DETERMINATION OF RESERVOIR DAILY OPERATION
POLICIES BY STOCHASTIC DYNAMIC PROGRAMMING

by

C. ANTHONY TSOU
B.A. Hon., Oxford University, 1967.

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in the Department
of
Civil Engineering

We accept this thesis as conforming to the
required standard

THE UNIVERSITY OF BRITISH COLUMBIA
August, 1970
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Civil Engineering

The University of British Columbia  
Vancouver 8, Canada

Date 15 Sept 1970.
ABSTRACT

Reservoir operation policies are often formulated deterministically on the basis of critical flow hydrology. However, if a dynamic river daily flow forecast system is available for the whole season, the forecast information should be fully utilized in reservoir regulation. Given such a forecast system, two approaches to determining optimal daily operation policies for a single purpose flood control reservoir are suggested. Both approaches use stochastic dynamic programming: one involves the minimization of the expected value of flood damages, and the other involves minimizing the probability of occurrence of an undesirable event, which is a flood damage exceeding a certain amount.

The probabilistic approach not only offers a set of alternative optimal daily operation policies, but also indicates the probabilities of being able to achieve the objectives, and thus it forms a basis for comparing and evaluating the alternative objectives.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>BASIC PROBABILITY AND STATISTICS</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>FORECAST SYSTEM</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>DYNAMIC PROGRAMMING</td>
<td>17</td>
</tr>
<tr>
<td>5</td>
<td>APPLICATION OF DYNAMIC PROGRAMMING TO A SINGLE PURPOSE FLOOD CONTROL RESERVOIR</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>REFERENCES</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>APPENDICES</td>
<td>54</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>TITLE</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>PROBABILITY DENSITY FUNCTION</td>
<td>8</td>
</tr>
<tr>
<td>3.1</td>
<td>HYPOTHETICAL HYDROGRAPH</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>MARGINAL PROBABILITY DENSITY FUNCTION</td>
<td>13</td>
</tr>
<tr>
<td>3.3</td>
<td>JOINT BIVARIATE NORMAL DISTRIBUTION DENSITY FUNCTION</td>
<td>14</td>
</tr>
<tr>
<td>3.4</td>
<td>CONDITIONAL PROBABILITY DENSITY FUNCTION</td>
<td>16</td>
</tr>
<tr>
<td>4.1</td>
<td>ONE-STAGE DECISION MODEL</td>
<td>17</td>
</tr>
<tr>
<td>4.2</td>
<td>MULTI-STAGE DETERMINISTIC DECISION MODEL</td>
<td>20</td>
</tr>
<tr>
<td>4.3</td>
<td>MULTI-STAGE STOCHASTIC DECISION MODEL</td>
<td>24</td>
</tr>
<tr>
<td>5.1</td>
<td>PROBABILITY FUNCTION</td>
<td>32</td>
</tr>
<tr>
<td>5.2</td>
<td>FLOOD DAMAGE FUNCTION</td>
<td>34</td>
</tr>
<tr>
<td>5.3</td>
<td>HYPOTHETICAL RESERVOIR RULE CURVE</td>
<td>36</td>
</tr>
<tr>
<td>5.4</td>
<td>MULTI-STAGE STOCHASTIC RESERVOIR OPERATION MODEL</td>
<td>39</td>
</tr>
<tr>
<td>5.5</td>
<td>POSSIBLE RESULTS FROM A PARTICULAR DECISION</td>
<td>42</td>
</tr>
<tr>
<td>5.6</td>
<td>OUTPUT OF THE PROBABILISTIC APPROACH TO</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>DETERMINING OPTIMAL RESERVOIR DAILY OPERATION POLICIES</td>
<td></td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>TITLE</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>OUTPUT OF THE PROBABILISTIC APPROACH TO DETERMINING OPTIMAL RESERVOIR DAILY OPERATION POLICIES</td>
<td>43</td>
</tr>
<tr>
<td>5.2</td>
<td>OUTPUT OF THE EXPECTED VALUE APPROACH TO DETERMINING OPTIMAL RESERVOIR DAILY OPERATION POLICIES</td>
<td>46</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENT

The author wishes to thank Professor L.G. Mitten, Department of Commerce and Business Administration, U.B.C., for his help and guidance in dynamic programming, and Professor S.W. Nash, Department of Mathematics, U.B.C., for his kind assistance in the field of probability theories.

