S1_3 TAF	S1_4 TAF	S1_5 TAF	S2_2 TAF	C33 CFS	C34B CFS	EI	S3_3 TAF	S3_4 TAF	S4_3 TAF	S4_4 TAF
13	0	0	0	0	0	5,960	0	0.65	68	0	14	0
14	0	0	0	0	0	6,484	0	0.65	127	0	42	0
15	1	0	0	0	0	6,967	0	0.65	205	0	85	0
16	15	0	0	0	0	7,386	0	0.65	291	0	146	0
17	180	0	0	0	14	5,013	258	0.35	280	0	144	0
18	382	0	0	0	37	5,169	360	0.35	267	0	143	0
19	597	0	0	0	67	5,325	461	0.35	252	0	145	0
20	837	0	0	0	105	5,481	563	0.35	236	0	148	0
21	1,087	0	0	0	148	5,637	664	0.35	220	0	154	0
22	1,165	26	0	0	201	8,571	0	0.65	323	0	258	0
23	1,165	162	0	0	260	8,571	0	0.65	416	0	372	0
24	1,165	369	0	0	326	7,836	0	0.65	455	0	445	47
25	1,165	651	0	0	402	7,304	0	0.65	455	0	445	183
26	1,165	785	163	0	484	7,433	0	0.65	455	0	445	326
27	1,165	785	494	0	578	7,575	0	0.65	455	0	445	485

CHAPTER 8: SUMMARY AND CONCLUSIONS

Throughout the United States, water resources projects are experiencing reduced ability to fulfill demands. Increases in water demands have intensified competition over water allocation and operations. Water resources system models are often used to analyze trade-offs, facilitate better decision-making, and resolve conflict. Most newer water supply simulations models employ optimization methods to allocate water and operations according to fixed operational priorities for each time-step, simulating the efforts of capable system operators attempting to achieve a given set of operational priorities. For extensive complex networks with return flows, loops arising from pumping, and proportional delivery reductions for equal-priority deliveries, the assignment of unit weights can be a matter of some art and controversy.

This dissertation presents a generalized method to automate the computation of unit weights to guarantee priority-preserving behavior for network flow- and linear programming-based simulation models. The method presented in this dissertation both simplifies and an extends of the work presented by Israel and Lund (1999). The simplification lies in the reduction of the eight original constraints to three more general constraints. The extension to Israel and Lund is threefold: procedures to compute (i) weights that result in proportional delivery reductions for equal priorities on arc-flow variables, (ii) negative weights to minimize water allocation where it is not desired, and (iii) objective function coefficients for non-NFP decision variables that are introduced to the LP driven model to better represent the physical system it simulates.

Many test case examples are presented in this dissertation. The examples illustrate various network configurations and how priority preserving weights are computed and used to allocate water by priority. The examples also illustrate some paradoxes of water management under prioritized deliveries and return flows, such as occasions when priority optimization implies that some lower priorities receive water when intermediate priorities do not (to better supply more senior demands with return flows). Chapter 4 presents the application of the method to a general LP driven simulation model. The automated method can compute most, but not all, objective function coefficients for an LP driven model. Depending on the types of non-NFP programming constraints, manual calibration might be required for some coefficients.

A step-by-step procedure to generate priority preserving weights for LP driven simulations is described in Chapter 7. For LP driven simulations, in addition to defining the network connectivity and return flow factors, weighted decision variables must be sorted by type. The arc-flow decision variables must be prioritized separately for variables where allocation of water is desired and for those where delivery of water is to be minimized (e.g., floods). Positive weights on arc-flow variables are computed with the automated procedure while the negative weights on arc-flow variables are temporarily set to allow the calibration of the objective function coefficients of non-NFP decision variables to take place. The final set of negative weights on arc-flow variables can then be calculated using the automated procedure. The resulting weight set should then be tested to ensure both priority preservation and accurate simulation of the system being modeled.

In Chapter 7 I explain how the weight set generated by the method described in Chapter 2 is priority preserving for NFP driven simulations. I further extend this to discuss how different types of non-NFP linear constraints can affect the allocation of water in a LP driven simulation and how the allocation remains priority preserving. Non-NFP decision variables and constraints are classified and the effect of their addition to NFP driven simulation models is discussed. The addition of any linear constraint simply reduces the solution space of the original NFP driven model. Furthermore, a weight set that is preserves priorities for a particular solution space is also preserves priorities for a subset of this solution space created by adding more linear constraints to the problem. Linear objective functions in both NFPs and LPs guarantee that the priority preserving quality of the weights computed is maintained.

The method proposed in this dissertation fills a gap in planning and management simulation modeling. It provides practitioners with a tractable and defendable procedure to generate objective function coefficients for LP driven simulations that is easily implemented with a simple description of network connectivity and water demand priorities. The method should facilitate studies in which major changes to the system are made, removing some "art" and adding more "science".

An additional contribution of this research lies in the analysis of the LP driven simulation problem itself. The classification of variables and constraint helps tease out the different aspects of the LP, what each decision variable is (arc-flow vs. non arc-flow) and how its objective function coefficient should be considered, computed, and its role in the simulation. While objective function coefficients of arc-flow decision variables are part of the prioritized weight set, objective function coefficients of non arc-flow priorities are parameters to be calibrated within a given priority preserving weight set to accurately reflect the regulatory and institutional constraints of the system being simulated. Furthermore, the understanding of the role additional non-NFP constraints play as higher level priorities than those associated with delivery allocation is an important consideration when evaluating modeling results.

Another contribution of this dissertation is the method used in Chapter 7 to verify that the weight set computed is priority preserving. By striping a LP driven model of all non-NFP constraints and progressively adding these constraints in successive simulations where water supply is gradually increased while keeping all demands constant, it is possible to confirm that the weight set computed for a particular simulation model is priority preserving.

Further research in the procedure to generate priority preserving weights for LP driven simulations include a method to incorporate more non-NFP variables in the automated algorithm. This may or may not be feasible, and might depend how similar a non-NFP variable is to arc-flow variables and the extent to which it is a true calibration parameter (slack and surplus variables). Methods to reduce the numerical spread in the computed weights for equal priority demands would also be an improvement to the method presented in this dissertation. Another potential improvement to the automated method would be to compute positive and negative weights simultaneously, at the very least for NFP variables.

REFERENCES

Andreu, J., J. Capilla, and E. Sanchı´s (1996). “AQUATOOL, a generalized decision-support system for water-resources planning and operational management.” *J. Hydro*., Amsterdam, 177, 269–291.

Andrews, E., F.I. Chung, and J.R. Lund, (1992). “Multilayered, priority-based simulation of conjunctive facilities.” *Journal of Water Resources Planning and Management*, ASCE, 118(1), 32–53.

Brendecke, C. M, W.B. DeOreo, Payton, E.A. Payton, and L.T. Rozaklis, (1989). “Network models of water rights and system operation.” *Journal of Water Resources Planning and Management*, ASCE, 115(5), 684–698.

Brown, M. J. (2005). “Priority Based Reservoir Optimization using Linear Programming: Application to Flood Operation of the Iowa/Des Moines River System.” Masters Thesis, University of California, Davis, California.

Chung, F. I., M.C. Archer, J.J. DeVries (1989). “Network flow algorithm applied to California aqueduct simulation.” *Journal of Water Resources Planning and Management*, ASCE, 115(2), 131–147.

Draper, A. J., A. Munevar, S.K. Arora, E. Reyes, N.L. Parker, F.I. Chung, L.E. Peterson (2004). “CalSim: Generalized Model for Reservoir System Analysis.” *Journal of Water Resources Planning and Management*, ASCE, 130(6), 480-489.

Eschenbach, E.,T. Magee, E. Zagona, M. Goronflo, and R. Shane (2001). “Goal programming decision support system for multiobjective operation of reservoir systems.” *Journal of Water Resources Planning and Management,* 127(2), 108–120.

Ferreira, I.C., S.K Tanaka, S.P. Hollinshead, and J.R. Lund (2005). “Musings on the CalSim II Model: A Survey of California’s Water Community.” *San Francisco Estuary and Watershed Science*, 3(1), Article 1.

Fredericks, J. W., J.W. Labadie and Altenhofen, J. M. (1998). “Decision support system for conjunctive stream-aquifer management.” *Journal of Water Resources Planning and Management*, ASCE, 124(2), 69–78.

Frevert, D. K., J.W. Labadie, R.K. Larson, and N.L. Parker (1994). “Integration of Water Rights and Network Flow Modeling in the Upper Snake River Basin.” *Water Policy and Management: Solving the Problems. Proc. of the 21st Annual ASCE Water Resources Planning and Management Conf.*, ASCE, Denver, Colorado.

Graham, L. P., J.W. Labadie, I.P. Hutchison, and K.A. Ferguson, (1986). “Allocation of augmented water supply under a priority water rights system.” *Water Resources Research*, 22(7), 1083–1094.

Israel, M S., and J.R. Lund, (1999). “Priority preserving unit penalties in network flow modeling.” *Journal of Water Resources Planning and Management,* 125(4), 205–214.

Karamouz, M.,M. Houck and J. Delleur, (1992). "Optimization and simulation of multiple reservoir systems." *Journal of Water Resources Planning and Management,* 118(1), 71–81.

Kuczera, G., and G. Diment (1988). "General water supply system simulation model: WASP." *Journal of Water Resources Planning and Management,* ASCE, 114(4), 365–383.

Labadie, J. W., A.M. Pineda, and D.A. Bode (1995). "River basin model for water rights planning, MODSIM." *Tech. Manual,* Department of Civil Engineering, Colorado State University, Fort Collins, Colorado.

Labadie, J. W. (1994) "Integration of Water Quantity and Quality in River Basin Network Flow Modeling." *Water Policy and Management: Solving the Problems. Proc. of the 21st Annual ASCE Water Resources Planning and Management Conf.*, ASCE, Denver, Colorado.

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

Palmer, R. (2000). "Task committee report on shared vision modeling in water resources planning." *Proc., 26th Annual ASCE Water Resources Planning and Management Conf.*, Minneapolis.

SEI (Stockholm Environment Institute) 2001. *WEAP: Water evaluation and planning system – user guide.* Boston, USA.

user guide. Boston, USA. Sigvaldason, O. T. (1976). "A simulation model for operating a multipurpose multireservoir system." *Water Resources Research*, 12(2), 263–278.

Wurbs, R. (1996). *Modeling and analysis of reservoir system operations*, Prentice-Hall, Upper Saddle River, NJ, 356 pp.

ReVelle, C. S., E.E. Whitlatch, Jr., J.R. Wright,. (2004). *Civil and environmental systems engineering,* Pearson Prentice Hall, Upper Saddle River, NJ, 552 pp.

XA Software: Sunset Software Technology, 1613 Chelsea Road, Suite 153, San Marino, CA 91105, 626-441-1565 www.sunsetsoft.com

Hydrologics, Inc. 10440 Shaker Drive, Suite 104, Columbia, MD 2104 *What is OASIS with OCL?* Online document. www.hydrologics.net/pdf/oasis.pdf

APPENDIX A

Appendix A-1: XA Output for Examples 1

```
Rows 63 Columns 7 NonZeros 83 A's 1stDimSize 100
Minimize Solve Number 1
OBJ: X1 - X7

Constraints
C1: X6 - X7 >= 1
C2: X5 - X6 >= 1
C3: X1 - X2 >= 1
C4: X7 >= 1
C5: X4 - X5 >= 1
C6: X2 - X3 >= 1
C7: X3 - X4 >= 1
C8: X6 - X7 >= 1
C9: X5 - 0.5 X7 >= 1
C10: X1 - 0.5 X2 - 0.5 X3 - X4 - X7 >= 1
C11: X7 >= 1
C12: X4 >= 1
C13: X2 - X3 >= 1
C14: X3 >= 1
C15: X6 >= 1
C16: X6 >= 1
C17: X6 >= 1
C18: X6 >= 1
C19: X6 >= 1
C20: X6 >= 1
C21: X6 >= 1
C22: X5 >= 1
C23: X5 >= 1
C24: X5 >= 1
C25: X5 >= 1
C26: X5 >= 1
C27: X5 >= 1
C28: X5 >= 1
C29: X1 >= 1
C30: X1 - 2 X5 >= 1
C31: X1 >= 1
C32: X1 >= 1
C33: X1 >= 1
C34: X1 >= 1
C35: X1 >= 1
C36: X7 >= 1
C37: X7 >= 1
C38: X7 >= 1
C39: X7 >= 1
C40: X7 >= 1
C41: X7 >= 1
C42: X7 >= 1
C43: X4 >= 1
C44: X4 - 2 X5 >= 1
C45: X4 >= 1
C46: X4 - 2 X7 >= 1
C47: X4 >= 1
C48: X4 >= 1
C49: X4 >= 1
C50: X2 >= 1
C51: X2 - 2 X5 >= 1
C52: X2 >= 1
C53: X2 - 2 X7 >= 1
```

```
C54: X2 >= 1
C55: X2 >= 1
C56: X2 >= 1
C57: X3 >= 1
C58: X3 - 2 X5 >= 1
C59: X3 >= 1
C60: X3 - 2 X7 >= 1
C61: X3 >= 1
C62: X3 >= 1
C63: X3 >= 1


STATISTICS - RUNTIME Wed Nov 30 16:06:14 2005
  xa VERSION 13.66 NT DLL    USABLE MEMORY 635.5 MBYTE
  ENV ID 1 SOLVE NUMBER 1
  VARIABLES 7
     0 LOWER, 0 FIXED, 0 UPPER, 0 FREE
  CONSTRAINTS 64
     63 GE, 0 EQ, 0 LE, 1 NULL/FREE, 0 RANGED.
     85 NON-ZEROS WORK 55,528,504
  MINIMIZATION.
  University of California, Davis - 1206701
  Civil & Environmental Engineering/Ines Ferreira 32420-21000


L P   O P T I M A L   S O L U T I O N ---> OBJECTIVE 16.50000
SOLVE 1 TIME 00:00:00  ITER 5  MEMORY USED   0.0%


File: RUNTIME                                           Wed Nov 30 16:06:14 2005
Page 1
SOLUTION REPORT - COLUMN ACTIVITY SOLVE NUMBER 1
NUMBER.COLUMNS  AT  .ACTIVITY..INPUT COST..LOWER LIMIT.UPPER LIMIT..REDUCED COST.
     0  X1      BS    17.50000    1.00000          .            NONE         .
     1  X2      BS     9.00000     .               .            NONE         .
     2  X3      BS     8.00000     .               .            NONE         .
     3  X4      BS     7.00000     .               .            NONE         .
     4  X5      BS     3.00000     .               .            NONE         .
     5  X6      BS     2.00000     .               .            NONE         .
     6  X7      BS     1.00000   -1.00000          .            NONE         .

File: RUNTIME                                           Wed Nov 30 16:06:14 2005
Page 2
CONSTRAINT REPORT - ROW ACTIVITY SOLVE NUMBER 1
NUMBER..ROW.. AT .ACTIVITY.SLACK ACTIVITY..LOWER LIMIT...UPPER LIMIT..DUAL ACTIVITY
     0  OBJ   BS   16.50000   -16.50000        NONE         NONE         -1.00000
     1  C1    LL    1.00000       .          1.00000        NONE          4.00000
     2  C2    LL    1.00000       .          1.00000        NONE          4.00000
     3  C3    BS    8.50000    -7.50000      1.00000        NONE           .
     4  C4    LL    1.00000       .          1.00000        NONE          4.00000
     5  C5    BS    4.00000    -3.00000      1.00000        NONE           .
     6  C6    LL    1.00000       .          1.00000        NONE          0.50000
     7  C7    LL    1.00000       .          1.00000        NONE          1.00000
     8  C8    BS    1.00000       .          1.00000        NONE           .
     9  C9    BS    2.50000    -1.50000      1.00000        NONE           .
    10  C10   LL    1.00000       .          1.00000        NONE          1.00000
    11  C11   BS    1.00000       .          1.00000        NONE           .
    12  C12   BS    7.00000    -6.00000      1.00000        NONE           .
    13  C13   BS    1.00000       .          1.00000        NONE           .
    14  C14   BS    8.00000    -7.00000      1.00000        NONE           .
    15  C15   BS    2.00000    -1.00000      1.00000        NONE           .
    16  C16   BS    2.00000    -1.00000      1.00000        NONE           .
    17  C17   BS    2.00000    -1.00000      1.00000        NONE           .
```

18	C18	BS	2.00000	-1.00000	1.00000	NONE	.
19	C19	BS	2.00000	-1.00000	1.00000	NONE	.
20	C20	BS	2.00000	-1.00000	1.00000	NONE	.
21	C21	BS	2.00000	-1.00000	1.00000	NONE	.
22	C22	BS	3.00000	-2.00000	1.00000	NONE	.
23	C23	BS	3.00000	-2.00000	1.00000	NONE	.
24	C24	BS	3.00000	-2.00000	1.00000	NONE	.
25	C25	BS	3.00000	-2.00000	1.00000	NONE	.
26	C26	BS	3.00000	-2.00000	1.00000	NONE	.
27	C27	BS	3.00000	-2.00000	1.00000	NONE	.
28	C28	BS	3.00000	-2.00000	1.00000	NONE	.
29	C29	BS	17.50000	-16.50000	1.00000	NONE	.
30	C30	BS	11.50000	-10.50000	1.00000	NONE	.
31	C31	BS	17.50000	-16.50000	1.00000	NONE	.
32	C32	BS	17.50000	-16.50000	1.00000	NONE	.
33	C33	BS	17.50000	-16.50000	1.00000	NONE	.
34	C34	BS	17.50000	-16.50000	1.00000	NONE	.
35	C35	BS	17.50000	-16.50000	1.00000	NONE	.
36	C36	BS	1.00000	.	1.00000	NONE	.
37	C37	BS	1.00000	.	1.00000	NONE	.
38	C38	BS	1.00000	.	1.00000	NONE	.
39	C39	BS	1.00000	.	1.00000	NONE	.
40	C40	BS	1.00000	.	1.00000	NONE	.
41	C41	BS	1.00000	.	1.00000	NONE	.
42	C42	BS	1.00000	.	1.00000	NONE	.
43	C43	BS	7.00000	-6.00000	1.00000	NONE	.
44	C44	LL	1.00000	.	1.00000	NONE	2.00000
45	C45	BS	7.00000	-6.00000	1.00000	NONE	.
46	C46	BS	5.00000	-4.00000	1.00000	NONE	.
47	C47	BS	7.00000	-6.00000	1.00000	NONE	.
48	C48	BS	7.00000	-6.00000	1.00000	NONE	.
49	C49	BS	7.00000	-6.00000	1.00000	NONE	.
50	C50	BS	9.00000	-8.00000	1.00000	NONE	.
51	C51	BS	3.00000	-2.00000	1.00000	NONE	.
52	C52	BS	9.00000	-8.00000	1.00000	NONE	.
53	C53	BS	7.00000	-6.00000	1.00000	NONE	.
54	C54	BS	9.00000	-8.00000	1.00000	NONE	.
55	C55	BS	9.00000	-8.00000	1.00000	NONE	.
56	C56	BS	9.00000	-8.00000	1.00000	NONE	.
57	C57	BS	8.00000	-7.00000	1.00000	NONE	.
58	C58	BS	2.00000	-1.00000	1.00000	NONE	.
59	C59	BS	8.00000	-7.00000	1.00000	NONE	.
60	C60	BS	6.00000	-5.00000	1.00000	NONE	.
61	C61	BS	8.00000	-7.00000	1.00000	NONE	.
62	C62	BS	8.00000	-7.00000	1.00000	NONE	.
63	C63	BS	8.00000	-7.00000	1.00000	NONE	.

Appendix A-2: CalSim Input Files for Example 1

Channel-table.wresl

```
define C1 {lower 0 upper 3000000 kind 'FLOW-CHANNEL' units 'TAF'} !Release from
Reservoir 1
define C2 {lower 0 upper 3000000 kind 'FLOW-CHANNEL' units 'TAF'}
define C3 {lower 0 upper 3000000 kind 'FLOW-CHANNEL' units 'TAF'}
define C4 {lower 0 upper 3000000 kind 'FLOW-CHANNEL' units 'TAF'}
define C5 {lower 0 upper 3000000 kind 'FLOW-CHANNEL' units 'TAF'}
define C6 {lower 0 upper 3000000 kind 'FLOW-CHANNEL' units 'TAF'}
define C7 {lower 0 upper 3000000 kind 'FLOW-CHANNEL' units 'TAF'}
```

Connectivity-table.wresl

```
goal continuity1 {I1-C1-D1=S1-S1(-1)}
goal continuity2 {C1-C2-D2=0}
goal continuity3 {C2+R3-C3-D3=0}
goal continuity4 {C3+R4a+R4b-C4=S4-S4(-1)}
goal continuity5 {C4-C5-D5=0}
goal continuity6 {C5-C6-D6=0}
goal continuity7 {C6+R7a+R7b-C7=0}
```

Delivery-table.wresl

```
define D1 {upper 10 kind 'FLOW-DELIVERY' units 'TAF'}
define D2 {upper 10 kind 'FLOW-DELIVERY' units 'TAF'}
define D3 {upper 10 kind 'FLOW-DELIVERY' units 'TAF'}
define D4 {upper 10 kind 'FLOW-DELIVERY' units 'TAF'}
define D5 {upper 10 kind 'FLOW-DELIVERY' units 'TAF'}
define D6 {upper 10 kind 'FLOW-DELIVERY' units 'TAF'}
```

Inflow-table.wresl

```
define I1 {timeseries kind' FLOW-INFLOW' units 'TAF'} !Inflow to reservoir 1
```

Reservoir-table.wresl

```
define S1 {std kind 'STORAGE' units 'TAF'}
define S1Cap {value 80}
goal S1 {S1 < S1Cap }
define S4 {std kind 'STORAGE' units 'TAF'}
define S4Cap {value 80}
goal S4 {S4 < S4Cap }
```

Return-table.wresl

```
define R3 {std kind 'FLOW-RETURN' units 'TAF'}
define rfactor_R3 {value 0.5}
goal returnflowR3 {R3=rfactor_R3*D1}
define R4a {std kind 'FLOW-RETURN' units 'TAF'}
define rfactor_R4a {value 0.5}
goal returnflowR4a {R4a=rfactor_R4a*D2}
define R4b {std kind 'FLOW-RETURN' units 'TAF'}
define rfactor_R4b {value 0.5}
goal returnflowR4b {R4b=rfactor_R4b*D3}
define R7a {std kind 'FLOW-RETURN' units 'TAF'}
define rfactor_R7a {value 0.5}
goal returnflowR7a {R7a=rfactor_R7a*D6}
define R7b {std kind 'FLOW-RETURN' units 'TAF'}
define rfactor_R7b {value 0.5}
goal returnflowR7b {R7b=rfactor_R7b*D5}
```

Weight-table.wresl

```
Objective obj = {[S1  ,   2.00],
[D1  ,   3.00],
[D2  ,  17.50],
[D3  ,   1.00],
[S4  ,   7.00],
[D5  ,   9.00],
[D6  ,   8.00]}
```

Appendix A-3: Preprocessor Output for Example 2

```
Rows 80 Columns 8 NonZeros 105 A's 1stDimSize 100
Minimize Solve Number 1
OBJ: X1 - X8

Constraints
```

```
C1: X4 - X5 >= 1
C2: X1 - X2 >= 1
C3: X3 - X4 >= 1
C4: X7 - X8 >= 1
C5: X8 >= 1
C6: X5 - X6 >= 1
C7: X6 - X7 >= 1
C8: X2 - X3 >= 1
C9: X4 - X5 - X6 - X7 - X8 >= 1
C10: X1 - X2 - X3 - 0.5 X5 - X6 - 0.5 X7 - 0.5 X8 >= 1
C11: X3 - X5 - 0.5 X6 - X7 - X8 >= 1
C12: X7 - 0.5 X8 >= 1
C13: X8 >= 1
C14: X5 >= 1
C15: X6 >= 1
C16: X2 >= 1
C17: X4 >= 1
C18: X4 >= 1
C19: X4 >= 1
C20: X4 >= 1
C21: X4 >= 1
C22: X4 >= 1
C23: X4 >= 1
C24: X4 >= 1
C25: X1 >= 1
C26: X1 >= 1
C27: X1 >= 1
C28: X1 >= 1
C29: X1 >= 1
C30: X1 >= 1
C31: X1 >= 1
C32: X1 >= 1
C33: X3 >= 1
C34: X3 >= 1
C35: X3 >= 1
C36: X3 >= 1
C37: X3 >= 1
C38: X3 >= 1
C39: X3 >= 1
C40: X3 >= 1
C41: X7 >= 1
C42: X7 >= 1
C43: X7 >= 1
C44: X7 >= 1
C45: X7 >= 1
C46: X7 >= 1
C47: X7 >= 1
C48: X7 >= 1
C49: X8 >= 1
C50: X8 >= 1
C51: X8 >= 1
C52: X8 >= 1
C53: X8 >= 1
C54: X8 >= 1
C55: X8 >= 1
C56: X8 >= 1
C57: X5 >= 1
C58: X5 >= 1
C59: X5 >= 1
C60: X5 - 2 X7 >= 1
C61: X5 - 2 X8 >= 1
C62: X5 >= 1
C63: X5 >= 1
```

```
C64: X5 >= 1
C65: X6 >= 1
C66: X6 >= 1
C67: X6 >= 1
C68: X6 >= 1
C69: X6 >= 1
C70: X6 >= 1
C71: X6 >= 1
C72: X6 >= 1
C73: X2 >= 1
C74: X2 >= 1
C75: X2 - 2 X3 >= 1
C76: X2 >= 1
C77: X2 >= 1
C78: X2 >= 1
C79: X2 - 2 X6 >= 1
C80: X2 >= 1


STATISTICS - RUNTIME Sun Oct 16 10:22:38 2005
  xa VERSION 13.66 NT DLL    USABLE MEMORY 635.5 MBYTE
  ENV ID 1 SOLVE NUMBER 1
  VARIABLES 8
     0 LOWER, 0 FIXED, 0 UPPER, 0 FREE
  CONSTRAINTS 81
     80 GE, 0 EQ, 0 LE, 1 NULL/FREE, 0 RANGED.
     107 NON-ZEROS WORK 55,528,014
  MINIMIZATION.
  University of California, Davis - 1206701
  Civil & Environmental Engineering/Ines Ferreira 32420-21000


L P   O P T I M A L   S O L U T I O N ---> OBJECTIVE 47.00000
SOLVE 1 TIME 00:00:00  ITER 7  MEMORY USED   0.0%


File: RUNTIME                                   Sun Oct 16 10:22:38 2005 Page 1
SOLUTION REPORT - COLUMN ACTIVITY SOLVE NUMBER 1
NUMBER.COLUMNS AT ..ACTIVITY.INPUT COST.LOWER LIMIT.UPPER LIMIT.REDUCED COST.
  0       X1    BS     48.00000     1.00000      .             NONE         .
  1       X2    BS     27.00000      .           .             NONE         .
  2       X3    BS     13.00000      .           .             NONE         .
  3       X4    BS     12.00000      .           .             NONE         .
  4       X5    BS      5.00000      .           .             NONE         .
  5       X6    BS      3.00000      .           .             NONE         .
  6       X7    BS      2.00000      .           .             NONE         .
  7       X8    BS      1.00000    -1.00000      .             NONE         .

File: RUNTIME                                           Sun Oct 16 10:22:38
2005 Page 2
CONSTRAINT REPORT - ROW ACTIVITY SOLVE NUMBER 1
NUMBER..ROW.. AT.ACTIVITY..SLACK ACTIVITY.LOWER LIMIT.UPPER LIMIT.DUAL ACTIVITY
   0    OBJ   BS   47.00000    -47.00000       NONE            NONE   -1.00000
   1    C1    BS    7.00000     -6.00000     1.00000           NONE       .
   2    C2    BS   21.00000    -20.00000     1.00000           NONE       .
   3    C3    LL    1.00000       .          1.00000           NONE    3.00000
   4    C4    LL    1.00000       .          1.00000           NONE   14.50000
   5    C5    LL    1.00000       .          1.00000           NONE   17.00000
   6    C6    BS    2.00000     -1.00000     1.00000           NONE       .
   7    C7    LL    1.00000       .          1.00000           NONE    4.00000
   8    C8    BS   14.00000    -13.00000     1.00000           NONE       .
   9    C9    LL    1.00000       .          1.00000           NONE    3.00000
  10    C10   LL    1.00000       .          1.00000           NONE    1.00000
```

11	C11	BS	3.50000	-2.50000	1.00000	NONE	.
12	C12	BS	1.50000	-0.50000	1.00000	NONE	.
13	C13	BS	1.00000	.	1.00000	NONE	.
14	C14	BS	5.00000	-4.00000	1.00000	NONE	.
15	C15	BS	3.00000	-2.00000	1.00000	NONE	.
16	C16	BS	27.00000	-26.00000	1.00000	NONE	.
17	C17	BS	12.00000	-11.00000	1.00000	NONE	.
18	C18	BS	12.00000	-11.00000	1.00000	NONE	.
19	C19	BS	12.00000	-11.00000	1.00000	NONE	.
20	C20	BS	12.00000	-11.00000	1.00000	NONE	.
21	C21	BS	12.00000	-11.00000	1.00000	NONE	.
22	C22	BS	12.00000	-11.00000	1.00000	NONE	.
23	C23	BS	12.00000	-11.00000	1.00000	NONE	.
24	C24	BS	12.00000	-11.00000	1.00000	NONE	.
25	C25	BS	48.00000	-47.00000	1.00000	NONE	.
26	C26	BS	48.00000	-47.00000	1.00000	NONE	.
27	C27	BS	48.00000	-47.00000	1.00000	NONE	.
28	C28	BS	48.00000	-47.00000	1.00000	NONE	.
29	C29	BS	48.00000	-47.00000	1.00000	NONE	.
30	C30	BS	48.00000	-47.00000	1.00000	NONE	.
31	C31	BS	48.00000	-47.00000	1.00000	NONE	.
32	C32	BS	48.00000	-47.00000	1.00000	NONE	.
33	C33	BS	13.00000	-12.00000	1.00000	NONE	.
34	C34	BS	13.00000	-12.00000	1.00000	NONE	.
35	C35	BS	13.00000	-12.00000	1.00000	NONE	.
36	C36	BS	13.00000	-12.00000	1.00000	NONE	.
37	C37	BS	13.00000	-12.00000	1.00000	NONE	.
38	C38	BS	13.00000	-12.00000	1.00000	NONE	.
39	C39	BS	13.00000	-12.00000	1.00000	NONE	.
40	C40	BS	13.00000	-12.00000	1.00000	NONE	.
41	C41	BS	2.00000	-1.00000	1.00000	NONE	.
42	C42	BS	2.00000	-1.00000	1.00000	NONE	.
43	C43	BS	2.00000	-1.00000	1.00000	NONE	.
44	C44	BS	2.00000	-1.00000	1.00000	NONE	.
45	C45	BS	2.00000	-1.00000	1.00000	NONE	.
46	C46	BS	2.00000	-1.00000	1.00000	NONE	.
47	C47	BS	2.00000	-1.00000	1.00000	NONE	.
48	C48	BS	2.00000	-1.00000	1.00000	NONE	.
49	C49	BS	1.00000	.	1.00000	NONE	.
50	C50	BS	1.00000	.	1.00000	NONE	.
51	C51	BS	1.00000	.	1.00000	NONE	.
52	C52	BS	1.00000	.	1.00000	NONE	.
53	C53	BS	1.00000	.	1.00000	NONE	.
54	C54	BS	1.00000	.	1.00000	NONE	.
55	C55	BS	1.00000	.	1.00000	NONE	.
56	C56	BS	1.00000	.	1.00000	NONE	.
57	C57	BS	5.00000	-4.00000	1.00000	NONE	.
58	C58	BS	5.00000	-4.00000	1.00000	NONE	.
59	C59	BS	5.00000	-4.00000	1.00000	NONE	.
60	C60	LL	1.00000	.	1.00000	NONE	3.50000
61	C61	BS	5.00000	-4.00000	1.00000	NONE	.
62	C62	BS	5.00000	-4.00000	1.00000	NONE	.
63	C63	BS	5.00000	-4.00000	1.00000	NONE	.
64	C64	BS	5.00000	-4.00000	1.00000	NONE	.
65	C65	BS	3.00000	-2.00000	1.00000	NONE	.
66	C66	BS	3.00000	-2.00000	1.00000	NONE	.
67	C67	BS	3.00000	-2.00000	1.00000	NONE	.
68	C68	BS	3.00000	-2.00000	1.00000	NONE	.
69	C69	BS	3.00000	-2.00000	1.00000	NONE	.
70	C70	BS	3.00000	-2.00000	1.00000	NONE	.
71	C71	BS	3.00000	-2.00000	1.00000	NONE	.
72	C72	BS	3.00000	-2.00000	1.00000	NONE	.
73	C73	BS	27.00000	-26.00000	1.00000	NONE	.

74	C74	BS	27.00000	-26.00000	1.00000	NONE	.
75	C75	LL	1.00000	.	1.00000	NONE	1.00000
76	C76	BS	27.00000	-26.00000	1.00000	NONE	.
77	C77	BS	27.00000	-26.00000	1.00000	NONE	.
78	C78	BS	27.00000	-26.00000	1.00000	NONE	.
79	C79	BS	21.00000	-20.00000	1.00000	NONE	.
80	C80	BS	27.00000	-26.00000	1.00000	NONE	.

Appendix A-4: Output for Example 3

```
Rows 168 Columns 13 NonZeros 221 A's 1stDimSize 500
Minimize Solve Number 1
OBJ: X1 - X12

Constraints
C1: X9 - X10 >= 1
C2: X7 - X8 >= 1
C3: X6 - X7 >= 1
C4: X1 - X2 >= 1
C5: X5 - X6 >= 1
C6: X8 - X9 >= 1
C7: X4 - X5 >= 1
C8: X2 - X3 >= 1
C9: X10 - X11 >= 1
C10: X3 - X4 >= 1
C11: X11 - X12 >= 1
C12: X12 >= 1
C13: X9 - X10 - X11 - X12 >= 1
C14: X7 - 0.5 X10 - 0.5 X11 - 0.5 X12 >= 1
C15: X6 - 0.5 X10 - 0.5 X11 - 0.5 X12 >= 1
C16: X1 - X2 - X3 - X4 - X5 - X8 - X10 - X11 - X12 >= 1
C17: X5 - X8 - 0.5 X10 - 0.5 X11 - 0.5 X12 >= 1
C18: X8 - 0.5 X10 - 0.5 X11 - 0.5 X12 >= 1
C19: X4 - X10 - X11 - X12 >= 1
C20: X2 - 0.5 X10 >= 1
C21: X10 >= 1
C22: X3 - 0.5 X11 - 0.5 X12 >= 1
C23: X11 - X12 >= 1
C24: X12 >= 1
C25: X9 >= 1
C26: X9 >= 1
C27: X9 >= 1
C28: X9 >= 1
C29: X9 >= 1
C30: X9 >= 1
C31: X9 >= 1
C32: X9 >= 1
C33: X9 >= 1
C34: X9 >= 1
C35: X9 >= 1
C36: X9 >= 1
C37: X7 >= 1
C38: X7 >= 1
C39: X7 >= 1
C40: X7 >= 1
C41: X7 >= 1
C42: X7 >= 1
C43: X7 >= 1
C44: X7 >= 1
C45: X7 >= 1
C46: X7 >= 1
```

```
C47: X7 >= 1
C48: X7 >= 1
C49: X6 >= 1
C50: X6 - 2 X7 >= 1
C51: X6 >= 1
C52: X6 >= 1
C53: X6 >= 1
C54: X6 >= 1
C55: X6 >= 1
C56: X6 >= 1
C57: X6 >= 1
C58: X6 >= 1
C59: X6 >= 1
C60: X6 >= 1
C61: X1 >= 1
C62: X1 >= 1
C63: X1 >= 1
C64: X1 >= 1
C65: X1 >= 1
C66: X1 >= 1
C67: X1 >= 1
C68: X1 >= 1
C69: X1 >= 1
C70: X1 >= 1
C71: X1 >= 1
C72: X1 >= 1
C73: X5 >= 1
C74: X5 >= 1
C75: X5 >= 1
C76: X5 >= 1
C77: X5 >= 1
C78: X5 >= 1
C79: X5 >= 1
C80: X5 >= 1
C81: X5 >= 1
C82: X5 >= 1
C83: X5 >= 1
C84: X5 >= 1
C85: X8 >= 1
C86: X8 >= 1
C87: X8 >= 1
C88: X8 >= 1
C89: X8 >= 1
C90: X8 >= 1
C91: X8 >= 1
C92: X8 >= 1
C93: X8 >= 1
C94: X8 >= 1
C95: X8 >= 1
C96: X8 >= 1
C97: X4 >= 1
C98: X4 >= 1
C99: X4 >= 1
C100: X4 >= 1
C101: X4 - 2 X5 >= 1
C102: X4 - 2 X8 >= 1
C103: X4 >= 1
C104: X4 >= 1
C105: X4 >= 1
C106: X4 >= 1
C107: X4 >= 1
C108: X4 >= 1
C109: X2 >= 1
```

```
C110: X2 - 2 X7 >= 1
C111: X2 - 2 X6 >= 1
C112: X2 >= 1
C113: X2 - 2 X5 >= 1
C114: X2 - 2 X8 >= 1
C115: X2 >= 1
C116: X2 >= 1
C117: X2 >= 1
C118: X2 >= 1
C119: X2 >= 1
C120: X2 >= 1
C121: X10 >= 1
C122: X10 >= 1
C123: X10 >= 1
C124: X10 >= 1
C125: X10 >= 1
C126: X10 >= 1
C127: X10 >= 1
C128: X10 >= 1
C129: X10 >= 1
C130: X10 >= 1
C131: X10 >= 1
C132: X10 >= 1
C133: X3 >= 1
C134: X3 - 2 X7 >= 1
C135: X3 - 2 X6 >= 1
C136: X3 >= 1
C137: X3 - 2 X5 >= 1
C138: X3 - 2 X8 >= 1
C139: X3 >= 1
C140: X3 >= 1
C141: X3 >= 1
C142: X3 >= 1
C143: X3 >= 1
C144: X3 >= 1
C145: X11 >= 1
C146: X11 >= 1
C147: X11 >= 1
C148: X11 >= 1
C149: X11 >= 1
C150: X11 >= 1
C151: X11 >= 1
C152: X11 >= 1
C153: X11 >= 1
C154: X11 >= 1
C155: X11 >= 1
C156: X11 >= 1
C157: X12 >= 1
C158: X12 >= 1
C159: X12 >= 1
C160: X12 >= 1
C161: X12 >= 1
C162: X12 >= 1
C163: X12 >= 1
C164: X12 >= 1
C165: X12 >= 1
C166: X12 >= 1
C167: X12 >= 1
C168: X12 >= 1


STATISTICS - RUNTIME Mon Feb 13 11:14:22 2006
  xa VERSION 13.66 NT DLL    USABLE MEMORY 635.5 MBYTE
```

```
ENV ID 1 SOLVE NUMBER 1
VARIABLES 13
   0 LOWER, 0 FIXED, 0 UPPER, 0 FREE
CONSTRAINTS 169
   168 GE, 0 EQ, 0 LE, 1 NULL/FREE, 0 RANGED.
   223 NON-ZEROS WORK 55,525,156
MINIMIZATION.
University of California, Davis - 1206701
Civil & Environmental Engineering/Ines Ferreira 32420-21000
```

```
L P   O P T I M A L   S O L U T I O N ---> OBJECTIVE 160.00000
SOLVE 1 TIME 00:00:00  ITER 10  MEMORY USED   0.0%
```

File: RUNTIME Mon Feb 13 11:14:22 2006 Page 1

SOLUTION REPORT - COLUMN ACTIVITY SOLVE NUMBER 1

NUMBER	COLUMNS	AT	ACTIVITY	INPUT COST	LOWER LIMIT	UPPER LIMIT	REDUCED COST
0	X1	BS	161.00000	1.00000	.	NONE	.
1	X2	BS	43.00000	.	.	NONE	.
2	X3	BS	42.00000	.	.	NONE	.
3	X4	BS	41.00000	.	.	NONE	.
4	X5	BS	20.00000	.	.	NONE	.
5	X6	BS	19.00000	.	.	NONE	.
6	X7	BS	9.00000	.	.	NONE	.
7	X8	BS	8.00000	.	.	NONE	.
8	X9	BS	7.00000	.	.	NONE	.
9	X10	BS	3.00000	.	.	NONE	.
10	X11	BS	2.00000	.	.	NONE	.
11	X12	BS	1.00000	-1.00000	.	NONE	.

File: RUNTIME Mon Feb 13 11:14:22 2006 Page 2

CONSTRAINT REPORT - ROW ACTIVITY SOLVE NUMBER 1

NUMBER	ROW	AT	ACTIVITY	SLACK ACTIVITY	LOWER LIMIT	UPPER LIMIT	DUAL ACTIVITY
0	OBJ	BS	160.00000	-160.00000	NONE	NONE	-1.00000
1	C1	BS	4.00000	-3.00000	1.00000	NONE	.
2	C2	LL	1.00000	.	1.00000	NONE	14.00000
3	C3	BS	10.00000	-9.00000	1.00000	NONE	.
4	C4	BS	118.00000	-117.00000	1.00000	NONE	.
5	C5	LL	1.00000	.	1.00000	NONE	7.00000
6	C6	LL	1.00000	.	1.00000	NONE	15.00000
7	C7	BS	21.00000	-20.00000	1.00000	NONE	.
8	C8	LL	1.00000	.	1.00000	NONE	1.00000
9	C9	LL	1.00000	.	1.00000	NONE	16.00000
10	C10	LL	1.00000	.	1.00000	NONE	2.00000
11	C11	LL	1.00000	.	1.00000	NONE	32.00000
12	C12	LL	1.00000	.	1.00000	NONE	47.00000
13	C13	LL	1.00000	.	1.00000	NONE	15.00000
14	C14	BS	6.00000	-5.00000	1.00000	NONE	.
15	C15	BS	16.00000	-15.00000	1.00000	NONE	.
16	C16	LL	1.00000	.	1.00000	NONE	1.00000
17	C17	BS	9.00000	-8.00000	1.00000	NONE	.
18	C18	BS	5.00000	-4.00000	1.00000	NONE	.
19	C19	BS	35.00000	-34.00000	1.00000	NONE	.
20	C20	BS	41.50000	-40.50000	1.00000	NONE	.
21	C21	BS	3.00000	-2.00000	1.00000	NONE	.
22	C22	BS	40.50000	-39.50000	1.00000	NONE	.
23	C23	BS	1.00000	.	1.00000	NONE	.
24	C24	BS	1.00000	.	1.00000	NONE	.

25	C25	BS	7.00000	-6.00000	1.00000	NONE	.
26	C26	BS	7.00000	-6.00000	1.00000	NONE	.
27	C27	BS	7.00000	-6.00000	1.00000	NONE	.
28	C28	BS	7.00000	-6.00000	1.00000	NONE	.
29	C29	BS	7.00000	-6.00000	1.00000	NONE	.
30	C30	BS	7.00000	-6.00000	1.00000	NONE	.
31	C31	BS	7.00000	-6.00000	1.00000	NONE	.
32	C32	BS	7.00000	-6.00000	1.00000	NONE	.
33	C33	BS	7.00000	-6.00000	1.00000	NONE	.
34	C34	BS	7.00000	-6.00000	1.00000	NONE	.
35	C35	BS	7.00000	-6.00000	1.00000	NONE	.
36	C36	BS	7.00000	-6.00000	1.00000	NONE	.
37	C37	BS	9.00000	-8.00000	1.00000	NONE	.
38	C38	BS	9.00000	-8.00000	1.00000	NONE	.
39	C39	BS	9.00000	-8.00000	1.00000	NONE	.
40	C40	BS	9.00000	-8.00000	1.00000	NONE	.
41	C41	BS	9.00000	-8.00000	1.00000	NONE	.
42	C42	BS	9.00000	-8.00000	1.00000	NONE	.
43	C43	BS	9.00000	-8.00000	1.00000	NONE	.
44	C44	BS	9.00000	-8.00000	1.00000	NONE	.
45	C45	BS	9.00000	-8.00000	1.00000	NONE	.
46	C46	BS	9.00000	-8.00000	1.00000	NONE	.
47	C47	BS	9.00000	-8.00000	1.00000	NONE	.
48	C48	BS	9.00000	-8.00000	1.00000	NONE	.
49	C49	BS	19.00000	-18.00000	1.00000	NONE	.
50	C50	LL	1.00000	.	1.00000	NONE	7.00000
51	C51	BS	19.00000	-18.00000	1.00000	NONE	.
52	C52	BS	19.00000	-18.00000	1.00000	NONE	.
53	C53	BS	19.00000	-18.00000	1.00000	NONE	.
54	C54	BS	19.00000	-18.00000	1.00000	NONE	.
55	C55	BS	19.00000	-18.00000	1.00000	NONE	.
56	C56	BS	19.00000	-18.00000	1.00000	NONE	.
57	C57	BS	19.00000	-18.00000	1.00000	NONE	.
58	C58	BS	19.00000	-18.00000	1.00000	NONE	.
59	C59	BS	19.00000	-18.00000	1.00000	NONE	.
60	C60	BS	19.00000	-18.00000	1.00000	NONE	.
61	C61	BS	161.00000	-160.00000	1.00000	NONE	.
62	C62	BS	161.00000	-160.00000	1.00000	NONE	.
63	C63	BS	161.00000	-160.00000	1.00000	NONE	.
64	C64	BS	161.00000	-160.00000	1.00000	NONE	.
65	C65	BS	161.00000	-160.00000	1.00000	NONE	.
66	C66	BS	161.00000	-160.00000	1.00000	NONE	.
67	C67	BS	161.00000	-160.00000	1.00000	NONE	.
68	C68	BS	161.00000	-160.00000	1.00000	NONE	.
69	C69	BS	161.00000	-160.00000	1.00000	NONE	.
70	C70	BS	161.00000	-160.00000	1.00000	NONE	.
71	C71	BS	161.00000	-160.00000	1.00000	NONE	.
72	C72	BS	161.00000	-160.00000	1.00000	NONE	.
73	C73	BS	20.00000	-19.00000	1.00000	NONE	.
74	C74	BS	20.00000	-19.00000	1.00000	NONE	.
75	C75	BS	20.00000	-19.00000	1.00000	NONE	.
76	C76	BS	20.00000	-19.00000	1.00000	NONE	.
77	C77	BS	20.00000	-19.00000	1.00000	NONE	.
78	C78	BS	20.00000	-19.00000	1.00000	NONE	.
79	C79	BS	20.00000	-19.00000	1.00000	NONE	.
80	C80	BS	20.00000	-19.00000	1.00000	NONE	.
81	C81	BS	20.00000	-19.00000	1.00000	NONE	.
82	C82	BS	20.00000	-19.00000	1.00000	NONE	.
83	C83	BS	20.00000	-19.00000	1.00000	NONE	.
84	C84	BS	20.00000	-19.00000	1.00000	NONE	.
85	C85	BS	8.00000	-7.00000	1.00000	NONE	.
86	C86	BS	8.00000	-7.00000	1.00000	NONE	.
87	C87	BS	8.00000	-7.00000	1.00000	NONE	.