The author also wishes to thank Mr. S.O. Russell, Department of Civil Engineering, U.B.C., for his assistance in the preparation of this thesis.
CHAPTER 1

INTRODUCTION

Dynamic programming was developed by R. Bellman in the 1950's (references 1, 2) but the application of dynamic programming to water resources systems was first introduced by W.A. Hall and N. Buras in 1961 (reference 3). Since then dynamic programming has proved to be a most powerful and versatile optimization technique in the field of water resources both for design and development of operating policies for a single purpose reservoir, a multiple purpose reservoir, or a system of such reservoirs (references 4, 5, 6, 7, 8). However, the operating policy or rule curve of a reservoir is usually developed on the basis of a deterministic input, usually that of the 'critical period' which can be either obtained from the historical data or from generated synthetic flows. While this is usually satisfactory for design purposes, it is less satisfactory for operating purposes where flows are not known in advance. In the case of a reservoir used for flood control an operation policy based on a sound rule curve can minimize flood damage for certain river flows, but it is not likely to minimize flood damage for unusual patterns of flood hydrographs. In certain cases, particularly in an area such as British Columbia where a large proportion of the runoff comes from snowmelt, it is possible to forecast flows with some degree of accuracy. Hopefully, it should be possible to use flow forecasts to improve operating policies.
The purpose of this study is to demonstrate the feasibility of using dynamic programming to determine the optimal daily operating strategy for a reservoir, given a long-range river flow forecast system. The example used in the study is a single purpose flood control reservoir with limited storage, but unlimited discharge capacity, which is operated to minimize the damage downstream of the reservoir for a whole flood season.

Assuming that when the river flow downstream of the reservoir exceeds a certain value $Q_L$, flood damage will occur, then once the inflow to the reservoir $Q$ begins to exceed $Q_L$, the reservoir operator must decide whether to store the excess flow $(Q - Q_L)$ to prevent any immediate damage, to store part of the excess flow, or even not to store any water now in order to reserve storage to be able to prevent greater damage in the future, when the natural river flow reaches its maximum value. If he is to make an appropriate decision, and not operate simply on a rule curve basis, the reservoir operator must have some knowledge of the possible future daily inflows for the whole flood period. Provided with such a long-range flow forecast, and realizing the uncertainties in the forecast value, dynamic programming can assist the reservoir operator to seek a "best" reservoir operating policy such that the possibility and amount of flood damage during the whole season is minimized.

As long as the inflow to the reservoir is not greater than $Q_L$, there is no actual decision to make but to let the water pass the reservoir. However when the inflow to the reservoir reaches $Q_L$ at the beginning of the flood season, a decision on reservoir operating policy has to be made on the basis of estimates of how much and for how long the future river flows
may exceed QL. It is assumed that the forecast system gives a set of daily inflows from then on to the end of the period. If the forecast system is perfect, i.e. if the values of daily flow it predicts will actually occur, then it is a simple matter to decide the reservoir operating policy that makes the best use of the limited reservoir storage capacity to minimize the flood damage caused during the whole season. Such a decision process is deterministic. However, in reality, no forecast system can ever achieve perfect accuracy. Hence, uncertainties in the predicted future flows must be considered in developing a reservoir operating policy. Criteria considered in this study were that either the expected total damage or the probability of occurrence of a flood greater than a certain magnitude be minimized for the considered season. Policies are derived on the basis of forecast future daily inflows and the conditional probability distributions of those predicted flows. Such a decision process is stochastic. This study is based on daily discharges of the Fraser River at Hope, which are believed to be representative of snowmelt flood discharge patterns in British Columbia, and for which flow records are readily available. The reservoir is assumed to be just upstream of Hope, an assumption which simplified the computation but does not invalidate the approach. Since no long range daily flow forecast system was available, a simple forecast system based on linear regression was improvised to provide data for the optimization analysis. Two types of flood damages are considered. One is a recursive type of damage, i.e. the amount of damage done in the past may be repeated in the future if the flood reaches the same level. The other is a non-recursive type of damage, i.e. unless the future flood level exceeds that of the past, no extra damage will be caused in that particular flood season. For the case of recursive type of damage, an
optimal strategy is determined by dynamic programming which minimizes the expected total damage for the whole season. For the case of non-recursive type of damage, again by dynamic programming, an optimal set of decisions is formed such that the probability of the total damage for the entire season exceeding a certain amount is minimized. To state this criterion in reverse, the probability that the total seasonal damage will not exceed a certain amount is maximized.