88	C88	BS	8.00000	-7.00000	1.00000	NONE	.
89	C89	BS	8.00000	-7.00000	1.00000	NONE	.
90	C90	BS	8.00000	-7.00000	1.00000	NONE	.
91	C91	BS	8.00000	-7.00000	1.00000	NONE	.
92	C92	BS	8.00000	-7.00000	1.00000	NONE	.
93	C93	BS	8.00000	-7.00000	1.00000	NONE	.
94	C94	BS	8.00000	-7.00000	1.00000	NONE	.
95	C95	BS	8.00000	-7.00000	1.00000	NONE	.
96	C96	BS	8.00000	-7.00000	1.00000	NONE	.
97	C97	BS	41.00000	-40.00000	1.00000	NONE	.
98	C98	BS	41.00000	-40.00000	1.00000	NONE	.
99	C99	BS	41.00000	-40.00000	1.00000	NONE	.
100	C100	BS	41.00000	-40.00000	1.00000	NONE	.
101	C101	LL	1.00000	.	1.00000	NONE	3.00000
102	C102	BS	25.00000	-24.00000	1.00000	NONE	.
103	C103	BS	41.00000	-40.00000	1.00000	NONE	.
104	C104	BS	41.00000	-40.00000	1.00000	NONE	.
105	C105	BS	41.00000	-40.00000	1.00000	NONE	.
106	C106	BS	41.00000	-40.00000	1.00000	NONE	.
107	C107	BS	41.00000	-40.00000	1.00000	NONE	.
108	C108	BS	41.00000	-40.00000	1.00000	NONE	.
109	C109	BS	43.00000	-42.00000	1.00000	NONE	.
110	C110	BS	25.00000	-24.00000	1.00000	NONE	.
111	C111	BS	5.00000	-4.00000	1.00000	NONE	.
112	C112	BS	43.00000	-42.00000	1.00000	NONE	.
113	C113	BS	3.00000	-2.00000	1.00000	NONE	.
114	C114	BS	27.00000	-26.00000	1.00000	NONE	.
115	C115	BS	43.00000	-42.00000	1.00000	NONE	.
116	C116	BS	43.00000	-42.00000	1.00000	NONE	.
117	C117	BS	43.00000	-42.00000	1.00000	NONE	.
118	C118	BS	43.00000	-42.00000	1.00000	NONE	.
119	C119	BS	43.00000	-42.00000	1.00000	NONE	.
120	C120	BS	43.00000	-42.00000	1.00000	NONE	.
121	C121	BS	3.00000	-2.00000	1.00000	NONE	.
122	C122	BS	3.00000	-2.00000	1.00000	NONE	.
123	C123	BS	3.00000	-2.00000	1.00000	NONE	.
124	C124	BS	3.00000	-2.00000	1.00000	NONE	.
125	C125	BS	3.00000	-2.00000	1.00000	NONE	.
126	C126	BS	3.00000	-2.00000	1.00000	NONE	.
127	C127	BS	3.00000	-2.00000	1.00000	NONE	.
128	C128	BS	3.00000	-2.00000	1.00000	NONE	.
129	C129	BS	3.00000	-2.00000	1.00000	NONE	.
130	C130	BS	3.00000	-2.00000	1.00000	NONE	.
131	C131	BS	3.00000	-2.00000	1.00000	NONE	.
132	C132	BS	3.00000	-2.00000	1.00000	NONE	.
133	C133	BS	42.00000	-41.00000	1.00000	NONE	.
134	C134	BS	24.00000	-23.00000	1.00000	NONE	.
135	C135	BS	4.00000	-3.00000	1.00000	NONE	.
136	C136	BS	42.00000	-41.00000	1.00000	NONE	.
137	C137	BS	2.00000	-1.00000	1.00000	NONE	.
138	C138	BS	26.00000	-25.00000	1.00000	NONE	.
139	C139	BS	42.00000	-41.00000	1.00000	NONE	.
140	C140	BS	42.00000	-41.00000	1.00000	NONE	.
141	C141	BS	42.00000	-41.00000	1.00000	NONE	.
142	C142	BS	42.00000	-41.00000	1.00000	NONE	.
143	C143	BS	42.00000	-41.00000	1.00000	NONE	.
144	C144	BS	42.00000	-41.00000	1.00000	NONE	.
145	C145	BS	2.00000	-1.00000	1.00000	NONE	.
146	C146	BS	2.00000	-1.00000	1.00000	NONE	.
147	C147	BS	2.00000	-1.00000	1.00000	NONE	.
148	C148	BS	2.00000	-1.00000	1.00000	NONE	.
149	C149	BS	2.00000	-1.00000	1.00000	NONE	.
150	C150	BS	2.00000	-1.00000	1.00000	NONE	.

151	C151	BS	2.00000	-1.00000	1.00000	NONE	.
152	C152	BS	2.00000	-1.00000	1.00000	NONE	.
153	C153	BS	2.00000	-1.00000	1.00000	NONE	.
154	C154	BS	2.00000	-1.00000	1.00000	NONE	.
155	C155	BS	2.00000	-1.00000	1.00000	NONE	.
156	C156	BS	2.00000	-1.00000	1.00000	NONE	.
157	C157	BS	1.00000	.	1.00000	NONE	.
158	C158	BS	1.00000	.	1.00000	NONE	.
159	C159	BS	1.00000	.	1.00000	NONE	.
160	C160	BS	1.00000	.	1.00000	NONE	.
161	C161	BS	1.00000	.	1.00000	NONE	.
162	C162	BS	1.00000	.	1.00000	NONE	.
163	C163	BS	1.00000	.	1.00000	NONE	.
164	C164	BS	1.00000	.	1.00000	NONE	.
165	C165	BS	1.00000	.	1.00000	NONE	.
166	C166	BS	1.00000	.	1.00000	NONE	.
167	C167	BS	1.00000	.	1.00000	NONE	.
168	C168	BS	1.00000	.	1.00000	NONE	.

Appendix A-5: Preprocessor Output for Example 4

```
Rows 35 Columns 5 NonZeros 54 A's 1stDimSize 500
Minimize Solve Number 1
OBJ: X1 - X5

Constraints
C1: X2 - X3 >= 1
C2: X3 - X4 >= 1
C3: X1 - X2 >= 1
C4: X5 >= 1
C5: X4 - X5 >= 1
C6: X2 - X3 - X4 - X5 >= 1
C7: X3 - 0.5 X4 - 0.5 X5 >= 1
C8: X1 - X2 - 0.5 X3 - 0.5 X4 - 0.5 X5 >= 1
C9: X5 >= 1
C10: X4 >= 1
C11: X2 >= 1
C12: X2 - 2 X3 >= 1
C13: X2 >= 1
C14: X2 - 2 X5 >= 1
C15: X2 >= 1
C16: X3 >= 1
C17: X3 >= 1
C18: X3 >= 1
C19: X3 - 2 X5 >= 1
C20: X3 >= 1
C21: X1 >= 1
C22: X1 - 2 X3 >= 1
C23: X1 >= 1
C24: X1 - 2 X5 >= 1
C25: X1 >= 1
C26: X5 >= 1
C27: X5 >= 1
C28: X5 >= 1
C29: X5 >= 1
C30: X5 >= 1
C31: X4 >= 1
C32: X4 >= 1
C33: X4 >= 1
C34: X4 - 2 X5 >= 1
C35: X4 >= 1
```

```
STATISTICS - RUNTIME Thu Nov 03 10:26:27 2005
  xa VERSION 13.66 NT DLL    USABLE MEMORY 635.5 MBYTE
  ENV ID 1 SOLVE NUMBER 1
  VARIABLES 5
     0 LOWER, 0 FIXED, 0 UPPER, 0 FREE
  CONSTRAINTS 36
     35 GE, 0 EQ, 0 LE, 1 NULL/FREE, 0 RANGED.
     56 NON-ZEROS WORK 55,529,291
  MINIMIZATION.
  University of California, Davis - 1206701
  Civil & Environmental Engineering/Ines Ferreira 32420-21000


L P   O P T I M A L   S O L U T I O N ---> OBJECTIVE 13.00000
SOLVE 1 TIME 00:00:00  ITER 4  MEMORY USED   0.0%


File: RUNTIME                                    Thu Nov 03 10:26:27 2005 Page 1
SOLUTION REPORT - COLUMN ACTIVITY SOLVE NUMBER 1
NUMBER.COLUMNS AT..ACTIVITY..INPUT COST..LOWER LIMIT..UPPER LIMIT..REDUCED COST.
  0     X1     BS    14.00000     1.00000      .             NONE         .
  1     X2     BS     9.00000      .           .             NONE         .
  2     X3     BS     4.00000      .           .             NONE         .
  3     X4     BS     3.00000      .           .             NONE         .
  4     X5     BS     1.00000    -1.00000      .             NONE         .

File: RUNTIME                                    Thu Nov 03 10:26:27 2005 Page 2
CONSTRAINT REPORT - ROW ACTIVITY SOLVE NUMBER 1
NUMBER..ROW..AT...ACTIVITY..SLACK ACTIVITY..LOWER LIMIT..UPPER LIMIT..DUAL ACTIVITY
  0     OBJ  BS   13.00000    -13.00000         NONE         NONE       -1.00000
  1     C1   BS    5.00000     -4.00000       1.00000        NONE         .
  2     C2   LL    1.00000       .            1.00000        NONE        1.50000
  3     C3   BS    5.00000     -4.00000       1.00000        NONE         .
  4     C4   LL    1.00000       .            1.00000        NONE        6.50000
  5     C5   BS    2.00000     -1.00000       1.00000        NONE         .
  6     C6   LL    1.00000       .            1.00000        NONE        1.00000
  7     C7   BS    2.00000     -1.00000       1.00000        NONE         .
  8     C8   LL    1.00000       .            1.00000        NONE        1.00000
  9     C9   BS    1.00000       .            1.00000        NONE         .
 10     C10  BS    3.00000     -2.00000       1.00000        NONE         .
 11     C11  BS    9.00000     -8.00000       1.00000        NONE         .
 12     C12  BS    1.00000       .            1.00000        NONE         .
 13     C13  BS    9.00000     -8.00000       1.00000        NONE         .
 14     C14  BS    7.00000     -6.00000       1.00000        NONE         .
 15     C15  BS    9.00000     -8.00000       1.00000        NONE         .
 16     C16  BS    4.00000     -3.00000       1.00000        NONE         .
 17     C17  BS    4.00000     -3.00000       1.00000        NONE         .
 18     C18  BS    4.00000     -3.00000       1.00000        NONE         .
 19     C19  BS    2.00000     -1.00000       1.00000        NONE         .
 20     C20  BS    4.00000     -3.00000       1.00000        NONE         .
 21     C21  BS   14.00000    -13.00000       1.00000        NONE         .
 22     C22  BS    6.00000     -5.00000       1.00000        NONE         .
 23     C23  BS   14.00000    -13.00000       1.00000        NONE         .
 24     C24  BS   12.00000    -11.00000       1.00000        NONE         .
 25     C25  BS   14.00000    -13.00000       1.00000        NONE         .
 26     C26  BS    1.00000       .            1.00000        NONE         .
 27     C27  BS    1.00000       .            1.00000        NONE         .
 28     C28  BS    1.00000       .            1.00000        NONE         .
 29     C29  BS    1.00000       .            1.00000        NONE         .
 30     C30  BS    1.00000       .            1.00000        NONE         .
 31     C31  BS    3.00000     -2.00000       1.00000        NONE         .
```

32	C32	BS	3.00000	-2.00000	1.00000	NONE	.
33	C33	BS	3.00000	-2.00000	1.00000	NONE	.
34	C34	LL	1.00000	.	1.00000	NONE	3.00000
35	C35	BS	3.00000	-2.00000	1.00000	NONE	.

Appendix A-6: Preprocessor Output for Example 4

```
Rows 195 Columns 13 NonZeros 259 A's 1stDimSize 500
Minimize Solve Number 1
OBJ: X1 - X13

Constraints
C1: X10 - X11 >= 1
C2: X8 - X9 >= 1
C3: X7 - X8 >= 1
C4: X2 - X3 >= 1
C5: X6 - X7 >= 1
C6: X9 - X10 >= 1
C7: X5 - X6 >= 1
C8: X1 - X2 >= 1
C9: X3 - X4 >= 1
C10: X11 - X12 >= 1
C11: X4 - X5 >= 1
C12: X12 - X13 >= 1
C13: X13 >= 1
C14: X10 - X11 - X12 - X13 >= 1
C15: X8 - 0.5 X11 - X12 - 0.5 X13 >= 1
C16: X7 - 0.5 X11 - X12 - 0.5 X13 >= 1
C17: X2 - X3 - X4 - X5 - X6 - X9 - X11 - X12 - X13 >= 1
C18: X6 - X9 - 0.5 X11 - X12 - 0.5 X13 >= 1
C19: X9 - 0.5 X11 - X12 - 0.5 X13 >= 1
C20: X5 - X11 - X12 - X13 >= 1
C21: X1 - X3 - X4 - X11 - X12 - X13 >= 1
C22: X3 - 0.5 X11 >= 1
C23: X11 >= 1
C24: X4 - X12 - 0.5 X13 >= 1
C25: X12 - X13 >= 1
C26: X13 >= 1
C27: X10 >= 1
C28: X10 >= 1
C29: X10 >= 1
C30: X10 >= 1
C31: X10 >= 1
C32: X10 >= 1
C33: X10 >= 1
C34: X10 >= 1
C35: X10 >= 1
C36: X10 >= 1
C37: X10 >= 1
C38: X10 >= 1
C39: X10 >= 1
C40: X8 >= 1
C41: X8 >= 1
C42: X8 >= 1
C43: X8 >= 1
C44: X8 >= 1
C45: X8 >= 1
C46: X8 >= 1
C47: X8 >= 1
C48: X8 >= 1
C49: X8 >= 1
```

```
C50: X8 >= 1
C51: X8 >= 1
C52: X8 >= 1
C53: X7 >= 1
C54: X7 - 2 X8 >= 1
C55: X7 >= 1
C56: X7 >= 1
C57: X7 >= 1
C58: X7 >= 1
C59: X7 >= 1
C60: X7 >= 1
C61: X7 >= 1
C62: X7 >= 1
C63: X7 >= 1
C64: X7 >= 1
C65: X7 >= 1
C66: X2 >= 1
C67: X2 >= 1
C68: X2 >= 1
C69: X2 >= 1
C70: X2 >= 1
C71: X2 >= 1
C72: X2 >= 1
C73: X2 >= 1
C74: X2 >= 1
C75: X2 >= 1
C76: X2 >= 1
C77: X2 >= 1
C78: X2 >= 1
C79: X6 >= 1
C80: X6 >= 1
C81: X6 >= 1
C82: X6 >= 1
C83: X6 >= 1
C84: X6 >= 1
C85: X6 >= 1
C86: X6 >= 1
C87: X6 >= 1
C88: X6 >= 1
C89: X6 >= 1
C90: X6 >= 1
C91: X6 >= 1
C92: X9 >= 1
C93: X9 >= 1
C94: X9 >= 1
C95: X9 >= 1
C96: X9 >= 1
C97: X9 >= 1
C98: X9 >= 1
C99: X9 >= 1
C100: X9 >= 1
C101: X9 >= 1
C102: X9 >= 1
C103: X9 >= 1
C104: X9 >= 1
C105: X5 >= 1
C106: X5 >= 1
C107: X5 >= 1
C108: X5 >= 1
C109: X5 - 2 X6 >= 1
C110: X5 - 2 X9 >= 1
C111: X5 >= 1
C112: X5 >= 1
```

```
C113: X5 >= 1
C114: X5 >= 1
C115: X5 >= 1
C116: X5 >= 1
C117: X5 >= 1
C118: X1 >= 1
C119: X1 - 2 X8 >= 1
C120: X1 - 2 X7 >= 1
C121: X1 >= 1
C122: X1 - 2 X6 >= 1
C123: X1 - 2 X9 >= 1
C124: X1 >= 1
C125: X1 >= 1
C126: X1 >= 1
C127: X1 >= 1
C128: X1 >= 1
C129: X1 >= 1
C130: X1 >= 1
C131: X3 >= 1
C132: X3 - 2 X8 >= 1
C133: X3 - 2 X7 >= 1
C134: X3 >= 1
C135: X3 - 2 X6 >= 1
C136: X3 - 2 X9 >= 1
C137: X3 >= 1
C138: X3 >= 1
C139: X3 >= 1
C140: X3 >= 1
C141: X3 >= 1
C142: X3 >= 1
C143: X3 >= 1
C144: X11 >= 1
C145: X11 >= 1
C146: X11 >= 1
C147: X11 >= 1
C148: X11 >= 1
C149: X11 >= 1
C150: X11 >= 1
C151: X11 >= 1
C152: X11 >= 1
C153: X11 >= 1
C154: X11 >= 1
C155: X11 >= 1
C156: X11 >= 1
C157: X4 >= 1
C158: X4 - 2 X8 >= 1
C159: X4 - 2 X7 >= 1
C160: X4 >= 1
C161: X4 - 2 X6 >= 1
C162: X4 - 2 X9 >= 1
C163: X4 >= 1
C164: X4 >= 1
C165: X4 >= 1
C166: X4 >= 1
C167: X4 >= 1
C168: X4 >= 1
C169: X4 >= 1
C170: X12 >= 1
C171: X12 >= 1
C172: X12 >= 1
C173: X12 >= 1
C174: X12 >= 1
C175: X12 >= 1
```

```
C176: X12 >= 1
C177: X12 >= 1
C178: X12 >= 1
C179: X12 >= 1
C180: X12 >= 1
C181: X12 >= 1
C182: X12 >= 1
C183: X13 >= 1
C184: X13 >= 1
C185: X13 >= 1
C186: X13 >= 1
C187: X13 >= 1
C188: X13 >= 1
C189: X13 >= 1
C190: X13 >= 1
C191: X13 >= 1
C192: X13 >= 1
C193: X13 >= 1
C194: X13 >= 1
C195: X13 >= 1


STATISTICS - RUNTIME Mon Oct 17 11:36:47 2005
  xa VERSION 13.66 NT DLL    USABLE MEMORY 635.5 MBYTE
  ENV ID 1 SOLVE NUMBER 1
  VARIABLES 13
     0 LOWER, 0 FIXED, 0 UPPER, 0 FREE
  CONSTRAINTS 196
     195 GE, 0 EQ, 0 LE, 1 NULL/FREE, 0 RANGED.
     261 NON-ZEROS WORK 55,524,742
  MINIMIZATION.
  University of California, Davis - 1206701
  Civil & Environmental Engineering/Ines Ferreira 32420-21000


L P   O P T I M A L   S O L U T I O N ---> OBJECTIVE 161.00000
SOLVE 1 TIME 00:00:00  ITER 10  MEMORY USED   0.0%


File: RUNTIME                                   Mon Oct 17 11:36:47 2005 Page 1
SOLUTION REPORT - COLUMN ACTIVITY SOLVE NUMBER 1
NUMBER.COLUMNS  AT .ACTIVITY..INPUT COST..LOWER LIMIT.UPPER LIMIT.REDUCED COST.
   0    X1      BS   162.00000     1.00000        .               NONE          .
   1    X2      BS   161.00000      .             .               NONE          .
   2    X3      BS    43.00000      .             .               NONE          .
   3    X4      BS    42.00000      .             .               NONE          .
   4    X5      BS    41.00000      .             .               NONE          .
   5    X6      BS    20.00000      .             .               NONE          .
   6    X7      BS    19.00000      .             .               NONE          .
   7    X8      BS     9.00000      .             .               NONE          .
   8    X9      BS     8.00000      .             .               NONE          .
   9    X10     BS     7.00000      .             .               NONE          .
  10    X11     BS     3.00000      .             .               NONE          .
  11    X12     BS     2.00000      .             .               NONE          .
  12    X13     BS     1.00000    -1.00000        .               NONE          .

File: RUNTIME                                       Mon Oct 17 11:36:47 2005 Page 2
CONSTRAINT REPORT - ROW ACTIVITY SOLVE NUMBER 1
NUMBER..ROW.. AT...ACTIVITY.SLACK ACTIVITY..LOWER LIMIT..UPPER LIMIT..DUAL ACTIVITY
   0    OBJ   BS  161.00000   -161.00000        NONE          NONE       -1.00000
   1    C1    BS    4.00000     -3.00000      1.00000         NONE          .
   2    C2    LL    1.00000       .           1.00000         NONE       14.00000
   3    C3    BS   10.00000     -9.00000      1.00000         NONE          .
```

4	C4	BS	118.00000	-117.00000	1.00000	NONE	.
5	C5	LL	1.00000	.	1.00000	NONE	7.00000
6	C6	LL	1.00000	.	1.00000	NONE	15.00000
7	C7	BS	21.00000	-20.00000	1.00000	NONE	.
8	C8	LL	1.00000	.	1.00000	NONE	1.00000
9	C9	LL	1.00000	.	1.00000	NONE	1.00000
10	C10	LL	1.00000	.	1.00000	NONE	16.00000
11	C11	LL	1.00000	.	1.00000	NONE	2.00000
12	C12	LL	1.00000	.	1.00000	NONE	32.00000
13	C13	LL	1.00000	.	1.00000	NONE	47.00000
14	C14	LL	1.00000	.	1.00000	NONE	15.00000
15	C15	BS	5.00000	-4.00000	1.00000	NONE	.
16	C16	BS	15.00000	-14.00000	1.00000	NONE	.
17	C17	LL	1.00000	.	1.00000	NONE	1.00000
18	C18	BS	8.00000	-7.00000	1.00000	NONE	.
19	C19	BS	4.00000	-3.00000	1.00000	NONE	.
20	C20	BS	35.00000	-34.00000	1.00000	NONE	.
21	C21	BS	71.00000	-70.00000	1.00000	NONE	.
22	C22	BS	41.50000	-40.50000	1.00000	NONE	.
23	C23	BS	3.00000	-2.00000	1.00000	NONE	.
24	C24	BS	39.50000	-38.50000	1.00000	NONE	.
25	C25	BS	1.00000	.	1.00000	NONE	.
26	C26	BS	1.00000	.	1.00000	NONE	.
27	C27	BS	7.00000	-6.00000	1.00000	NONE	.
28	C28	BS	7.00000	-6.00000	1.00000	NONE	.
29	C29	BS	7.00000	-6.00000	1.00000	NONE	.
30	C30	BS	7.00000	-6.00000	1.00000	NONE	.
31	C31	BS	7.00000	-6.00000	1.00000	NONE	.
32	C32	BS	7.00000	-6.00000	1.00000	NONE	.
33	C33	BS	7.00000	-6.00000	1.00000	NONE	.
34	C34	BS	7.00000	-6.00000	1.00000	NONE	.
35	C35	BS	7.00000	-6.00000	1.00000	NONE	.
36	C36	BS	7.00000	-6.00000	1.00000	NONE	.
37	C37	BS	7.00000	-6.00000	1.00000	NONE	.
38	C38	BS	7.00000	-6.00000	1.00000	NONE	.
39	C39	BS	7.00000	-6.00000	1.00000	NONE	.
40	C40	BS	9.00000	-8.00000	1.00000	NONE	.
41	C41	BS	9.00000	-8.00000	1.00000	NONE	.
42	C42	BS	9.00000	-8.00000	1.00000	NONE	.
43	C43	BS	9.00000	-8.00000	1.00000	NONE	.
44	C44	BS	9.00000	-8.00000	1.00000	NONE	.
45	C45	BS	9.00000	-8.00000	1.00000	NONE	.
46	C46	BS	9.00000	-8.00000	1.00000	NONE	.
47	C47	BS	9.00000	-8.00000	1.00000	NONE	.
48	C48	BS	9.00000	-8.00000	1.00000	NONE	.
49	C49	BS	9.00000	-8.00000	1.00000	NONE	.
50	C50	BS	9.00000	-8.00000	1.00000	NONE	.
51	C51	BS	9.00000	-8.00000	1.00000	NONE	.
52	C52	BS	9.00000	-8.00000	1.00000	NONE	.
53	C53	BS	19.00000	-18.00000	1.00000	NONE	.
54	C54	LL	1.00000	.	1.00000	NONE	7.00000
55	C55	BS	19.00000	-18.00000	1.00000	NONE	.
56	C56	BS	19.00000	-18.00000	1.00000	NONE	.
57	C57	BS	19.00000	-18.00000	1.00000	NONE	.
58	C58	BS	19.00000	-18.00000	1.00000	NONE	.
59	C59	BS	19.00000	-18.00000	1.00000	NONE	.
60	C60	BS	19.00000	-18.00000	1.00000	NONE	.
61	C61	BS	19.00000	-18.00000	1.00000	NONE	.
62	C62	BS	19.00000	-18.00000	1.00000	NONE	.
63	C63	BS	19.00000	-18.00000	1.00000	NONE	.
64	C64	BS	19.00000	-18.00000	1.00000	NONE	.
65	C65	BS	19.00000	-18.00000	1.00000	NONE	.
66	C66	BS	161.00000	-160.00000	1.00000	NONE	.

67	C67	BS	161.00000	-160.00000	1.00000	NONE	.
68	C68	BS	161.00000	-160.00000	1.00000	NONE	.
69	C69	BS	161.00000	-160.00000	1.00000	NONE	.
70	C70	BS	161.00000	-160.00000	1.00000	NONE	.
71	C71	BS	161.00000	-160.00000	1.00000	NONE	.
72	C72	BS	161.00000	-160.00000	1.00000	NONE	.
73	C73	BS	161.00000	-160.00000	1.00000	NONE	.
74	C74	BS	161.00000	-160.00000	1.00000	NONE	.
75	C75	BS	161.00000	-160.00000	1.00000	NONE	.
76	C76	BS	161.00000	-160.00000	1.00000	NONE	.
77	C77	BS	161.00000	-160.00000	1.00000	NONE	.
78	C78	BS	161.00000	-160.00000	1.00000	NONE	.
79	C79	BS	20.00000	-19.00000	1.00000	NONE	.
80	C80	BS	20.00000	-19.00000	1.00000	NONE	.
81	C81	BS	20.00000	-19.00000	1.00000	NONE	.
82	C82	BS	20.00000	-19.00000	1.00000	NONE	.
83	C83	BS	20.00000	-19.00000	1.00000	NONE	.
84	C84	BS	20.00000	-19.00000	1.00000	NONE	.
85	C85	BS	20.00000	-19.00000	1.00000	NONE	.
86	C86	BS	20.00000	-19.00000	1.00000	NONE	.
87	C87	BS	20.00000	-19.00000	1.00000	NONE	.
88	C88	BS	20.00000	-19.00000	1.00000	NONE	.
89	C89	BS	20.00000	-19.00000	1.00000	NONE	.
90	C90	BS	20.00000	-19.00000	1.00000	NONE	.
91	C91	BS	20.00000	-19.00000	1.00000	NONE	.
92	C92	BS	8.00000	-7.00000	1.00000	NONE	.
93	C93	BS	8.00000	-7.00000	1.00000	NONE	.
94	C94	BS	8.00000	-7.00000	1.00000	NONE	.
95	C95	BS	8.00000	-7.00000	1.00000	NONE	.
96	C96	BS	8.00000	-7.00000	1.00000	NONE	.
97	C97	BS	8.00000	-7.00000	1.00000	NONE	.
98	C98	BS	8.00000	-7.00000	1.00000	NONE	.
99	C99	BS	8.00000	-7.00000	1.00000	NONE	.
100	C100	BS	8.00000	-7.00000	1.00000	NONE	.
101	C101	BS	8.00000	-7.00000	1.00000	NONE	.
102	C102	BS	8.00000	-7.00000	1.00000	NONE	.
103	C103	BS	8.00000	-7.00000	1.00000	NONE	.
104	C104	BS	8.00000	-7.00000	1.00000	NONE	.
105	C105	BS	41.00000	-40.00000	1.00000	NONE	.
106	C106	BS	41.00000	-40.00000	1.00000	NONE	.
107	C107	BS	41.00000	-40.00000	1.00000	NONE	.
108	C108	BS	41.00000	-40.00000	1.00000	NONE	.
109	C109	LL	1.00000	.	1.00000	NONE	3.00000
110	C110	BS	25.00000	-24.00000	1.00000	NONE	.
111	C111	BS	41.00000	-40.00000	1.00000	NONE	.
112	C112	BS	41.00000	-40.00000	1.00000	NONE	.
113	C113	BS	41.00000	-40.00000	1.00000	NONE	.
114	C114	BS	41.00000	-40.00000	1.00000	NONE	.
115	C115	BS	41.00000	-40.00000	1.00000	NONE	.
116	C116	BS	41.00000	-40.00000	1.00000	NONE	.
117	C117	BS	41.00000	-40.00000	1.00000	NONE	.
118	C118	BS	162.00000	-161.00000	1.00000	NONE	.
119	C119	BS	144.00000	-143.00000	1.00000	NONE	.
120	C120	BS	124.00000	-123.00000	1.00000	NONE	.
121	C121	BS	162.00000	-161.00000	1.00000	NONE	.
122	C122	BS	122.00000	-121.00000	1.00000	NONE	.
123	C123	BS	146.00000	-145.00000	1.00000	NONE	.
124	C124	BS	162.00000	-161.00000	1.00000	NONE	.
125	C125	BS	162.00000	-161.00000	1.00000	NONE	.
126	C126	BS	162.00000	-161.00000	1.00000	NONE	.
127	C127	BS	162.00000	-161.00000	1.00000	NONE	.
128	C128	BS	162.00000	-161.00000	1.00000	NONE	.
129	C129	BS	162.00000	-161.00000	1.00000	NONE	.

130	C130	BS	162.00000	-161.00000	1.00000	NONE	.
131	C131	BS	43.00000	-42.00000	1.00000	NONE	.
132	C132	BS	25.00000	-24.00000	1.00000	NONE	.
133	C133	BS	5.00000	-4.00000	1.00000	NONE	.
134	C134	BS	43.00000	-42.00000	1.00000	NONE	.
135	C135	BS	3.00000	-2.00000	1.00000	NONE	.
136	C136	BS	27.00000	-26.00000	1.00000	NONE	.
137	C137	BS	43.00000	-42.00000	1.00000	NONE	.
138	C138	BS	43.00000	-42.00000	1.00000	NONE	.
139	C139	BS	43.00000	-42.00000	1.00000	NONE	.
140	C140	BS	43.00000	-42.00000	1.00000	NONE	.
141	C141	BS	43.00000	-42.00000	1.00000	NONE	.
142	C142	BS	43.00000	-42.00000	1.00000	NONE	.
143	C143	BS	43.00000	-42.00000	1.00000	NONE	.
144	C144	BS	3.00000	-2.00000	1.00000	NONE	.
145	C145	BS	3.00000	-2.00000	1.00000	NONE	.
146	C146	BS	3.00000	-2.00000	1.00000	NONE	.
147	C147	BS	3.00000	-2.00000	1.00000	NONE	.
148	C148	BS	3.00000	-2.00000	1.00000	NONE	.
149	C149	BS	3.00000	-2.00000	1.00000	NONE	.
150	C150	BS	3.00000	-2.00000	1.00000	NONE	.
151	C151	BS	3.00000	-2.00000	1.00000	NONE	.
152	C152	BS	3.00000	-2.00000	1.00000	NONE	.
153	C153	BS	3.00000	-2.00000	1.00000	NONE	.
154	C154	BS	3.00000	-2.00000	1.00000	NONE	.
155	C155	BS	3.00000	-2.00000	1.00000	NONE	.
156	C156	BS	3.00000	-2.00000	1.00000	NONE	.
157	C157	BS	42.00000	-41.00000	1.00000	NONE	.
158	C158	BS	24.00000	-23.00000	1.00000	NONE	.
159	C159	BS	4.00000	-3.00000	1.00000	NONE	.
160	C160	BS	42.00000	-41.00000	1.00000	NONE	.
161	C161	BS	2.00000	-1.00000	1.00000	NONE	.
162	C162	BS	26.00000	-25.00000	1.00000	NONE	.
163	C163	BS	42.00000	-41.00000	1.00000	NONE	.
164	C164	BS	42.00000	-41.00000	1.00000	NONE	.
165	C165	BS	42.00000	-41.00000	1.00000	NONE	.
166	C166	BS	42.00000	-41.00000	1.00000	NONE	.
167	C167	BS	42.00000	-41.00000	1.00000	NONE	.
168	C168	BS	42.00000	-41.00000	1.00000	NONE	.
169	C169	BS	42.00000	-41.00000	1.00000	NONE	.
170	C170	BS	2.00000	-1.00000	1.00000	NONE	.
171	C171	BS	2.00000	-1.00000	1.00000	NONE	.
172	C172	BS	2.00000	-1.00000	1.00000	NONE	.
173	C173	BS	2.00000	-1.00000	1.00000	NONE	.
174	C174	BS	2.00000	-1.00000	1.00000	NONE	.
175	C175	BS	2.00000	-1.00000	1.00000	NONE	.
176	C176	BS	2.00000	-1.00000	1.00000	NONE	.
177	C177	BS	2.00000	-1.00000	1.00000	NONE	.
178	C178	BS	2.00000	-1.00000	1.00000	NONE	.
179	C179	BS	2.00000	-1.00000	1.00000	NONE	.
180	C180	BS	2.00000	-1.00000	1.00000	NONE	.
181	C181	BS	2.00000	-1.00000	1.00000	NONE	.
182	C182	BS	2.00000	-1.00000	1.00000	NONE	.
183	C183	BS	1.00000	.	1.00000	NONE	.
184	C184	BS	1.00000	.	1.00000	NONE	.
185	C185	BS	1.00000	.	1.00000	NONE	.
186	C186	BS	1.00000	.	1.00000	NONE	.
187	C187	BS	1.00000	.	1.00000	NONE	.
188	C188	BS	1.00000	.	1.00000	NONE	.
189	C189	BS	1.00000	.	1.00000	NONE	.
190	C190	BS	1.00000	.	1.00000	NONE	.
191	C191	BS	1.00000	.	1.00000	NONE	.
192	C192	BS	1.00000	.	1.00000	NONE	.

193	C193	BS	1.00000	.	1.00000	NONE	.
194	C194	BS	1.00000	.	1.00000	NONE	.
195	C195	BS	1.00000	.	1.00000	NONE	.

APPENDIX B

Appendix B-1: Simplified Two River System Model, Run I

```
>> CALSIM Version 1.2.
This program is Copyright (C) 1998 State of California, all rights reserved
2001D10A
CLP options: MATLIST both
>> These extra XA options were obtained:
CLP options: MUTE NO LISTINPUT NO
>> Solving at date  1/31, of water year 1922
Maximize Solve Number 1
OBJ: OBJ1 + OBJ0

Constraints
1OBJECTIVE:  - OBJ1 = 0
0OBJECTIVE:  - OBJ0 + 343809 S1_1 + 6505.38 S1_2 + 3252.69 S1_3 + 1626.34 S1_4
   487.903 S1_5 + 2550 D30 + 2560 C30_MIF + 2550 D31 + 301524 S2_1
   6342.74 S2_2 + 3090.05 S2_3 + 1463.71 S2_4 + 162.634 S2_5 + 2550 D2
   2560 C2_MIF + 2550 D33 + 2550 D34A + 2550 C34A + 2550 D34B + 41797 S3_1
   6668.01 S3_2 + 6668.01 S3_3 + 650.538 S3_4 + 420 D3 + 41797 S4_1
   6668.01 S4_2 + 6668.01 S4_3 + 325.269 S4_4 + 420 D4 - 53669.4 S1_6 - 3400 F1
    - 34153.2 S2_6 - 3400 F2 - 550 C34B - 10571.2 S3_5 - 3400 F3 - 10571.2 S4_5
    - 3400 F4 = 0
0C2TOTAL/1:  - C2_MIF + C2 - C2_EXC = 0
0C2MINFLOW/1: C2_MIF <= 1000
0C30TOTAL/1:  - C30_MIF + C30 - C30_EXC = 0
0S1ZONE1/1: S1_1 <= 550
0S1ZONE2/1: S1_2 <= 1165
0S1ZONE3/1: S1_3 <= 785
0S1ZONE4/1: S1_4 <= 1100
0S1ZONE5/1: S1_5 <= 400
0S1ZONE6/1: S1_6 <= 552
0STORAGE1/1:  - S1_1 - S1_2 - S1_3 - S1_4 - S1_5 - S1_6 + S1 = 0
0MAXRELEASE1/1: C1 <= 12702.7
0S2ZONE1/1: S2_1 <= 29.6
0S2ZONE2/1: S2_2 <= 822.4
0S2ZONE3/1: S2_3 <= 1618
0S2ZONE4/1: S2_4 <= 530
0S2ZONE5/1: S2_5 <= 250
0S2ZONE6/1: S2_6 <= 308
0STORAGE2/1:  - S2_1 - S2_2 - S2_3 - S2_4 - S2_5 - S2_6 + S2 = 0
0MAXRELEASE2/1: C2 <= 50000
0S3ZONE1/1: S3_1 <= 45
0S3ZONE2/1: S3_2 <= 0
0S3ZONE3/1: S3_3 <= 455
0S3ZONE4/1: S3_4 <= 450
0S3ZONE5/1: S3_5 <= 22
0STORAGE3/1:  - S3_1 - S3_2 - S3_3 - S3_4 - S3_5 + S3 = 0
0MAXRELEASE3/1: D3 <= 14376
0S4ZONE1/1: S4_1 <= 55
0S4ZONE2/1: S4_2 <= 0
0S4ZONE3/1: S4_3 <= 445
0S4ZONE4/1: S4_4 <= 500
0S4ZONE5/1: S4_5 <= 67
0STORAGE4/1:  - S4_1 - S4_2 - S4_3 - S4_4 - S4_5 + S4 = 0
0MAXRELEASE4/1: D4 <= 14376
0CONTINUITY1/1:  - F1 - C1 - 16.2634 S1 = -9944.89
0CONTINUITY30/1:  - D30 + C1 - C30 = 0
0CONTINUITY2/1:  - D2 - F2 - C2 - 16.2634 S2 = -737.903
0CONTINUITY31/1:  - D31 + C30 - C31 = 0
0CONTINUITY32/1: C2 + C31 - C32 = 0
```

```
0CONTINUITY33/1:  - D33 + C32 - C33 = 0
0CONTINUITY34/1:  - D34A - C34A - D34B - C34B + C33 - D34C - D34D = 0
0CONTINUITY3/1:  - D3 - F3 + D34C - 16.2634 S3 = -731.855
0CONTINUITY4/1:  - D4 - F4 + D34D - 16.2634 S4 = -894.489
0SETMRDO/1: C34A <= 1000
0MEETC30MIN/1: C30_MIF <= 3500
0SETKESWICK_MIN/1: KESWICK_MIN = 3500

OBJ1 = FREE | OBJ0 = FREE | D30 <= 1000 | D31 <= 1000 | D2 <= 1000 | D33 = 0
D34A <= 1000 | C34A <= 210000 | D34B <= 1000 | D3 <= 1000 | D4 <= 1000 |
C34B <= 210000 | C1 <= 50000 | C2 <= 80000 | C30 <= 80000 | C31 <= 80000 |
C32 <= 80000 | C33 <= 80000 | D34C <= 4600 | D34D <= 6680 |
-999999 <= KESWICK_MIN <= 999999 |

Maximize Solve Number 1
OBJ1: OBJ - 1OBJECTIVE = FREE
OBJ0: OBJ - 0OBJECTIVE = FREE
S1_1: 343809 0OBJECTIVE + 0S1ZONE1/1 - 0STORAGE1/1
S1_2: 6505.38 0OBJECTIVE + 0S1ZONE2/1 - 0STORAGE1/1
S1_3: 3252.69 0OBJECTIVE + 0S1ZONE3/1 - 0STORAGE1/1
S1_4: 1626.34 0OBJECTIVE + 0S1ZONE4/1 - 0STORAGE1/1
S1_5: 487.903 0OBJECTIVE + 0S1ZONE5/1 - 0STORAGE1/1
D30: 2550 0OBJECTIVE - 0CONTINUITY30/1 <= 1000
C30_MIF: 2560 0OBJECTIVE - 0C30TOTAL/1 + 0MEETC30MIN/1
D31: 2550 0OBJECTIVE - 0CONTINUITY31/1 <= 1000
S2_1: 301524 0OBJECTIVE + 0S2ZONE1/1 - 0STORAGE2/1
S2_2: 6342.74 0OBJECTIVE + 0S2ZONE2/1 - 0STORAGE2/1
S2_3: 3090.05 0OBJECTIVE + 0S2ZONE3/1 - 0STORAGE2/1
S2_4: 1463.71 0OBJECTIVE + 0S2ZONE4/1 - 0STORAGE2/1
S2_5: 162.634 0OBJECTIVE + 0S2ZONE5/1 - 0STORAGE2/1
D2: 2550 0OBJECTIVE - 0CONTINUITY2/1 <= 1000
C2_MIF: 2560 0OBJECTIVE - 0C2TOTAL/1 + 0C2MINFLOW/1
D33: 2550 0OBJECTIVE - 0CONTINUITY33/1 = 0
D34A: 2550 0OBJECTIVE - 0CONTINUITY34/1 <= 1000
C34A: 2550 0OBJECTIVE - 0CONTINUITY34/1 + 0SETMRDO/1 <= 210000
D34B: 2550 0OBJECTIVE - 0CONTINUITY34/1 <= 1000
S3_1: 41797 0OBJECTIVE + 0S3ZONE1/1 - 0STORAGE3/1
S3_2: 6668.01 0OBJECTIVE + 0S3ZONE2/1 - 0STORAGE3/1
S3_3: 6668.01 0OBJECTIVE + 0S3ZONE3/1 - 0STORAGE3/1
S3_4: 650.538 0OBJECTIVE + 0S3ZONE4/1 - 0STORAGE3/1
D3: 420 0OBJECTIVE + 0MAXRELEASE3/1 - 0CONTINUITY3/1 <= 1000
S4_1: 41797 0OBJECTIVE + 0S4ZONE1/1 - 0STORAGE4/1
S4_2: 6668.01 0OBJECTIVE + 0S4ZONE2/1 - 0STORAGE4/1
S4_3: 6668.01 0OBJECTIVE + 0S4ZONE3/1 - 0STORAGE4/1
S4_4: 325.269 0OBJECTIVE + 0S4ZONE4/1 - 0STORAGE4/1
D4: 420 0OBJECTIVE + 0MAXRELEASE4/1 - 0CONTINUITY4/1 <= 1000
S1_6:  - 53669.4 0OBJECTIVE + 0S1ZONE6/1 - 0STORAGE1/1
F1:  - 3400 0OBJECTIVE - 0CONTINUITY1/1
S2_6:  - 34153.2 0OBJECTIVE + 0S2ZONE6/1 - 0STORAGE2/1
F2:  - 3400 0OBJECTIVE - 0CONTINUITY2/1
C34B:  - 550 0OBJECTIVE - 0CONTINUITY34/1 <= 210000
S3_5:  - 10571.2 0OBJECTIVE + 0S3ZONE5/1 - 0STORAGE3/1
F3:  - 3400 0OBJECTIVE - 0CONTINUITY3/1
S4_5:  - 10571.2 0OBJECTIVE + 0S4ZONE5/1 - 0STORAGE4/1
F4:  - 3400 0OBJECTIVE - 0CONTINUITY4/1
C1: 0MAXRELEASE1/1 - 0CONTINUITY1/1 + 0CONTINUITY30/1 <= 50000
C2: 0C2TOTAL/1 + 0MAXRELEASE2/1 - 0CONTINUITY2/1 + 0CONTINUITY32/1 <= 80000
C2_EXC:  - 0C2TOTAL/1
C30: 0C30TOTAL/1 - 0CONTINUITY30/1 + 0CONTINUITY31/1 <= 80000
C30_EXC:  - 0C30TOTAL/1
C31:  - 0CONTINUITY31/1 + 0CONTINUITY32/1 <= 80000
C32:  - 0CONTINUITY32/1 + 0CONTINUITY33/1 <= 80000
C33:  - 0CONTINUITY33/1 + 0CONTINUITY34/1 <= 80000
```

```
D34C:  - 0CONTINUITY34/1 + 0CONTINUITY3/1 <= 4600
D34D:  - 0CONTINUITY34/1 + 0CONTINUITY4/1 <= 6680
S1: 0STORAGE1/1 - 16.2634 0CONTINUITY1/1
S2: 0STORAGE2/1 - 16.2634 0CONTINUITY2/1
S3: 0STORAGE3/1 - 16.2634 0CONTINUITY3/1
S4: 0STORAGE4/1 - 16.2634 0CONTINUITY4/1
KESWICK_MIN: 0SETKESWICK_MIN/1 >= -999999 <= 999999

1OBJECTIVE = 0 | 0OBJECTIVE = 0 | 0C2TOTAL/1 = 0 | 0C2MINFLOW/1 <= 1000 |
0C30TOTAL/1 = 0 | 0S1ZONE1/1 <= 550 | 0S1ZONE2/1 <= 1165 | 0S1ZONE3/1 <= 785
0S1ZONE4/1 <= 1100 | 0S1ZONE5/1 <= 400 | 0S1ZONE6/1 <= 552 | 0STORAGE1/1 = 0
0MAXRELEASE1/1 <= 12702.7 | 0S2ZONE1/1 <= 29.6 | 0S2ZONE2/1 <= 822.4 |
0S2ZONE3/1 <= 1618 | 0S2ZONE4/1 <= 530 | 0S2ZONE5/1 <= 250 | 0S2ZONE6/1 <= 308
0STORAGE2/1 = 0 | 0MAXRELEASE2/1 <= 50000 | 0S3ZONE1/1 <= 45 | 0S3ZONE2/1 <= 0
0S3ZONE3/1 <= 455 | 0S3ZONE4/1 <= 450 | 0S3ZONE5/1 <= 22 | 0STORAGE3/1 = 0
0MAXRELEASE3/1 <= 14376 | 0S4ZONE1/1 <= 55 | 0S4ZONE2/1 <= 0 | 0S4ZONE3/1 <= 445
0S4ZONE4/1 <= 500 | 0S4ZONE5/1 <= 67 | 0STORAGE4/1 = 0 | 0MAXRELEASE4/1 <= 14376
0CONTINUITY1/1 = -9944.89 | 0CONTINUITY30/1 = 0 | 0CONTINUITY2/1 = -737.903
0CONTINUITY31/1 = 0 | 0CONTINUITY32/1 = 0 | 0CONTINUITY33/1 = 0 |
0CONTINUITY34/1 = 0 | 0CONTINUITY3/1 = -731.855 | 0CONTINUITY4/1 = -894.489
0SETMRDO/1 <= 1000 | 0MEETC30MIN/1 <= 3500 | 0SETKESWICK_MIN/1 = 3500 |
```