In the following chapters, some applicable probability theories are first outlined, followed by a description of the simplified flow forecast system which was developed for purposes of the study. Next, a brief summary of dynamic programming and the basic recursive formulae are given, since the study relies so heavily on this relatively new optimization technique. Finally, the two above mentioned approaches are developed and presented together with results and comments. The computer programmes developed in the course of the study are given in Appendices 1, 2, and 3.
Some of the main concepts of probability and statistical analysis used in this study are outlined below. Details can be found in references 9, 10, and 11.

2.1 In the usual language of probability theories, the following symbols are defined:

- \( A \cap B \) : the intersection of \( A \) and \( B \)
- \( P(A \cap B) \) : the probability of event \( A \) and event \( B \) occurring together
- \( P(A) \) : the probability of event \( A \) occurring
- \( P(B) \) : the probability of event \( B \) occurring
- \( P(A/B) \) : the conditional probability of event \( A \) occurring given that event \( B \) has already occurred
- \( P(B/A) \) : the conditional probability of event \( B \) occurring given that event \( A \) has already occurred

then,

\[
P(A \cap B) = P(A) \cdot P(B/A) = P(B) \cdot P(A/B)
\]

If \( A \) and \( B \) are independent events, then

\[
P(B/A) = P(B),
\]
\[
P(A/B) = P(A)
\]

therefore,

\[
P(A \cap B) = P(A) \cdot P(B)
\]
2.2 \[ P(A \cap B \cap \ldots \cap M \cap N \cap \ldots \cap Y \cap Z) \]
\[ = P(A \cap B \cap \ldots \cap M) \times P(N \cap \ldots \cap Y \cap Z / A \cap B \ldots \cap M) \]

i.e. the probability of events A, B, \ldots, M, \ldots, Y, Z occurring together is equal to the probability of events A, B, \ldots, M occurring together multiplied by the conditional probability of events N, \ldots, Y, Z occurring together given that events A, B, \ldots, M have already occurred. Similarly if events A, B, \ldots, M, N, \ldots, Y, Z are independent, then

\[ P(A \cap B \cap \ldots \cap M \cap N \cap \ldots \cap Y \cap Z) \]
\[ = P(A) \times P(B) \times \ldots \times P(M) \times P(N) \times \ldots \times P(Y) \times P(Z) \]

2.3 If X is a discrete random variable with probability function P(X), the expected value of H(X), written as E[H(X)], is defined as:

\[ E[H(X)] = \sum_{X} H(X) \times P(X) \]

2.4 If A, B, \ldots, Y, Z are independent discrete random variables with probability functions P(A), P(B), \ldots, P(Y), P(Z), the expected value of H(A, B, \ldots, Y, Z) is defined as:

\[ E[H(A, B, \ldots, Y, Z)] = \sum_{A} \sum_{B} \ldots \sum_{Y} \sum_{Z} [H(A, B, \ldots, Y, Z) \times P(A) \times P(B) \times \ldots \times P(Y) \times P(Z)] \]

N.B. if the random variables are continuous rather than discrete, then the summation signs are replaced by integral signs.

In engineering studies it is often found more convenient to convert a continuous random variable to a discrete random variable. It will be found that throughout this study, the summation signs rather than the integral signs are used, and that a continuous function (or variable) is treated as a discrete function (or variable).
2.5 The expected value of a random variable $X$ is called the mean value of $X$, is written $\mu_X$, and is defined as:

$$\mu_X = E(X)$$

2.6 The variance of a random variable $X$, written as $\sigma_X^2$, is defined as:

$$\sigma_X^2 = E[(X - \mu_X)^2] = E(X^2) - \mu_X^2$$

Its positive square root is denoted by $\sigma_X$, and is called the standard deviation of $X$.