Appendix B-2: Simplified Two River System Model, Run II

```
>> CALSIM Version 1.2.
This program is Copyright (C) 1998 State of California, all rights reserved
2001D10A
CLP options: MATLIST both
>> These extra XA options were obtained:
CLP options: MUTE NO LISTINPUT NO
>> Solving at date  1/31, of water year 1922
Maximize Solve Number 1
OBJ: OBJ1 + OBJ0

Constraints
1OBJECTIVE:  - OBJ1 = 0
0OBJECTIVE:  - OBJ0 + 343809 S1_1 + 6505.38 S1_2 + 3252.69 S1_3 + 1626.34 S1_4
   487.903 S1_5 + 2550 D30 + 2560 C30_MIF + 2550 D31 + 301524 S2_1
   6342.74 S2_2 + 3090.05 S2_3 + 1463.71 S2_4 + 162.634 S2_5 + 2550 D2
   2560 C2_MIF + 2550 D33 + 2550 D34A + 2550 C34A + 2550 D34B + 41797 S3_1
   6668.01 S3_2 + 6668.01 S3_3 + 650.538 S3_4 + 420 D3 + 41797 S4_1
   6668.01 S4_2 + 6668.01 S4_3 + 325.269 S4_4 + 420 D4 - 53669.4 S1_6 - 3400 F1
    - 34153.2 S2_6 - 3400 F2 - 550 C34B - 10571.2 S3_5 - 3400 F3 - 10571.2 S4_5
    - 3400 F4 = 0
0C2TOTAL/1:  - C2_MIF + C2 - C2_EXC = 0
0C2MINFLOW/1: C2_MIF <= 1000
0C30TOTAL/1:  - C30_MIF + C30 - C30_EXC = 0
0S1ZONE1/1: S1_1 <= 550
0S1ZONE2/1: S1_2 <= 1165
0S1ZONE3/1: S1_3 <= 785
0S1ZONE4/1: S1_4 <= 1100
0S1ZONE5/1: S1_5 <= 400
0S1ZONE6/1: S1_6 <= 552
0STORAGE1/1:  - S1_1 - S1_2 - S1_3 - S1_4 - S1_5 - S1_6 + S1 = 0
0AREA1/1:  - 8.91348 S1 + A1 = 2099.39
0EVAP1/1: 61.4876 E1 - 0.220781 A1 = 1545.86
0MAXRELEASE1/1: C1 <= 12702.7
0S2ZONE1/1: S2_1 <= 29.6
0S2ZONE2/1: S2_2 <= 822.4
0S2ZONE3/1: S2_3 <= 1618
0S2ZONE4/1: S2_4 <= 530
0S2ZONE5/1: S2_5 <= 250
```

```
0S2ZONE6/1: S2_6 <= 308
0STORAGE2/1:  - S2_1 - S2_2 - S2_3 - S2_4 - S2_5 - S2_6 + S2 = 0
0AREA2/1:  - 14.1896 S2 + A2 = 172.154
0EVAP2/1: 61.4876 E2 - 0.117924 A2 = 70.4996
0MAXRELEASE2/1: C2 <= 50000
0S3ZONE1/1: S3_1 <= 45
0S3ZONE2/1: S3_2 <= 0
0S3ZONE3/1: S3_3 <= 455
0S3ZONE4/1: S3_4 <= 450
0S3ZONE5/1: S3_5 <= 22
0STORAGE3/1:  - S3_1 - S3_2 - S3_3 - S3_4 - S3_5 + S3 = 0
0AREA3/1:  - 13.0406 S3 + A3 = 1190.72
0EVAP3/1: 61.4876 E3 - 0.301332 A3 = 535.631
0MAXRELEASE3/1: D3 <= 14376
0S4ZONE1/1: S4_1 <= 55
0S4ZONE2/1: S4_2 <= 0
0S4ZONE3/1: S4_3 <= 445
0S4ZONE4/1: S4_4 <= 500
0S4ZONE5/1: S4_5 <= 67
0STORAGE4/1:  - S4_1 - S4_2 - S4_3 - S4_4 - S4_5 + S4 = 0
0AREA4/1:  - 13.0303 S4 + A4 = 1316.73
0EVAP4/1: 61.4876 E4 - 0.301332 A4 = 612.726
0MAXRELEASE4/1: D4 <= 14376
0CONTINUITY1/1:  - F1 - C1 - 16.2634 S1 - E1 = -9944.89
0CONTINUITY30/1:  - D30 + C1 - C30 = 0
0CONTINUITY2/1:  - D2 - F2 - C2 - 16.2634 S2 - E2 = -737.903
0CONTINUITY31/1:  - D31 + C30 - C31 = 0
0CONTINUITY32/1: C2 + C31 - C32 = 0
0CONTINUITY33/1:  - D33 + C32 - C33 = 0
0CONTINUITY34/1:  - D34A - C34A - D34B - C34B + C33 - D34C - D34D = 0
0CONTINUITY3/1:  - D3 - F3 + D34C - 16.2634 S3 - E3 = -731.855
0CONTINUITY4/1:  - D4 - F4 + D34D - 16.2634 S4 - E4 = -894.489
0SETMRDO/1: C34A <= 1000
0MEETC30MIN/1: C30_MIF <= 3500
0SETKESWICK_MIN/1: KESWICK_MIN = 3500

OBJ1 = FREE | OBJ0 = FREE | D30 <= 1000 | D31 <= 1000 | D2 <= 1000 | D33 = 0
D34A <= 1000 | C34A <= 210000 | D34B <= 1000 | D3 <= 1000 | D4 <= 1000 |
C34B <= 210000 | C1 <= 50000 | C2 <= 80000 | C30 <= 80000 | C31 <= 80000 |
C32 <= 80000 | C33 <= 80000 | D34C <= 4600 | D34D <= 6680 | E1 = FREE |
E2 = FREE | E3 = FREE | E4 = FREE | -999999 <= KESWICK_MIN <= 999999 |

Maximize Solve Number 1
OBJ1: OBJ - 1OBJECTIVE = FREE
OBJ0: OBJ - 0OBJECTIVE = FREE
S1_1: 343809 0OBJECTIVE + 0S1ZONE1/1 - 0STORAGE1/1
S1_2: 6505.38 0OBJECTIVE + 0S1ZONE2/1 - 0STORAGE1/1
S1_3: 3252.69 0OBJECTIVE + 0S1ZONE3/1 - 0STORAGE1/1
S1_4: 1626.34 0OBJECTIVE + 0S1ZONE4/1 - 0STORAGE1/1
S1_5: 487.903 0OBJECTIVE + 0S1ZONE5/1 - 0STORAGE1/1
D30: 2550 0OBJECTIVE - 0CONTINUITY30/1 <= 1000
C30_MIF: 2560 0OBJECTIVE - 0C30TOTAL/1 + 0MEETC30MIN/1
D31: 2550 0OBJECTIVE - 0CONTINUITY31/1 <= 1000
S2_1: 301524 0OBJECTIVE + 0S2ZONE1/1 - 0STORAGE2/1
S2_2: 6342.74 0OBJECTIVE + 0S2ZONE2/1 - 0STORAGE2/1
S2_3: 3090.05 0OBJECTIVE + 0S2ZONE3/1 - 0STORAGE2/1
S2_4: 1463.71 0OBJECTIVE + 0S2ZONE4/1 - 0STORAGE2/1
S2_5: 162.634 0OBJECTIVE + 0S2ZONE5/1 - 0STORAGE2/1
D2: 2550 0OBJECTIVE - 0CONTINUITY2/1 <= 1000
C2_MIF: 2560 0OBJECTIVE - 0C2TOTAL/1 + 0C2MINFLOW/1
D33: 2550 0OBJECTIVE - 0CONTINUITY33/1 = 0
D34A: 2550 0OBJECTIVE - 0CONTINUITY34/1 <= 1000
C34A: 2550 0OBJECTIVE - 0CONTINUITY34/1 + 0SETMRDO/1 <= 210000
```

```
D34B: 2550 0OBJECTIVE - 0CONTINUITY34/1 <= 1000
S3_1: 41797 0OBJECTIVE + 0S3ZONE1/1 - 0STORAGE3/1
S3_2: 6668.01 0OBJECTIVE + 0S3ZONE2/1 - 0STORAGE3/1
S3_3: 6668.01 0OBJECTIVE + 0S3ZONE3/1 - 0STORAGE3/1
S3_4: 650.538 0OBJECTIVE + 0S3ZONE4/1 - 0STORAGE3/1
D3: 420 0OBJECTIVE + 0MAXRELEASE3/1 - 0CONTINUITY3/1 <= 1000
S4_1: 41797 0OBJECTIVE + 0S4ZONE1/1 - 0STORAGE4/1
S4_2: 6668.01 0OBJECTIVE + 0S4ZONE2/1 - 0STORAGE4/1
S4_3: 6668.01 0OBJECTIVE + 0S4ZONE3/1 - 0STORAGE4/1
S4_4: 325.269 0OBJECTIVE + 0S4ZONE4/1 - 0STORAGE4/1
D4: 420 0OBJECTIVE + 0MAXRELEASE4/1 - 0CONTINUITY4/1 <= 1000
S1_6:  - 53669.4 0OBJECTIVE + 0S1ZONE6/1 - 0STORAGE1/1
F1:  - 3400 0OBJECTIVE - 0CONTINUITY1/1
S2_6:  - 34153.2 0OBJECTIVE + 0S2ZONE6/1 - 0STORAGE2/1
F2:  - 3400 0OBJECTIVE - 0CONTINUITY2/1
C34B:  - 550 0OBJECTIVE - 0CONTINUITY34/1 <= 210000
S3_5:  - 10571.2 0OBJECTIVE + 0S3ZONE5/1 - 0STORAGE3/1
F3:  - 3400 0OBJECTIVE - 0CONTINUITY3/1
S4_5:  - 10571.2 0OBJECTIVE + 0S4ZONE5/1 - 0STORAGE4/1
F4:  - 3400 0OBJECTIVE - 0CONTINUITY4/1
C1: 0MAXRELEASE1/1 - 0CONTINUITY1/1 + 0CONTINUITY30/1 <= 50000
C2: 0C2TOTAL/1 + 0MAXRELEASE2/1 - 0CONTINUITY2/1 + 0CONTINUITY32/1 <= 80000
C2_EXC:  - 0C2TOTAL/1
C30: 0C30TOTAL/1 - 0CONTINUITY30/1 + 0CONTINUITY31/1 <= 80000
C30_EXC:  - 0C30TOTAL/1
C31:  - 0CONTINUITY31/1 + 0CONTINUITY32/1 <= 80000
C32:  - 0CONTINUITY32/1 + 0CONTINUITY33/1 <= 80000
C33:  - 0CONTINUITY33/1 + 0CONTINUITY34/1 <= 80000
D34C:  - 0CONTINUITY34/1 + 0CONTINUITY3/1 <= 4600
D34D:  - 0CONTINUITY34/1 + 0CONTINUITY4/1 <= 6680
S1: 0STORAGE1/1 - 8.91348 0AREA1/1 - 16.2634 0CONTINUITY1/1
E1: 61.4876 0EVAP1/1 - 0CONTINUITY1/1 = FREE
A1: 0AREA1/1 - 0.220781 0EVAP1/1
S2: 0STORAGE2/1 - 14.1896 0AREA2/1 - 16.2634 0CONTINUITY2/1
E2: 61.4876 0EVAP2/1 - 0CONTINUITY2/1 = FREE
A2: 0AREA2/1 - 0.117924 0EVAP2/1
S3: 0STORAGE3/1 - 13.0406 0AREA3/1 - 16.2634 0CONTINUITY3/1
E3: 61.4876 0EVAP3/1 - 0CONTINUITY3/1 = FREE
A3: 0AREA3/1 - 0.301332 0EVAP3/1
S4: 0STORAGE4/1 - 13.0303 0AREA4/1 - 16.2634 0CONTINUITY4/1
E4: 61.4876 0EVAP4/1 - 0CONTINUITY4/1 = FREE
A4: 0AREA4/1 - 0.301332 0EVAP4/1
KESWICK_MIN: 0SETKESWICK_MIN/1 >= -999999 <= 999999

1OBJECTIVE = 0 | 0OBJECTIVE = 0 | 0C2TOTAL/1 = 0 | 0C2MINFLOW/1 <= 1000 |
0C30TOTAL/1 = 0 | 0S1ZONE1/1 <= 550 | 0S1ZONE2/1 <= 1165 | 0S1ZONE3/1 <= 785
0S1ZONE4/1 <= 1100 | 0S1ZONE5/1 <= 400 | 0S1ZONE6/1 <= 552 | 0STORAGE1/1 = 0
0AREA1/1 = 2099.39 | 0EVAP1/1 = 1545.86 | 0MAXRELEASE1/1 <= 12702.7 |
0S2ZONE1/1 <= 29.6 | 0S2ZONE2/1 <= 822.4 | 0S2ZONE3/1 <= 1618 |
0S2ZONE4/1 <= 530 | 0S2ZONE5/1 <= 250 | 0S2ZONE6/1 <= 308 | 0STORAGE2/1 = 0
0AREA2/1 = 172.154 | 0EVAP2/1 = 70.4996 | 0MAXRELEASE2/1 <= 50000 |
0S3ZONE1/1 <= 45 | 0S3ZONE2/1 <= 0 | 0S3ZONE3/1 <= 455 | 0S3ZONE4/1 <= 450
0S3ZONE5/1 <= 22 | 0STORAGE3/1 = 0 | 0AREA3/1 = 1190.72 | 0EVAP3/1 = 535.631
0MAXRELEASE3/1 <= 14376 | 0S4ZONE1/1 <= 55 | 0S4ZONE2/1 <= 0 | 0S4ZONE3/1 <=
445
0S4ZONE4/1 <= 500 | 0S4ZONE5/1 <= 67 | 0STORAGE4/1 = 0 | 0AREA4/1 = 1316.73
0EVAP4/1 = 612.726 | 0MAXRELEASE4/1 <= 14376 | 0CONTINUITY1/1 = -9944.89 |
0CONTINUITY30/1 = 0 | 0CONTINUITY2/1 = -737.903 | 0CONTINUITY31/1 = 0 |
0CONTINUITY32/1 = 0 | 0CONTINUITY33/1 = 0 | 0CONTINUITY34/1 = 0 |
0CONTINUITY3/1 = -731.855 | 0CONTINUITY4/1 = -894.489 | 0SETMRDO/1 <= 1000
0MEETC30MIN/1 <= 3500 | 0SETKESWICK_MIN/1 = 3500 |
```

Appendix B-3– Simplified Two River System Model, Run III

```
>> CALSIM Version 1.2.
This program is Copyright (C) 1998 State of California, all rights reserved
2001D10A
CLP options: MATLIST both
>> These extra XA options were obtained:
CLP options: MUTE NO LISTINPUT NO
>> Solving at date  1/31, of water year 1922
Maximize Solve Number 1
OBJ: OBJ1 + OBJ0

Constraints
1OBJECTIVE:  - OBJ1 = 0
0OBJECTIVE:  - OBJ0 + 343809 S1_1 + 6505.38 S1_2 + 3252.69 S1_3 + 1626.34 S1_4
   487.903 S1_5 + 2550 D30 + 2560 C30_MIF + 2550 D31 + 301524 S2_1
   6342.74 S2_2 + 3090.05 S2_3 + 1463.71 S2_4 + 162.634 S2_5 + 2550 D2
   2560 C2_MIF + 2550 D33 + 2550 D34A + 2550 C34A + 2550 D34B + 41797 S3_1
   6668.01 S3_2 + 6668.01 S3_3 + 650.538 S3_4 + 420 D3 + 41797 S4_1
   6668.01 S4_2 + 6668.01 S4_3 + 325.269 S4_4 + 420 D4 - 53669.4 S1_6 - 3400 F1
    - 34153.2 S2_6 - 3400 F2 - 550 C34B - 10571.2 S3_5 - 3400 F3 - 10571.2 S4_5
    - 3400 F4 = 0
0C2TOTAL/1:  - C2_MIF + C2 - C2_EXC = 0
0C2MINFLOW/1: C2_MIF <= 1000
0C30TOTAL/1:  - C30_MIF + C30 - C30_EXC = 0
0S1ZONE1/1: S1_1 <= 550
0S1ZONE2/1: S1_2 <= 1165
0S1ZONE3/1: S1_3 <= 785
0S1ZONE4/1: S1_4 <= 1100
0S1ZONE5/1: S1_5 <= 400
0S1ZONE6/1: S1_6 <= 552
0STORAGE1/1:  - S1_1 - S1_2 - S1_3 - S1_4 - S1_5 - S1_6 + S1 = 0
0AREA1/1:  - 8.91348 S1 + A1 = 2099.39
0EVAP1/1: 61.4876 E1 - 0.220781 A1 = 1545.86
0MAXRELEASE1/1: C1 <= 12702.7
0S2ZONE1/1: S2_1 <= 29.6
0S2ZONE2/1: S2_2 <= 822.4
0S2ZONE3/1: S2_3 <= 1618
0S2ZONE4/1: S2_4 <= 530
0S2ZONE5/1: S2_5 <= 250
0S2ZONE6/1: S2_6 <= 308
0STORAGE2/1:  - S2_1 - S2_2 - S2_3 - S2_4 - S2_5 - S2_6 + S2 = 0
0AREA2/1:  - 14.1896 S2 + A2 = 172.154
0EVAP2/1: 61.4876 E2 - 0.117924 A2 = 70.4996
0MAXRELEASE2/1: C2 <= 50000
0S3ZONE1/1: S3_1 <= 45
0S3ZONE2/1: S3_2 <= 0
0S3ZONE3/1: S3_3 <= 455
0S3ZONE4/1: S3_4 <= 450
0S3ZONE5/1: S3_5 <= 22
0STORAGE3/1:  - S3_1 - S3_2 - S3_3 - S3_4 - S3_5 + S3 = 0
0AREA3/1:  - 13.0406 S3 + A3 = 1190.72
0EVAP3/1: 61.4876 E3 - 0.301332 A3 = 535.631
0MAXRELEASE3/1: D3 <= 14376
0S4ZONE1/1: S4_1 <= 55
0S4ZONE2/1: S4_2 <= 0
0S4ZONE3/1: S4_3 <= 445
0S4ZONE4/1: S4_4 <= 500
0S4ZONE5/1: S4_5 <= 67
0STORAGE4/1:  - S4_1 - S4_2 - S4_3 - S4_4 - S4_5 + S4 = 0
0AREA4/1:  - 13.0303 S4 + A4 = 1316.73
0EVAP4/1: 61.4876 E4 - 0.301332 A4 = 612.726
0MAXRELEASE4/1: D4 <= 14376
0CONTINUITY1/1:  - F1 - C1 - 16.2634 S1 - E1 = -9944.89
```

```
0CONTINUITY30/1:  - D30 + C1 - C30 = 0
0CONTINUITY2/1:  - D2 - F2 - C2 - 16.2634 S2 - E2 = -737.903
0CONTINUITY31/1:  - D31 + C30 - C31 = 0
0CONTINUITY32/1: C2 + C31 - C32 = 0
0CONTINUITY33/1:  - D33 + C32 - C33 = 0
0CONTINUITY34/1:  - D34A - C34A - D34B - C34B + C33 - D34C - D34D = 0
0CONTINUITY3/1:  - D3 - F3 + D34C - 16.2634 S3 - E3 = -731.855
0CONTINUITY4/1:  - D4 - F4 + D34D - 16.2634 S4 - E4 = -894.489
0SETMRDO/1: C34A <= 1000
0MEETC30MIN/1: C30_MIF <= 3500
0SETKESWICK_MIN/1: KESWICK_MIN = 3500
0EXPORTACTUAL_ALIAS/1:  - D34C - D34D + EXPORTACTUAL = 0
0INFLOW_ALIAS/1:  - C33 + INFLOW = 0
0EXPRATIO__ALIAS/1: EXPRATIO_ = 0.65
0FIND_MAX_EXPORT/1:  - 0.65 INFLOW + EIEXPCTRL = 0
0EXPORT_COMPLY/1: EXPORTACTUAL - EIEXPCTRL <= 0

OBJ1 = FREE | OBJ0 = FREE | D30 <= 1000 | D31 <= 1000 | D2 <= 1000 | D33 = 0
D34A <= 1000 | C34A <= 210000 | D34B <= 1000 | D3 <= 1000 | D4 <= 1000 |
C34B <= 210000 | C1 <= 50000 | C2 <= 80000 | C30 <= 80000 | C31 <= 80000 |
C32 <= 80000 | C33 <= 80000 | D34C <= 4600 | D34D <= 6680 | E1 = FREE |
E2 = FREE | E3 = FREE | E4 = FREE | -999999 <= KESWICK_MIN <= 999999 |
EXPORTACTUAL = FREE | INFLOW = FREE | EXPRATIO_ = FREE |

Maximize Solve Number 1
OBJ1: OBJ - 1OBJECTIVE = FREE
OBJ0: OBJ - 0OBJECTIVE = FREE
S1_1: 343809 0OBJECTIVE + 0S1ZONE1/1 - 0STORAGE1/1
S1_2: 6505.38 0OBJECTIVE + 0S1ZONE2/1 - 0STORAGE1/1
S1_3: 3252.69 0OBJECTIVE + 0S1ZONE3/1 - 0STORAGE1/1
S1_4: 1626.34 0OBJECTIVE + 0S1ZONE4/1 - 0STORAGE1/1
S1_5: 487.903 0OBJECTIVE + 0S1ZONE5/1 - 0STORAGE1/1
D30: 2550 0OBJECTIVE - 0CONTINUITY30/1 <= 1000
C30_MIF: 2560 0OBJECTIVE - 0C30TOTAL/1 + 0MEETC30MIN/1
D31: 2550 0OBJECTIVE - 0CONTINUITY31/1 <= 1000
S2_1: 301524 0OBJECTIVE + 0S2ZONE1/1 - 0STORAGE2/1
S2_2: 6342.74 0OBJECTIVE + 0S2ZONE2/1 - 0STORAGE2/1
S2_3: 3090.05 0OBJECTIVE + 0S2ZONE3/1 - 0STORAGE2/1
S2_4: 1463.71 0OBJECTIVE + 0S2ZONE4/1 - 0STORAGE2/1
S2_5: 162.634 0OBJECTIVE + 0S2ZONE5/1 - 0STORAGE2/1
D2: 2550 0OBJECTIVE - 0CONTINUITY2/1 <= 1000
C2_MIF: 2560 0OBJECTIVE - 0C2TOTAL/1 + 0C2MINFLOW/1
D33: 2550 0OBJECTIVE - 0CONTINUITY33/1 = 0
D34A: 2550 0OBJECTIVE - 0CONTINUITY34/1 <= 1000
C34A: 2550 0OBJECTIVE - 0CONTINUITY34/1 + 0SETMRDO/1 <= 210000
D34B: 2550 0OBJECTIVE - 0CONTINUITY34/1 <= 1000
S3_1: 41797 0OBJECTIVE + 0S3ZONE1/1 - 0STORAGE3/1
S3_2: 6668.01 0OBJECTIVE + 0S3ZONE2/1 - 0STORAGE3/1
S3_3: 6668.01 0OBJECTIVE + 0S3ZONE3/1 - 0STORAGE3/1
S3_4: 650.538 0OBJECTIVE + 0S3ZONE4/1 - 0STORAGE3/1
D3: 420 0OBJECTIVE + 0MAXRELEASE3/1 - 0CONTINUITY3/1 <= 1000
S4_1: 41797 0OBJECTIVE + 0S4ZONE1/1 - 0STORAGE4/1
S4_2: 6668.01 0OBJECTIVE + 0S4ZONE2/1 - 0STORAGE4/1
S4_3: 6668.01 0OBJECTIVE + 0S4ZONE3/1 - 0STORAGE4/1
S4_4: 325.269 0OBJECTIVE + 0S4ZONE4/1 - 0STORAGE4/1
D4: 420 0OBJECTIVE + 0MAXRELEASE4/1 - 0CONTINUITY4/1 <= 1000
S1_6:  - 53669.4 0OBJECTIVE + 0S1ZONE6/1 - 0STORAGE1/1
F1:  - 3400 0OBJECTIVE - 0CONTINUITY1/1
S2_6:  - 34153.2 0OBJECTIVE + 0S2ZONE6/1 - 0STORAGE2/1
F2:  - 3400 0OBJECTIVE - 0CONTINUITY2/1
C34B:  - 550 0OBJECTIVE - 0CONTINUITY34/1 <= 210000
S3_5:  - 10571.2 0OBJECTIVE + 0S3ZONE5/1 - 0STORAGE3/1
F3:  - 3400 0OBJECTIVE - 0CONTINUITY3/1
```

```
S4_5:  - 10571.2 0OBJECTIVE + 0S4ZONE5/1 - 0STORAGE4/1
F4:  - 3400 0OBJECTIVE - 0CONTINUITY4/1
C1: 0MAXRELEASE1/1 - 0CONTINUITY1/1 + 0CONTINUITY30/1 <= 50000
C2: 0C2TOTAL/1 + 0MAXRELEASE2/1 - 0CONTINUITY2/1 + 0CONTINUITY32/1 <= 80000
C2_EXC:  - 0C2TOTAL/1
C30: 0C30TOTAL/1 - 0CONTINUITY30/1 + 0CONTINUITY31/1 <= 80000
C30_EXC:  - 0C30TOTAL/1
C31:  - 0CONTINUITY31/1 + 0CONTINUITY32/1 <= 80000
C32:  - 0CONTINUITY32/1 + 0CONTINUITY33/1 <= 80000
C33:  - 0CONTINUITY33/1 + 0CONTINUITY34/1 - 0INFLOW_ALIAS/1 <= 80000
D34C:  - 0CONTINUITY34/1 + 0CONTINUITY3/1 - 0EXPORTACTUAL_ALIAS/1 <= 4600
D34D:  - 0CONTINUITY34/1 + 0CONTINUITY4/1 - 0EXPORTACTUAL_ALIAS/1 <= 6680
S1: 0STORAGE1/1 - 8.91348 0AREA1/1 - 16.2634 0CONTINUITY1/1
E1: 61.4876 0EVAP1/1 - 0CONTINUITY1/1 = FREE
A1: 0AREA1/1 - 0.220781 0EVAP1/1
S2: 0STORAGE2/1 - 14.1896 0AREA2/1 - 16.2634 0CONTINUITY2/1
E2: 61.4876 0EVAP2/1 - 0CONTINUITY2/1 = FREE
A2: 0AREA2/1 - 0.117924 0EVAP2/1
S3: 0STORAGE3/1 - 13.0406 0AREA3/1 - 16.2634 0CONTINUITY3/1
E3: 61.4876 0EVAP3/1 - 0CONTINUITY3/1 = FREE
A3: 0AREA3/1 - 0.301332 0EVAP3/1
S4: 0STORAGE4/1 - 13.0303 0AREA4/1 - 16.2634 0CONTINUITY4/1
E4: 61.4876 0EVAP4/1 - 0CONTINUITY4/1 = FREE
A4: 0AREA4/1 - 0.301332 0EVAP4/1
KESWICK_MIN: 0SETKESWICK_MIN/1 >= -999999 <= 999999
EXPORTACTUAL: 0EXPORTACTUAL_ALIAS/1 + 0EXPORT_COMPLY/1 = FREE
INFLOW: 0INFLOW_ALIAS/1 - 0.65 0FIND_MAX_EXPORT/1 = FREE
EXPRATIO_: 0EXPRATIO__ALIAS/1 = FREE
EIEXPCTRL: 0FIND_MAX_EXPORT/1 - 0EXPORT_COMPLY/1

1OBJECTIVE = 0 | 0OBJECTIVE = 0 | 0C2TOTAL/1 = 0 | 0C2MINFLOW/1 <= 1000 |
0C30TOTAL/1 = 0 | 0S1ZONE1/1 <= 550 | 0S1ZONE2/1 <= 1165 | 0S1ZONE3/1 <= 785
0S1ZONE4/1 <= 1100 | 0S1ZONE5/1 <= 400 | 0S1ZONE6/1 <= 552 | 0STORAGE1/1 = 0
0AREA1/1 = 2099.39 | 0EVAP1/1 = 1545.86 | 0MAXRELEASE1/1 <= 12702.7 |
0S2ZONE1/1 <= 29.6 | 0S2ZONE2/1 <= 822.4 | 0S2ZONE3/1 <= 1618 |
0S2ZONE4/1 <= 530 | 0S2ZONE5/1 <= 250 | 0S2ZONE6/1 <= 308 | 0STORAGE2/1 = 0
0AREA2/1 = 172.154 | 0EVAP2/1 = 70.4996 | 0MAXRELEASE2/1 <= 50000 |
0S3ZONE1/1 <= 45 | 0S3ZONE2/1 <= 0 | 0S3ZONE3/1 <= 455 | 0S3ZONE4/1 <= 450
0S3ZONE5/1 <= 22 | 0STORAGE3/1 = 0 | 0AREA3/1 = 1190.72 | 0EVAP3/1 = 535.631
0MAXRELEASE3/1 <= 14376 | 0S4ZONE1/1 <= 55 | 0S4ZONE2/1 <= 0 | 0S4ZONE3/1 <=
445
0S4ZONE4/1 <= 500 | 0S4ZONE5/1 <= 67 | 0STORAGE4/1 = 0 | 0AREA4/1 = 1316.73
0EVAP4/1 = 612.726 | 0MAXRELEASE4/1 <= 14376 | 0CONTINUITY1/1 = -9944.89 |
0CONTINUITY30/1 = 0 | 0CONTINUITY2/1 = -737.903 | 0CONTINUITY31/1 = 0 |
0CONTINUITY32/1 = 0 | 0CONTINUITY33/1 = 0 | 0CONTINUITY34/1 = 0 |
0CONTINUITY3/1 = -731.855 | 0CONTINUITY4/1 = -894.489 | 0SETMRDO/1 <= 1000
0MEETC30MIN/1 <= 3500 | 0SETKESWICK_MIN/1 = 3500 | 0EXPORTACTUAL_ALIAS/1 = 0
0INFLOW_ALIAS/1 = 0 | 0EXPRATIO__ALIAS/1 = 0.65 | 0FIND_MAX_EXPORT/1 = 0 |
0EXPORT_COMPLY/1 <= 0 |
```

Appendix B-4: Simplified Two River System Model, Run IVa

```
>> CALSIM Version 1.2.
This program is Copyright (C) 1998 State of California, all rights reserved
2001D10A
CLP options: MATLIST both
>> These extra XA options were obtained:
CLP options: MUTE NO LISTINPUT NO
>> Solving at date  1/31, of water year 1922
Maximize Solve Number 1
OBJ: OBJ1 + OBJ0
```

```
Constraints
1OBJECTIVE:  - OBJ1 = 0
0OBJECTIVE:  - OBJ0 + 343809 S1_1 + 6505.38 S1_2 + 3252.69 S1_3 + 1626.34 S1_4
   487.903 S1_5 + 2550 D30 + 2560 C30_MIF + 2550 D31 + 301524 S2_1
   6342.74 S2_2 + 3090.05 S2_3 + 1463.71 S2_4 + 162.634 S2_5 + 2550 D2
   2560 C2_MIF + 2550 D33 + 2550 D34A + 2550 C34A + 2550 D34B + 41797 S3_1
   6668.01 S3_2 + 6668.01 S3_3 + 650.538 S3_4 + 420 D3 + 41797 S4_1
   6668.01 S4_2 + 6668.01 S4_3 + 325.269 S4_4 + 420 D4 - 53669.4 S1_6 - 3400 F1
    - 34153.2 S2_6 - 3400 F2 - 550 C34B - 10571.2 S3_5 - 3400 F3 - 10571.2 S4_5
    - 3400 F4 - 2000 SLACK0126 - 2000 SLACK0127 = 0
0C2TOTAL/1:  - C2_MIF + C2 - C2_EXC = 0
0C2MINFLOW/1: C2_MIF <= 1000
0C30TOTAL/1:  - C30_MIF + C30 - C30_EXC = 0
0S1ZONE1/1: S1_1 <= 550
0S1ZONE2/1: S1_2 <= 1165
0S1ZONE3/1: S1_3 <= 785
0S1ZONE4/1: S1_4 <= 1100
0S1ZONE5/1: S1_5 <= 400
0S1ZONE6/1: S1_6 <= 552
0STORAGE1/1:  - S1_1 - S1_2 - S1_3 - S1_4 - S1_5 - S1_6 + S1 = 0
0AREA1/1:  - 8.91348 S1 + A1 = 2099.39
0EVAP1/1: 61.4876 E1 - 0.220781 A1 = 1545.86
0MAXRELEASE1/1: C1 <= 12702.7
0S2ZONE1/1: S2_1 <= 29.6
0S2ZONE2/1: S2_2 <= 822.4
0S2ZONE3/1: S2_3 <= 1618
0S2ZONE4/1: S2_4 <= 530
0S2ZONE5/1: S2_5 <= 250
0S2ZONE6/1: S2_6 <= 308
0STORAGE2/1:  - S2_1 - S2_2 - S2_3 - S2_4 - S2_5 - S2_6 + S2 = 0
0AREA2/1:  - 14.1896 S2 + A2 = 172.154
0EVAP2/1: 61.4876 E2 - 0.117924 A2 = 70.4996
0MAXRELEASE2/1: C2 <= 50000
0S3ZONE1/1: S3_1 <= 45
0S3ZONE2/1: S3_2 <= 0
0S3ZONE3/1: S3_3 <= 455
0S3ZONE4/1: S3_4 <= 450
0S3ZONE5/1: S3_5 <= 22
0STORAGE3/1:  - S3_1 - S3_2 - S3_3 - S3_4 - S3_5 + S3 = 0
0AREA3/1:  - 13.0406 S3 + A3 = 1190.72
0EVAP3/1: 61.4876 E3 - 0.301332 A3 = 535.631
0MAXRELEASE3/1: D3 <= 14376
0S4ZONE1/1: S4_1 <= 55
0S4ZONE2/1: S4_2 <= 0
0S4ZONE3/1: S4_3 <= 445
0S4ZONE4/1: S4_4 <= 500
0S4ZONE5/1: S4_5 <= 67
0STORAGE4/1:  - S4_1 - S4_2 - S4_3 - S4_4 - S4_5 + S4 = 0
0AREA4/1:  - 13.0303 S4 + A4 = 1316.73
0EVAP4/1: 61.4876 E4 - 0.301332 A4 = 612.726
0MAXRELEASE4/1: D4 <= 14376
0CONTINUITY1/1:  - F1 - C1 - 16.2634 S1 - E1 = -9944.89
0CONTINUITY30/1:  - D30 + C1 - C30 = 0
0CONTINUITY2/1:  - D2 - F2 - C2 - 16.2634 S2 - E2 = -737.903
0CONTINUITY31/1:  - D31 + C30 - C31 = 0
0CONTINUITY32/1: C2 + C31 - C32 = 0
0CONTINUITY33/1:  - D33 + C32 - C33 = 0
0CONTINUITY34/1:  - D34A - C34A - D34B - C34B + C33 - D34C - D34D = 0
0CONTINUITY3/1:  - D3 - F3 + D34C - 16.2634 S3 - E3 = -731.855
0CONTINUITY4/1:  - D4 - F4 + D34D - 16.2634 S4 - E4 = -894.489
0SETMRDO/1: C34A <= 1000
0MEETC30MIN/1: C30_MIF <= 3500
0SETKESWICK_MIN/1: KESWICK_MIN = 3500
```

```
0EXPORTACTUAL_ALIAS/1:  - D34C - D34D + EXPORTACTUAL = 0
0INFLOW_ALIAS/1:  - C33 + INFLOW = 0
0EXPRATIO__ALIAS/1: EXPRATIO_ = 0.65
0FIND_MAX_EXPORT/1:  - 0.65 INFLOW + EIEXPCTRL = 0
0EXPORT_COMPLY/1: EXPORTACTUAL - EIEXPCTRL <= 0
0MAXLIMITCVP/1: D34C <= 4600
0MAXLIMITSWP/1: D34D <= 6680
0MINLIMITCVP/1: D34C - SURPL0126 + SLACK0126 = 800
0MINLIMITSWP/1: D34D - SURPL0127 + SLACK0127 = 300
0SET_TOTAL/1:  - D34C - D34D + TOTALPUMPING = 0

OBJ1 = FREE | OBJ0 = FREE | D30 <= 1000 | D31 <= 1000 | D2 <= 1000 | D33 = 0
D34A <= 1000 | C34A <= 210000 | D34B <= 1000 | D3 <= 1000 | D4 <= 1000 |
C34B <= 210000 | C1 <= 50000 | C2 <= 80000 | C30 <= 80000 | C31 <= 80000 |
C32 <= 80000 | C33 <= 80000 | D34C <= 4600 | D34D <= 6680 | E1 = FREE |
E2 = FREE | E3 = FREE | E4 = FREE | -999999 <= KESWICK_MIN <= 999999 |
EXPORTACTUAL = FREE | INFLOW = FREE | EXPRATIO_ = FREE |

Maximize Solve Number 1
OBJ1: OBJ - 1OBJECTIVE = FREE
OBJ0: OBJ - 0OBJECTIVE = FREE
S1_1: 343809 0OBJECTIVE + 0S1ZONE1/1 - 0STORAGE1/1
S1_2: 6505.38 0OBJECTIVE + 0S1ZONE2/1 - 0STORAGE1/1
S1_3: 3252.69 0OBJECTIVE + 0S1ZONE3/1 - 0STORAGE1/1
S1_4: 1626.34 0OBJECTIVE + 0S1ZONE4/1 - 0STORAGE1/1
S1_5: 487.903 0OBJECTIVE + 0S1ZONE5/1 - 0STORAGE1/1
D30: 2550 0OBJECTIVE - 0CONTINUITY30/1 <= 1000
C30_MIF: 2560 0OBJECTIVE - 0C30TOTAL/1 + 0MEETC30MIN/1
D31: 2550 0OBJECTIVE - 0CONTINUITY31/1 <= 1000
S2_1: 301524 0OBJECTIVE + 0S2ZONE1/1 - 0STORAGE2/1
S2_2: 6342.74 0OBJECTIVE + 0S2ZONE2/1 - 0STORAGE2/1
S2_3: 3090.05 0OBJECTIVE + 0S2ZONE3/1 - 0STORAGE2/1
S2_4: 1463.71 0OBJECTIVE + 0S2ZONE4/1 - 0STORAGE2/1
S2_5: 162.634 0OBJECTIVE + 0S2ZONE5/1 - 0STORAGE2/1
D2: 2550 0OBJECTIVE - 0CONTINUITY2/1 <= 1000
C2_MIF: 2560 0OBJECTIVE - 0C2TOTAL/1 + 0C2MINFLOW/1
D33: 2550 0OBJECTIVE - 0CONTINUITY33/1 = 0
D34A: 2550 0OBJECTIVE - 0CONTINUITY34/1 <= 1000
C34A: 2550 0OBJECTIVE - 0CONTINUITY34/1 + 0SETMRDO/1 <= 210000
D34B: 2550 0OBJECTIVE - 0CONTINUITY34/1 <= 1000
S3_1: 41797 0OBJECTIVE + 0S3ZONE1/1 - 0STORAGE3/1
S3_2: 6668.01 0OBJECTIVE + 0S3ZONE2/1 - 0STORAGE3/1
S3_3: 6668.01 0OBJECTIVE + 0S3ZONE3/1 - 0STORAGE3/1
S3_4: 650.538 0OBJECTIVE + 0S3ZONE4/1 - 0STORAGE3/1
D3: 420 0OBJECTIVE + 0MAXRELEASE3/1 - 0CONTINUITY3/1 <= 1000
S4_1: 41797 0OBJECTIVE + 0S4ZONE1/1 - 0STORAGE4/1
S4_2: 6668.01 0OBJECTIVE + 0S4ZONE2/1 - 0STORAGE4/1
S4_3: 6668.01 0OBJECTIVE + 0S4ZONE3/1 - 0STORAGE4/1
S4_4: 325.269 0OBJECTIVE + 0S4ZONE4/1 - 0STORAGE4/1
D4: 420 0OBJECTIVE + 0MAXRELEASE4/1 - 0CONTINUITY4/1 <= 1000
S1_6:  - 53669.4 0OBJECTIVE + 0S1ZONE6/1 - 0STORAGE1/1
F1:  - 3400 0OBJECTIVE - 0CONTINUITY1/1
S2_6:  - 34153.2 0OBJECTIVE + 0S2ZONE6/1 - 0STORAGE2/1
F2:  - 3400 0OBJECTIVE - 0CONTINUITY2/1
C34B:  - 550 0OBJECTIVE - 0CONTINUITY34/1 <= 210000
S3_5:  - 10571.2 0OBJECTIVE + 0S3ZONE5/1 - 0STORAGE3/1
F3:  - 3400 0OBJECTIVE - 0CONTINUITY3/1
S4_5:  - 10571.2 0OBJECTIVE + 0S4ZONE5/1 - 0STORAGE4/1
F4:  - 3400 0OBJECTIVE - 0CONTINUITY4/1
C1: 0MAXRELEASE1/1 - 0CONTINUITY1/1 + 0CONTINUITY30/1 <= 50000
C2: 0C2TOTAL/1 + 0MAXRELEASE2/1 - 0CONTINUITY2/1 + 0CONTINUITY32/1 <= 80000
C2_EXC:  - 0C2TOTAL/1
C30: 0C30TOTAL/1 - 0CONTINUITY30/1 + 0CONTINUITY31/1 <= 80000
```

```
C30_EXC:  - 0C30TOTAL/1
C31:  - 0CONTINUITY31/1 + 0CONTINUITY32/1 <= 80000
C32:  - 0CONTINUITY32/1 + 0CONTINUITY33/1 <= 80000
C33:  - 0CONTINUITY33/1 + 0CONTINUITY34/1 - 0INFLOW_ALIAS/1 <= 80000
D34C:  - 0CONTINUITY34/1 + 0CONTINUITY3/1 - 0EXPORTACTUAL_ALIAS/1
   0MAXLIMITCVP/1 + 0MINLIMITCVP/1 - 0SET_TOTAL/1 <= 4600
D34D:  - 0CONTINUITY34/1 + 0CONTINUITY4/1 - 0EXPORTACTUAL_ALIAS/1
   0MAXLIMITSWP/1 + 0MINLIMITSWP/1 - 0SET_TOTAL/1 <= 6680
S1: 0STORAGE1/1 - 8.91348 0AREA1/1 - 16.2634 0CONTINUITY1/1
E1: 61.4876 0EVAP1/1 - 0CONTINUITY1/1 = FREE
A1: 0AREA1/1 - 0.220781 0EVAP1/1
S2: 0STORAGE2/1 - 14.1896 0AREA2/1 - 16.2634 0CONTINUITY2/1
E2: 61.4876 0EVAP2/1 - 0CONTINUITY2/1 = FREE
A2: 0AREA2/1 - 0.117924 0EVAP2/1
S3: 0STORAGE3/1 - 13.0406 0AREA3/1 - 16.2634 0CONTINUITY3/1
E3: 61.4876 0EVAP3/1 - 0CONTINUITY3/1 = FREE
A3: 0AREA3/1 - 0.301332 0EVAP3/1
S4: 0STORAGE4/1 - 13.0303 0AREA4/1 - 16.2634 0CONTINUITY4/1
E4: 61.4876 0EVAP4/1 - 0CONTINUITY4/1 = FREE
A4: 0AREA4/1 - 0.301332 0EVAP4/1
KESWICK_MIN: 0SETKESWICK_MIN/1 >= -999999 <= 999999
EXPORTACTUAL: 0EXPORTACTUAL_ALIAS/1 + 0EXPORT_COMPLY/1 = FREE
INFLOW: 0INFLOW_ALIAS/1 - 0.65 0FIND_MAX_EXPORT/1 = FREE
EXPRATIO_: 0EXPRATIO__ALIAS/1 = FREE
EIEXPCTRL: 0FIND_MAX_EXPORT/1 - 0EXPORT_COMPLY/1
SURPL0126:  - 0MINLIMITCVP/1
SLACK0126:  - 2000 0OBJECTIVE + 0MINLIMITCVP/1
SURPL0127:  - 0MINLIMITSWP/1
SLACK0127:  - 2000 0OBJECTIVE + 0MINLIMITSWP/1
TOTALPUMPING: 0SET_TOTAL/1

1OBJECTIVE = 0 | 0OBJECTIVE = 0 | 0C2TOTAL/1 = 0 | 0C2MINFLOW/1 <= 1000 |
0C30TOTAL/1 = 0 | 0S1ZONE1/1 <= 550 | 0S1ZONE2/1 <= 1165 | 0S1ZONE3/1 <= 785
0S1ZONE4/1 <= 1100 | 0S1ZONE5/1 <= 400 | 0S1ZONE6/1 <= 552 | 0STORAGE1/1 = 0
0AREA1/1 = 2099.39 | 0EVAP1/1 = 1545.86 | 0MAXRELEASE1/1 <= 12702.7 |
0S2ZONE1/1 <= 29.6 | 0S2ZONE2/1 <= 822.4 | 0S2ZONE3/1 <= 1618 |
0S2ZONE4/1 <= 530 | 0S2ZONE5/1 <= 250 | 0S2ZONE6/1 <= 308 | 0STORAGE2/1 = 0
0AREA2/1 = 172.154 | 0EVAP2/1 = 70.4996 | 0MAXRELEASE2/1 <= 50000 |
0S3ZONE1/1 <= 45 | 0S3ZONE2/1 <= 0 | 0S3ZONE3/1 <= 455 | 0S3ZONE4/1 <= 450
0S3ZONE5/1 <= 22 | 0STORAGE3/1 = 0 | 0AREA3/1 = 1190.72 | 0EVAP3/1 = 535.631
0MAXRELEASE3/1 <= 14376 | 0S4ZONE1/1 <= 55 | 0S4ZONE2/1 <= 0 | 0S4ZONE3/1 <=
445
0S4ZONE4/1 <= 500 | 0S4ZONE5/1 <= 67 | 0STORAGE4/1 = 0 | 0AREA4/1 = 1316.73
0EVAP4/1 = 612.726 | 0MAXRELEASE4/1 <= 14376 | 0CONTINUITY1/1 = -9944.89 |
0CONTINUITY30/1 = 0 | 0CONTINUITY2/1 = -737.903 | 0CONTINUITY31/1 = 0 |
0CONTINUITY32/1 = 0 | 0CONTINUITY33/1 = 0 | 0CONTINUITY34/1 = 0 |
0CONTINUITY3/1 = -731.855 | 0CONTINUITY4/1 = -894.489 | 0SETMRDO/1 <= 1000
0MEETC30MIN/1 <= 3500 | 0SETKESWICK_MIN/1 = 3500 | 0EXPORTACTUAL_ALIAS/1 = 0
0INFLOW_ALIAS/1 = 0 | 0EXPRATIO__ALIAS/1 = 0.65 | 0FIND_MAX_EXPORT/1 = 0 |
0EXPORT_COMPLY/1 <= 0 | 0MAXLIMITCVP/1 <= 4600 | 0MAXLIMITSWP/1 <= 6680 |
0MINLIMITCVP/1 = 800 | 0MINLIMITSWP/1 = 300 | 0SET_TOTAL/1 = 0 |
```

Appendix B-5: Simplified Two River System Model, Run IVb

```
>> CALSIM Version 1.2.
This program is Copyright (C) 1998 State of California, all rights reserved
2001D10A
CLP options: MATLIST both
>> These extra XA options were obtained:
CLP options: MUTE NO LISTINPUT NO
>> Solving at date  1/31, of water year 1922
Maximize Solve Number 1
```

```
OBJ: OBJ1 + OBJ0

Constraints
1OBJECTIVE:  - OBJ1 = 0
0OBJECTIVE:  - OBJ0 + 343809 S1_1 + 6505.38 S1_2 + 3252.69 S1_3 + 1626.34 S1_4
   487.903 S1_5 + 2550 D30 + 2560 C30_MIF + 2550 D31 + 301524 S2_1
   6342.74 S2_2 + 3090.05 S2_3 + 1463.71 S2_4 + 162.634 S2_5 + 2550 D2
   2560 C2_MIF + 2550 D33 + 2550 D34A + 2550 C34A + 2550 D34B + 41797 S3_1
   6668.01 S3_2 + 6668.01 S3_3 + 650.538 S3_4 + 420 D3 + 41797 S4_1
   6668.01 S4_2 + 6668.01 S4_3 + 325.269 S4_4 + 420 D4 - 53669.4 S1_6 - 3400 F1
    - 34153.2 S2_6 - 3400 F2 - 550 C34B - 10571.2 S3_5 - 3400 F3 - 10571.2 S4_5
    - 3400 F4 - 2e+006 SLACK0126 - 2e+006 SLACK0127 = 0
0C2TOTAL/1:  - C2_MIF + C2 - C2_EXC = 0
0C2MINFLOW/1: C2_MIF <= 1000
0C30TOTAL/1:  - C30_MIF + C30 - C30_EXC = 0
0S1ZONE1/1: S1_1 <= 550
0S1ZONE2/1: S1_2 <= 1165
0S1ZONE3/1: S1_3 <= 785
0S1ZONE4/1: S1_4 <= 1100
0S1ZONE5/1: S1_5 <= 400
0S1ZONE6/1: S1_6 <= 552
0STORAGE1/1:  - S1_1 - S1_2 - S1_3 - S1_4 - S1_5 - S1_6 + S1 = 0
0AREA1/1:  - 8.91348 S1 + A1 = 2099.39
0EVAP1/1: 61.4876 E1 - 0.220781 A1 = 1545.86
0MAXRELEASE1/1: C1 <= 12702.7
0S2ZONE1/1: S2_1 <= 29.6
0S2ZONE2/1: S2_2 <= 822.4
0S2ZONE3/1: S2_3 <= 1618
0S2ZONE4/1: S2_4 <= 530
0S2ZONE5/1: S2_5 <= 250
0S2ZONE6/1: S2_6 <= 308
0STORAGE2/1:  - S2_1 - S2_2 - S2_3 - S2_4 - S2_5 - S2_6 + S2 = 0
0AREA2/1:  - 14.1896 S2 + A2 = 172.154
0EVAP2/1: 61.4876 E2 - 0.117924 A2 = 70.4996
0MAXRELEASE2/1: C2 <= 50000
0S3ZONE1/1: S3_1 <= 45
0S3ZONE2/1: S3_2 <= 0
0S3ZONE3/1: S3_3 <= 455
0S3ZONE4/1: S3_4 <= 450
0S3ZONE5/1: S3_5 <= 22
0STORAGE3/1:  - S3_1 - S3_2 - S3_3 - S3_4 - S3_5 + S3 = 0
0AREA3/1:  - 13.0406 S3 + A3 = 1190.72
0EVAP3/1: 61.4876 E3 - 0.301332 A3 = 535.631
0MAXRELEASE3/1: D3 <= 14376
0S4ZONE1/1: S4_1 <= 55
0S4ZONE2/1: S4_2 <= 0
0S4ZONE3/1: S4_3 <= 445
0S4ZONE4/1: S4_4 <= 500
0S4ZONE5/1: S4_5 <= 67
0STORAGE4/1:  - S4_1 - S4_2 - S4_3 - S4_4 - S4_5 + S4 = 0
0AREA4/1:  - 13.0303 S4 + A4 = 1316.73
0EVAP4/1: 61.4876 E4 - 0.301332 A4 = 612.726
0MAXRELEASE4/1: D4 <= 14376
0CONTINUITY1/1:  - F1 - C1 - 16.2634 S1 - E1 = -9944.89
0CONTINUITY30/1:  - D30 + C1 - C30 = 0
0CONTINUITY2/1:  - D2 - F2 - C2 - 16.2634 S2 - E2 = -737.903
0CONTINUITY31/1:  - D31 + C30 - C31 = 0
0CONTINUITY32/1: C2 + C31 - C32 = 0
0CONTINUITY33/1:  - D33 + C32 - C33 = 0
0CONTINUITY34/1:  - D34A - C34A - D34B - C34B + C33 - D34C - D34D = 0
0CONTINUITY3/1:  - D3 - F3 + D34C - 16.2634 S3 - E3 = -731.855
0CONTINUITY4/1:  - D4 - F4 + D34D - 16.2634 S4 - E4 = -894.489
0SETMRDO/1: C34A <= 1000
```

```
0MEETC30MIN/1: C30_MIF <= 3500
0SETKESWICK_MIN/1: KESWICK_MIN = 3500
0EXPORTACTUAL_ALIAS/1:  - D34C - D34D + EXPORTACTUAL = 0
0INFLOW_ALIAS/1:  - C33 + INFLOW = 0
0EXPRATIO__ALIAS/1: EXPRATIO_ = 0.65
0FIND_MAX_EXPORT/1:  - 0.65 INFLOW + EIEXPCTRL = 0
0EXPORT_COMPLY/1: EXPORTACTUAL - EIEXPCTRL <= 0
0MAXLIMITCVP/1: D34C <= 4600
0MAXLIMITSWP/1: D34D <= 6680
0MINLIMITCVP/1: D34C - SURPL0126 + SLACK0126 = 800
0MINLIMITSWP/1: D34D - SURPL0127 + SLACK0127 = 300
0SET_TOTAL/1:  - D34C - D34D + TOTALPUMPING = 0

OBJ1 = FREE | OBJ0 = FREE | D30 <= 1000 | D31 <= 1000 | D2 <= 1000 | D33 = 0
D34A <= 1000 | C34A <= 210000 | D34B <= 1000 | D3 <= 1000 | D4 <= 1000 |
C34B <= 210000 | C1 <= 50000 | C2 <= 80000 | C30 <= 80000 | C31 <= 80000 |
C32 <= 80000 | C33 <= 80000 | D34C <= 4600 | D34D <= 6680 | E1 = FREE |
E2 = FREE | E3 = FREE | E4 = FREE | -999999 <= KESWICK_MIN <= 999999 |
EXPORTACTUAL = FREE | INFLOW = FREE | EXPRATIO_ = FREE |

Maximize Solve Number 1
OBJ1: OBJ - 1OBJECTIVE = FREE
OBJ0: OBJ - 0OBJECTIVE = FREE
S1_1: 343809 0OBJECTIVE + 0S1ZONE1/1 - 0STORAGE1/1
S1_2: 6505.38 0OBJECTIVE + 0S1ZONE2/1 - 0STORAGE1/1
S1_3: 3252.69 0OBJECTIVE + 0S1ZONE3/1 - 0STORAGE1/1
S1_4: 1626.34 0OBJECTIVE + 0S1ZONE4/1 - 0STORAGE1/1
S1_5: 487.903 0OBJECTIVE + 0S1ZONE5/1 - 0STORAGE1/1
D30: 2550 0OBJECTIVE - 0CONTINUITY30/1 <= 1000
C30_MIF: 2560 0OBJECTIVE - 0C30TOTAL/1 + 0MEETC30MIN/1
D31: 2550 0OBJECTIVE - 0CONTINUITY31/1 <= 1000
S2_1: 301524 0OBJECTIVE + 0S2ZONE1/1 - 0STORAGE2/1
S2_2: 6342.74 0OBJECTIVE + 0S2ZONE2/1 - 0STORAGE2/1
S2_3: 3090.05 0OBJECTIVE + 0S2ZONE3/1 - 0STORAGE2/1
S2_4: 1463.71 0OBJECTIVE + 0S2ZONE4/1 - 0STORAGE2/1
S2_5: 162.634 0OBJECTIVE + 0S2ZONE5/1 - 0STORAGE2/1
D2: 2550 0OBJECTIVE - 0CONTINUITY2/1 <= 1000
C2_MIF: 2560 0OBJECTIVE - 0C2TOTAL/1 + 0C2MINFLOW/1
D33: 2550 0OBJECTIVE - 0CONTINUITY33/1 = 0
D34A: 2550 0OBJECTIVE - 0CONTINUITY34/1 <= 1000
C34A: 2550 0OBJECTIVE - 0CONTINUITY34/1 + 0SETMRDO/1 <= 210000
D34B: 2550 0OBJECTIVE - 0CONTINUITY34/1 <= 1000
S3_1: 41797 0OBJECTIVE + 0S3ZONE1/1 - 0STORAGE3/1
S3_2: 6668.01 0OBJECTIVE + 0S3ZONE2/1 - 0STORAGE3/1
S3_3: 6668.01 0OBJECTIVE + 0S3ZONE3/1 - 0STORAGE3/1
S3_4: 650.538 0OBJECTIVE + 0S3ZONE4/1 - 0STORAGE3/1
D3: 420 0OBJECTIVE + 0MAXRELEASE3/1 - 0CONTINUITY3/1 <= 1000
S4_1: 41797 0OBJECTIVE + 0S4ZONE1/1 - 0STORAGE4/1
S4_2: 6668.01 0OBJECTIVE + 0S4ZONE2/1 - 0STORAGE4/1
S4_3: 6668.01 0OBJECTIVE + 0S4ZONE3/1 - 0STORAGE4/1
S4_4: 325.269 0OBJECTIVE + 0S4ZONE4/1 - 0STORAGE4/1
D4: 420 0OBJECTIVE + 0MAXRELEASE4/1 - 0CONTINUITY4/1 <= 1000
S1_6:  - 53669.4 0OBJECTIVE + 0S1ZONE6/1 - 0STORAGE1/1
F1:  - 3400 0OBJECTIVE - 0CONTINUITY1/1
S2_6:  - 34153.2 0OBJECTIVE + 0S2ZONE6/1 - 0STORAGE2/1
F2:  - 3400 0OBJECTIVE - 0CONTINUITY2/1
C34B:  - 550 0OBJECTIVE - 0CONTINUITY34/1 <= 210000
S3_5:  - 10571.2 0OBJECTIVE + 0S3ZONE5/1 - 0STORAGE3/1
F3:  - 3400 0OBJECTIVE - 0CONTINUITY3/1
S4_5:  - 10571.2 0OBJECTIVE + 0S4ZONE5/1 - 0STORAGE4/1
F4:  - 3400 0OBJECTIVE - 0CONTINUITY4/1
C1: 0MAXRELEASE1/1 - 0CONTINUITY1/1 + 0CONTINUITY30/1 <= 50000
C2: 0C2TOTAL/1 + 0MAXRELEASE2/1 - 0CONTINUITY2/1 + 0CONTINUITY32/1 <= 80000
```

```
C2_EXC:  - 0C2TOTAL/1
C30: 0C30TOTAL/1 - 0CONTINUITY30/1 + 0CONTINUITY31/1 <= 80000
C30_EXC:  - 0C30TOTAL/1
C31:  - 0CONTINUITY31/1 + 0CONTINUITY32/1 <= 80000
C32:  - 0CONTINUITY32/1 + 0CONTINUITY33/1 <= 80000
C33:  - 0CONTINUITY33/1 + 0CONTINUITY34/1 - 0INFLOW_ALIAS/1 <= 80000
D34C:  - 0CONTINUITY34/1 + 0CONTINUITY3/1 - 0EXPORTACTUAL_ALIAS/1
   0MAXLIMITCVP/1 + 0MINLIMITCVP/1 - 0SET_TOTAL/1 <= 4600
D34D:  - 0CONTINUITY34/1 + 0CONTINUITY4/1 - 0EXPORTACTUAL_ALIAS/1
   0MAXLIMITSWP/1 + 0MINLIMITSWP/1 - 0SET_TOTAL/1 <= 6680
S1: 0STORAGE1/1 - 8.91348 0AREA1/1 - 16.2634 0CONTINUITY1/1
E1: 61.4876 0EVAP1/1 - 0CONTINUITY1/1 = FREE
A1: 0AREA1/1 - 0.220781 0EVAP1/1
S2: 0STORAGE2/1 - 14.1896 0AREA2/1 - 16.2634 0CONTINUITY2/1
E2: 61.4876 0EVAP2/1 - 0CONTINUITY2/1 = FREE
A2: 0AREA2/1 - 0.117924 0EVAP2/1
S3: 0STORAGE3/1 - 13.0406 0AREA3/1 - 16.2634 0CONTINUITY3/1
E3: 61.4876 0EVAP3/1 - 0CONTINUITY3/1 = FREE
A3: 0AREA3/1 - 0.301332 0EVAP3/1
S4: 0STORAGE4/1 - 13.0303 0AREA4/1 - 16.2634 0CONTINUITY4/1
E4: 61.4876 0EVAP4/1 - 0CONTINUITY4/1 = FREE
A4: 0AREA4/1 - 0.301332 0EVAP4/1
KESWICK_MIN: 0SETKESWICK_MIN/1 >= -999999 <= 999999
EXPORTACTUAL: 0EXPORTACTUAL_ALIAS/1 + 0EXPORT_COMPLY/1 = FREE
INFLOW: 0INFLOW_ALIAS/1 - 0.65 0FIND_MAX_EXPORT/1 = FREE
EXPRATIO_: 0EXPRATIO__ALIAS/1 = FREE
EIEXPCTRL: 0FIND_MAX_EXPORT/1 - 0EXPORT_COMPLY/1
SURPL0126:  - 0MINLIMITCVP/1
SLACK0126:  - 2e+006 0OBJECTIVE + 0MINLIMITCVP/1
SURPL0127:  - 0MINLIMITSWP/1
SLACK0127:  - 2e+006 0OBJECTIVE + 0MINLIMITSWP/1
TOTALPUMPING: 0SET_TOTAL/1

1OBJECTIVE = 0 | 0OBJECTIVE = 0 | 0C2TOTAL/1 = 0 | 0C2MINFLOW/1 <= 1000 |
0C30TOTAL/1 = 0 | 0S1ZONE1/1 <= 550 | 0S1ZONE2/1 <= 1165 | 0S1ZONE3/1 <= 785
0S1ZONE4/1 <= 1100 | 0S1ZONE5/1 <= 400 | 0S1ZONE6/1 <= 552 | 0STORAGE1/1 = 0
0AREA1/1 = 2099.39 | 0EVAP1/1 = 1545.86 | 0MAXRELEASE1/1 <= 12702.7 |
0S2ZONE1/1 <= 29.6 | 0S2ZONE2/1 <= 822.4 | 0S2ZONE3/1 <= 1618 |
0S2ZONE4/1 <= 530 | 0S2ZONE5/1 <= 250 | 0S2ZONE6/1 <= 308 | 0STORAGE2/1 = 0
0AREA2/1 = 172.154 | 0EVAP2/1 = 70.4996 | 0MAXRELEASE2/1 <= 50000 |
0S3ZONE1/1 <= 45 | 0S3ZONE2/1 <= 0 | 0S3ZONE3/1 <= 455 | 0S3ZONE4/1 <= 450
0S3ZONE5/1 <= 22 | 0STORAGE3/1 = 0 | 0AREA3/1 = 1190.72 | 0EVAP3/1 = 535.631
0MAXRELEASE3/1 <= 14376 | 0S4ZONE1/1 <= 55 | 0S4ZONE2/1 <= 0 | 0S4ZONE3/1 <=
445
0S4ZONE4/1 <= 500 | 0S4ZONE5/1 <= 67 | 0STORAGE4/1 = 0 | 0AREA4/1 = 1316.73
0EVAP4/1 = 612.726 | 0MAXRELEASE4/1 <= 14376 | 0CONTINUITY1/1 = -9944.89 |
0CONTINUITY30/1 = 0 | 0CONTINUITY2/1 = -737.903 | 0CONTINUITY31/1 = 0 |
0CONTINUITY32/1 = 0 | 0CONTINUITY33/1 = 0 | 0CONTINUITY34/1 = 0 |
0CONTINUITY3/1 = -731.855 | 0CONTINUITY4/1 = -894.489 | 0SETMRDO/1 <= 1000
0MEETC30MIN/1 <= 3500 | 0SETKESWICK_MIN/1 = 3500 | 0EXPORTACTUAL_ALIAS/1 = 0
0INFLOW_ALIAS/1 = 0 | 0EXPRATIO__ALIAS/1 = 0.65 | 0FIND_MAX_EXPORT/1 = 0 |
0EXPORT_COMPLY/1 <= 0 | 0MAXLIMITCVP/1 <= 4600 | 0MAXLIMITSWP/1 <= 6680 |
0MINLIMITCVP/1 = 800 | 0MINLIMITSWP/1 = 300 | 0SET_TOTAL/1 = 0 |
```

Appendix B-6: Simplified Two River System Model, Run V

```
>> CALSIM Version 1.2.
This program is Copyright (C) 1998 State of California, all rights reserved
2001D10A
CLP options: MATLIST both
>> These extra XA options were obtained:
CLP options: MUTE NO LISTINPUT NO
```

```
>> Solving at date  1/31, of water year 1922
Maximize Solve Number 1
OBJ: OBJ1 + OBJ0

Constraints
1OBJECTIVE:  - OBJ1 = 0
0OBJECTIVE:  - OBJ0 + 343809 S1_1 + 6505.38 S1_2 + 3252.69 S1_3 + 1626.34 S1_4
   487.903 S1_5 + 2550 D30 + 2560 C30_MIF + 2550 D31 + 301524 S2_1
   6342.74 S2_2 + 3090.05 S2_3 + 1463.71 S2_4 + 162.634 S2_5 + 2550 D2
   2560 C2_MIF + 2550 D33 + 2550 D34A + 2550 C34A + 2550 D34B + 41797 S3_1
   6668.01 S3_2 + 6668.01 S3_3 + 650.538 S3_4 + 420 D3 + 41797 S4_1
   6668.01 S4_2 + 6668.01 S4_3 + 325.269 S4_4 + 420 D4 - 53669.4 S1_6 - 3400 F1
    - 34153.2 S2_6 - 3400 F2 - 550 C34B_CVP - 550 C34B_SWP - 450 UNUSED_FS
    - 450 UNUSED_SS - 10571.2 S3_5 - 3400 F3 - 10571.2 S4_5 - 3400 F4
    - 2000 SLACK0126 - 2000 SLACK0127 - 100 SURPL0159 - 100 SURPL0160 = 0
0C2TOTAL/1:  - C2_MIF + C2 - C2_EXC = 0
0C2MINFLOW/1: C2_MIF <= 1000
0C30TOTAL/1:  - C30_MIF + C30 - C30_EXC = 0
0S1ZONE1/1: S1_1 <= 550
0S1ZONE2/1: S1_2 <= 1165
0S1ZONE3/1: S1_3 <= 785
0S1ZONE4/1: S1_4 <= 1100
0S1ZONE5/1: S1_5 <= 400
0S1ZONE6/1: S1_6 <= 552
0STORAGE1/1:  - S1_1 - S1_2 - S1_3 - S1_4 - S1_5 - S1_6 + S1 = 0
0AREA1/1:  - 8.91348 S1 + A1 = 2099.39
0EVAP1/1: 61.4876 E1 - 0.220781 A1 = 1545.86
0MAXRELEASE1/1: C1 <= 12702.7
0S2ZONE1/1: S2_1 <= 29.6
0S2ZONE2/1: S2_2 <= 822.4
0S2ZONE3/1: S2_3 <= 1618
0S2ZONE4/1: S2_4 <= 530
0S2ZONE5/1: S2_5 <= 250
0S2ZONE6/1: S2_6 <= 308
0STORAGE2/1:  - S2_1 - S2_2 - S2_3 - S2_4 - S2_5 - S2_6 + S2 = 0
0AREA2/1:  - 14.1896 S2 + A2 = 172.154
0EVAP2/1: 61.4876 E2 - 0.117924 A2 = 70.4996
0MAXRELEASE2/1: C2 <= 50000
0S3ZONE1/1: S3_1 <= 45
0S3ZONE2/1: S3_2 <= 0
0S3ZONE3/1: S3_3 <= 455
0S3ZONE4/1: S3_4 <= 450
0S3ZONE5/1: S3_5 <= 22
0STORAGE3/1:  - S3_1 - S3_2 - S3_3 - S3_4 - S3_5 + S3 = 0
0AREA3/1:  - 13.0406 S3 + A3 = 1190.72
0EVAP3/1: 61.4876 E3 - 0.301332 A3 = 535.631
0MAXRELEASE3/1: D3 <= 14376
0S4ZONE1/1: S4_1 <= 55
0S4ZONE2/1: S4_2 <= 0
0S4ZONE3/1: S4_3 <= 445
0S4ZONE4/1: S4_4 <= 500
0S4ZONE5/1: S4_5 <= 67
0STORAGE4/1:  - S4_1 - S4_2 - S4_3 - S4_4 - S4_5 + S4 = 0
0AREA4/1:  - 13.0303 S4 + A4 = 1316.73
0EVAP4/1: 61.4876 E4 - 0.301332 A4 = 612.726
0MAXRELEASE4/1: D4 <= 14376
0CONTINUITY1/1:  - F1 - C1 - 16.2634 S1 - E1 = -9944.89
0CONTINUITY30/1:  - D30 + C1 - C30 = 0
0CONTINUITY2/1:  - D2 - F2 - C2 - 16.2634 S2 - E2 = -737.903
0CONTINUITY31/1:  - D31 + C30 - C31 = 0
0CONTINUITY32/1: C2 + C31 - C32 = 0
0CONTINUITY33/1:  - D33 + C32 - C33 = 0
0CONTINUITY34/1:  - D34A - C34A - D34B + C33 - C34B - D34C - D34D = 0
```

```
0CONTINUITY3/1:  - D3 - F3 + D34C - 16.2634 S3 - E3 = -731.855
0CONTINUITY4/1:  - D4 - F4 + D34D - 16.2634 S4 - E4 = -894.489
0SETMRDO/1: C34A <= 1000
0MEETC30MIN/1: C30_MIF <= 3500
0SETKESWICK_MIN/1: KESWICK_MIN = 3500
0EXPORTACTUAL_ALIAS/1:  - D34C - D34D + EXPORTACTUAL = 0
0INFLOW_ALIAS/1:  - C33 + INFLOW = 0
0EXPRATIO__ALIAS/1: EXPRATIO_ = 0.65
0FIND_MAX_EXPORT/1:  - 0.65 INFLOW + EIEXPCTRL = 0
0EXPORT_COMPLY/1: EXPORTACTUAL - EIEXPCTRL <= 0
0MAXLIMITCVP/1: D34C <= 4600
0MAXLIMITSWP/1: D34D <= 6680
0MINLIMITCVP/1: D34C - SURPL0126 + SLACK0126 = 800
0MINLIMITSWP/1: D34D - SURPL0127 + SLACK0127 = 300
0SET_TOTAL/1:  - D34C - D34D + TOTALPUMPING = 0
0SWP_STORAGE_CHANGE/1:  - D2 - C2 + SWPDS = -250
0CVP_STORAGE_CHANGE/1:  - C1 + CVPDS = -1000
0CVPARCSPLIT/1: D34C - D34C_EXP1 - D34C_EXP2 = 0
0SWPARCSPLIT/1: D34D - D34D_EXP2 - D34D_EXP1 = 0
0SRPARCSPLIT/1:  - C34B_CVP - C34B_SWP + C34B = 0
0COA_BALANCE/1:  - D34A - D34B - C34B_CVP - C34B_SWP - UNUSED_FS - UNUSED_SS
   SWPDS + CVPDS - D34C_EXP1 - D34D_EXP1 - IBU + UWFE = 0
0UWFE_FORCE/1:  - 1e+007 INT_IBU_UWFE + UWFE <= 0
0IBU_FORCE/1: 1e+007 INT_IBU_UWFE + IBU <= 1e+007
0CVP_SPLIT/1: 0.2 INT_IBU_UWFE + CVP_SHARE = 0.75
0SWP_SPLIT/1: CVP_SHARE + SWP_SHARE = 1
0COA_CVP3/1: D34B + C34B_CVP + UNUSED_FS - CVPDS + D34C_EXP1 + 0.75 IBU
    - 0.55 UWFE = 0
0COA_SWP3/1: D34A + C34B_SWP + UNUSED_SS - SWPDS + D34D_EXP1 + 0.25 IBU
    - 0.45 UWFE = 0
0SETUNUSED_FS/1:  - UNUSED_FS + D34D_EXP2 <= 0
0SET6UNUSED_SS/1:  - UNUSED_SS + D34C_EXP2 <= 0
0EI_SPLIT_SWP/1:  - 0.5 EIEXPCTRL + D34D_EXP1 - SURPL0159 + SLACK0159 = 0
0EI_SPLIT_CVP/1:  - 0.5 EIEXPCTRL + D34C_EXP1 - SURPL0160 + SLACK0160 = 0

OBJ1 = FREE | OBJ0 = FREE | D30 <= 1000 | D31 <= 1000 | D2 <= 1000 | D33 = 0
D34A <= 1000 | C34A <= 210000 | D34B <= 1000 | D3 <= 1000 | D4 <= 1000 |
C1 <= 50000 | C2 <= 80000 | C30 <= 80000 | C31 <= 80000 | C32 <= 80000 |
C33 <= 80000 | C34B <= 210000 | D34C <= 4600 | D34D <= 6680 | E1 = FREE |
E2 = FREE | E3 = FREE | E4 = FREE | -999999 <= KESWICK_MIN <= 999999 |
EXPORTACTUAL = FREE | INFLOW = FREE | EXPRATIO_ = FREE |
-1e+006 <= SWPDS <= 1e+006 | -1e+006 <= CVPDS <= 1e+006 |
[INT_IBU_UWFE] <= 1 BigM  |

Maximize Solve Number 1
OBJ1: OBJ - 1OBJECTIVE = FREE
OBJ0: OBJ - 0OBJECTIVE = FREE
S1_1: 343809 0OBJECTIVE + 0S1ZONE1/1 - 0STORAGE1/1
S1_2: 6505.38 0OBJECTIVE + 0S1ZONE2/1 - 0STORAGE1/1
S1_3: 3252.69 0OBJECTIVE + 0S1ZONE3/1 - 0STORAGE1/1
S1_4: 1626.34 0OBJECTIVE + 0S1ZONE4/1 - 0STORAGE1/1
S1_5: 487.903 0OBJECTIVE + 0S1ZONE5/1 - 0STORAGE1/1
D30: 2550 0OBJECTIVE - 0CONTINUITY30/1 <= 1000
C30_MIF: 2560 0OBJECTIVE - 0C30TOTAL/1 + 0MEETC30MIN/1
D31: 2550 0OBJECTIVE - 0CONTINUITY31/1 <= 1000
S2_1: 301524 0OBJECTIVE + 0S2ZONE1/1 - 0STORAGE2/1
S2_2: 6342.74 0OBJECTIVE + 0S2ZONE2/1 - 0STORAGE2/1
S2_3: 3090.05 0OBJECTIVE + 0S2ZONE3/1 - 0STORAGE2/1
S2_4: 1463.71 0OBJECTIVE + 0S2ZONE4/1 - 0STORAGE2/1
S2_5: 162.634 0OBJECTIVE + 0S2ZONE5/1 - 0STORAGE2/1
D2: 2550 0OBJECTIVE - 0CONTINUITY2/1 - 0SWP_STORAGE_CHANGE/1 <= 1000
C2_MIF: 2560 0OBJECTIVE - 0C2TOTAL/1 + 0C2MINFLOW/1
D33: 2550 0OBJECTIVE - 0CONTINUITY33/1 = 0
```

```
D34A: 2550 0OBJECTIVE - 0CONTINUITY34/1 - 0COA_BALANCE/1 + 0COA_SWP3/1 <= 1000
C34A: 2550 0OBJECTIVE - 0CONTINUITY34/1 + 0SETMRDO/1 <= 210000
D34B: 2550 0OBJECTIVE - 0CONTINUITY34/1 - 0COA_BALANCE/1 + 0COA_CVP3/1 <= 1000
S3_1: 41797 0OBJECTIVE + 0S3ZONE1/1 - 0STORAGE3/1
S3_2: 6668.01 0OBJECTIVE + 0S3ZONE2/1 - 0STORAGE3/1
S3_3: 6668.01 0OBJECTIVE + 0S3ZONE3/1 - 0STORAGE3/1
S3_4: 650.538 0OBJECTIVE + 0S3ZONE4/1 - 0STORAGE3/1
D3: 420 0OBJECTIVE + 0MAXRELEASE3/1 - 0CONTINUITY3/1 <= 1000
S4_1: 41797 0OBJECTIVE + 0S4ZONE1/1 - 0STORAGE4/1
S4_2: 6668.01 0OBJECTIVE + 0S4ZONE2/1 - 0STORAGE4/1
S4_3: 6668.01 0OBJECTIVE + 0S4ZONE3/1 - 0STORAGE4/1
S4_4: 325.269 0OBJECTIVE + 0S4ZONE4/1 - 0STORAGE4/1
D4: 420 0OBJECTIVE + 0MAXRELEASE4/1 - 0CONTINUITY4/1 <= 1000
S1_6:  - 53669.4 0OBJECTIVE + 0S1ZONE6/1 - 0STORAGE1/1
F1:  - 3400 0OBJECTIVE - 0CONTINUITY1/1
S2_6:  - 34153.2 0OBJECTIVE + 0S2ZONE6/1 - 0STORAGE2/1
F2:  - 3400 0OBJECTIVE - 0CONTINUITY2/1
C34B_CVP:  - 550 0OBJECTIVE - 0SRPARCSPLIT/1 - 0COA_BALANCE/1 + 0COA_CVP3/1
C34B_SWP:  - 550 0OBJECTIVE - 0SRPARCSPLIT/1 - 0COA_BALANCE/1 + 0COA_SWP3/1
UNUSED_FS:  - 450 0OBJECTIVE - 0COA_BALANCE/1 + 0COA_CVP3/1 - 0SETUNUSED_FS/1
UNUSED_SS:  - 450 0OBJECTIVE - 0COA_BALANCE/1 + 0COA_SWP3/1 - 0SET6UNUSED_SS/1
S3_5:  - 10571.2 0OBJECTIVE + 0S3ZONE5/1 - 0STORAGE3/1
F3:  - 3400 0OBJECTIVE - 0CONTINUITY3/1
S4_5:  - 10571.2 0OBJECTIVE + 0S4ZONE5/1 - 0STORAGE4/1
F4:  - 3400 0OBJECTIVE - 0CONTINUITY4/1
C1: 0MAXRELEASE1/1 - 0CONTINUITY1/1 + 0CONTINUITY30/1 - 0CVP_STORAGE_CHANGE/1
    <= 50000
C2: 0C2TOTAL/1 + 0MAXRELEASE2/1 - 0CONTINUITY2/1 + 0CONTINUITY32/1
    - 0SWP_STORAGE_CHANGE/1 <= 80000
C2_EXC:  - 0C2TOTAL/1
C30: 0C30TOTAL/1 - 0CONTINUITY30/1 + 0CONTINUITY31/1 <= 80000
C30_EXC:  - 0C30TOTAL/1
C31:  - 0CONTINUITY31/1 + 0CONTINUITY32/1 <= 80000
C32:  - 0CONTINUITY32/1 + 0CONTINUITY33/1 <= 80000
C33:  - 0CONTINUITY33/1 + 0CONTINUITY34/1 - 0INFLOW_ALIAS/1 <= 80000
C34B:  - 0CONTINUITY34/1 + 0SRPARCSPLIT/1 <= 210000
D34C:  - 0CONTINUITY34/1 + 0CONTINUITY3/1 - 0EXPORTACTUAL_ALIAS/1
   0MAXLIMITCVP/1 + 0MINLIMITCVP/1 - 0SET_TOTAL/1 + 0CVPARCSPLIT/1 <= 4600
D34D:  - 0CONTINUITY34/1 + 0CONTINUITY4/1 - 0EXPORTACTUAL_ALIAS/1
   0MAXLIMITSWP/1 + 0MINLIMITSWP/1 - 0SET_TOTAL/1 + 0SWPARCSPLIT/1 <= 6680
S1: 0STORAGE1/1 - 8.91348 0AREA1/1 - 16.2634 0CONTINUITY1/1
E1: 61.4876 0EVAP1/1 - 0CONTINUITY1/1 = FREE
A1: 0AREA1/1 - 0.220781 0EVAP1/1
S2: 0STORAGE2/1 - 14.1896 0AREA2/1 - 16.2634 0CONTINUITY2/1
E2: 61.4876 0EVAP2/1 - 0CONTINUITY2/1 = FREE
A2: 0AREA2/1 - 0.117924 0EVAP2/1
S3: 0STORAGE3/1 - 13.0406 0AREA3/1 - 16.2634 0CONTINUITY3/1
E3: 61.4876 0EVAP3/1 - 0CONTINUITY3/1 = FREE
A3: 0AREA3/1 - 0.301332 0EVAP3/1
S4: 0STORAGE4/1 - 13.0303 0AREA4/1 - 16.2634 0CONTINUITY4/1
E4: 61.4876 0EVAP4/1 - 0CONTINUITY4/1 = FREE
A4: 0AREA4/1 - 0.301332 0EVAP4/1
KESWICK_MIN: 0SETKESWICK_MIN/1 >= -999999 <= 999999
EXPORTACTUAL: 0EXPORTACTUAL_ALIAS/1 + 0EXPORT_COMPLY/1 = FREE
INFLOW: 0INFLOW_ALIAS/1 - 0.65 0FIND_MAX_EXPORT/1 = FREE
EXPRATIO_: 0EXPRATIO__ALIAS/1 = FREE
EIEXPCTRL: 0FIND_MAX_EXPORT/1 - 0EXPORT_COMPLY/1 - 0.5 0EI_SPLIT_SWP/1
    - 0.5 0EI_SPLIT_CVP/1
SURPL0126:  - 0MINLIMITCVP/1
SLACK0126:  - 2000 0OBJECTIVE + 0MINLIMITCVP/1
SURPL0127:  - 0MINLIMITSWP/1
SLACK0127:  - 2000 0OBJECTIVE + 0MINLIMITSWP/1
TOTALPUMPING: 0SET_TOTAL/1
```

```
SWPDS: 0SWP_STORAGE_CHANGE/1 + 0COA_BALANCE/1 - 0COA_SWP3/1 >= -1e+006 <=
1e+006
CVPDS: 0CVP_STORAGE_CHANGE/1 + 0COA_BALANCE/1 - 0COA_CVP3/1 >= -1e+006 <=
1e+006
[INT_IBU_UWFE]:  - 1e+007 0UWFE_FORCE/1 + 1e+007 0IBU_FORCE/1 + 0.2
0CVP_SPLIT/1
    <= 1 BigM
D34C_EXP1:  - 0CVPARCSPLIT/1 - 0COA_BALANCE/1 + 0COA_CVP3/1 + 0EI_SPLIT_CVP/1
D34C_EXP2:  - 0CVPARCSPLIT/1 + 0SET6UNUSED_SS/1
D34D_EXP2:  - 0SWPARCSPLIT/1 + 0SETUNUSED_FS/1
D34D_EXP1:  - 0SWPARCSPLIT/1 - 0COA_BALANCE/1 + 0COA_SWP3/1 + 0EI_SPLIT_SWP/1
IBU:  - 0COA_BALANCE/1 + 0IBU_FORCE/1 + 0.75 0COA_CVP3/1 + 0.25 0COA_SWP3/1
UWFE: 0COA_BALANCE/1 + 0UWFE_FORCE/1 - 0.55 0COA_CVP3/1 - 0.45 0COA_SWP3/1
CVP_SHARE: 0CVP_SPLIT/1 + 0SWP_SPLIT/1
SWP_SHARE: 0SWP_SPLIT/1
SURPL0159:  - 100 0OBJECTIVE - 0EI_SPLIT_SWP/1
SLACK0159: 0EI_SPLIT_SWP/1
SURPL0160:  - 100 0OBJECTIVE - 0EI_SPLIT_CVP/1
SLACK0160: 0EI_SPLIT_CVP/1

1OBJECTIVE = 0 | 0OBJECTIVE = 0 | 0C2TOTAL/1 = 0 | 0C2MINFLOW/1 <= 1000 |
0C30TOTAL/1 = 0 | 0S1ZONE1/1 <= 550 | 0S1ZONE2/1 <= 1165 | 0S1ZONE3/1 <= 785
0S1ZONE4/1 <= 1100 | 0S1ZONE5/1 <= 400 | 0S1ZONE6/1 <= 552 | 0STORAGE1/1 = 0
0AREA1/1 = 2099.39 | 0EVAP1/1 = 1545.86 | 0MAXRELEASE1/1 <= 12702.7 |
0S2ZONE1/1 <= 29.6 | 0S2ZONE2/1 <= 822.4 | 0S2ZONE3/1 <= 1618 |
0S2ZONE4/1 <= 530 | 0S2ZONE5/1 <= 250 | 0S2ZONE6/1 <= 308 | 0STORAGE2/1 = 0
0AREA2/1 = 172.154 | 0EVAP2/1 = 70.4996 | 0MAXRELEASE2/1 <= 50000 |
0S3ZONE1/1 <= 45 | 0S3ZONE2/1 <= 0 | 0S3ZONE3/1 <= 455 | 0S3ZONE4/1 <= 450
0S3ZONE5/1 <= 22 | 0STORAGE3/1 = 0 | 0AREA3/1 = 1190.72 | 0EVAP3/1 = 535.631
0MAXRELEASE3/1 <= 14376 | 0S4ZONE1/1 <= 55 | 0S4ZONE2/1 <= 0 | 0S4ZONE3/1 <=
445
0S4ZONE4/1 <= 500 | 0S4ZONE5/1 <= 67 | 0STORAGE4/1 = 0 | 0AREA4/1 = 1316.73
0EVAP4/1 = 612.726 | 0MAXRELEASE4/1 <= 14376 | 0CONTINUITY1/1 = -9944.89 |
0CONTINUITY30/1 = 0 | 0CONTINUITY2/1 = -737.903 | 0CONTINUITY31/1 = 0 |
0CONTINUITY32/1 = 0 | 0CONTINUITY33/1 = 0 | 0CONTINUITY34/1 = 0 |
0CONTINUITY3/1 = -731.855 | 0CONTINUITY4/1 = -894.489 | 0SETMRDO/1 <= 1000
0MEETC30MIN/1 <= 3500 | 0SETKESWICK_MIN/1 = 3500 | 0EXPORTACTUAL_ALIAS/1 = 0
0INFLOW_ALIAS/1 = 0 | 0EXPRATIO__ALIAS/1 = 0.65 | 0FIND_MAX_EXPORT/1 = 0 |
0EXPORT_COMPLY/1 <= 0 | 0MAXLIMITCVP/1 <= 4600 | 0MAXLIMITSWP/1 <= 6680 |
0MINLIMITCVP/1 = 800 | 0MINLIMITSWP/1 = 300 | 0SET_TOTAL/1 = 0 |
0SWP_STORAGE_CHANGE/1 = -250 | 0CVP_STORAGE_CHANGE/1 = -1000 |
0CVPARCSPLIT/1 = 0 | 0SWPARCSPLIT/1 = 0 | 0SRPARCSPLIT/1 = 0 |
0COA_BALANCE/1 = 0 | 0UWFE_FORCE/1 <= 0 | 0IBU_FORCE/1 <= 1e+007 |
0CVP_SPLIT/1 = 0.75 | 0SWP_SPLIT/1 = 1 | 0COA_CVP3/1 = 0 | 0COA_SWP3/1 = 0
0SETUNUSED_FS/1 <= 0 | 0SET6UNUSED_SS/1 <= 0 | 0EI_SPLIT_SWP/1 = 0 |
0EI_SPLIT_CVP/1 = 0 |
```