2.7 If there exist two random variables $X$ and $Y$ then the covariance of $X$ and $Y$ is defined as:

$$\sigma_{xy} = E[(X - \mu_X)(Y - \mu_Y)] = E(XY) - E(X) * E(Y) = E(XY) - \mu_X \mu_Y$$

The covariance gives a measure of how $X$ and $Y$ tend to vary together.

2.8 The correlation coefficient of $X$ and $Y$ is defined as:

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_X \sigma_Y}$$

$\rho_{xy}$ has the property that $|\rho_{xy}| < 1$ for any random variables $X$ and $Y$. If $X$ and $Y$ are independent random variables, then

$$\sigma_{xy} = \rho_{xy} = 0$$

2.9 If a random variable $X$ is Normally Distributed, the probability density function of $X$ is:

$$f(X) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2} \left(\frac{X-\mu}{\sigma}\right)^2}$$

where $\mu$ is the mean of the random variable $X$

$\sigma^2$ is the variance of random variable $X$. 
The probability of having a particular value of $X$ such that $X < X_o$ is

$$P(X < X_o) = \int_{-\infty}^{X_o} f(x) \, dx$$

i.e. the shaded area shown in Fig. 2.1.

2.10 If there exist two random variables $X$ and $Y$ with means and variances $\mu_x$, $\mu_y$ and $\sigma_x^2$, $\sigma_y^2$ respectively, and the correlation coefficient of random variables $X$ and $Y$ is $\rho$, and if $X$ and $Y$ are jointly normally distributed, the joint probability density function is

$$f(X,Y) = \frac{1}{2\pi \sigma_x \sigma_y \sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)} \left[ \frac{(X-\mu_x)^2}{\sigma_x^2} - 2\rho \frac{(X-\mu_x)(Y-\mu_y)}{\sigma_x \sigma_y} + \frac{(Y-\mu_y)^2}{\sigma_y^2} \right]\right)$$

Hence if random variable $X$ is normally distributed and random variables $S$ and $Y$ are jointly normally distributed, then the conditional probability density function of $Y$ given $X = X_o$ is:

$$f(Y|X = X_o) = \frac{1}{2\pi \sigma_y \sqrt{1-\rho^2}} \exp\left(-\frac{1}{2\sigma_y^2(1-\rho^2)} \left[ (Y-\mu_y) - \frac{\rho \sigma_y}{\sigma_x} (X_o - \mu_x) \right]^2 \right)$$
which can be rearranged as:

\[ f(Y|X = X_0) = \frac{1}{2\pi \sigma_y \sqrt{1-\rho^2}} \exp \left\{ -\frac{1}{2} \left( \frac{Y - \left( \mu_y + \rho \frac{\sigma_y}{\sigma_x} (X_0 - \mu_x) \right)}{\sigma_y \sqrt{1-\rho^2}} \right)^2 \right\} \]

i.e. the conditional probability density function of \( Y \) given \( X = X_0 \) is also normally distributed with

mean value = \( \mu_y + \rho \frac{\sigma_y}{\sigma_x} (X_0 - \mu_x) \)

and

variance = \( \sigma_y^2(1 - \rho^2) \)

2.11 Simple linear regression equation:

\[ Y = A + B \times X + E \]

where \( E \) is a random error component with mean = 0, and variance = \( \sigma^2 \)

A and B are constants

The estimates of A and B can be found either by the Least Squares procedure or by the Maximum Likelihood function approach:

\[ a = \frac{\Sigma Y_i}{N} \]

regression coefficient \( b = \rho_{xy} \frac{\sigma_y}{\sigma_x} \)

correlation coefficient = \( \rho_{xy} = \frac{\text{cov}(X,Y)}{\sigma_x \sigma_y} = \frac{\Sigma (X_i Y_i) - NY \bar{Y}}{\Sigma (X_i \bar{Y}_i - NY \bar{Y})} \)

Standard deviation of \( Y \), \( \sigma_y = \sqrt{\text{E}[(Y - \mu_y)^2]} = \sqrt{\frac{\Sigma (Y_i - \bar{Y})^2}{N}} \)
standard deviation of $X$, $\sigma_X = \mathbb{E}[(X - \mu_X)^2] = \sum_{i=1}^{N} (X_i^2) - N\overline{X}^2$

The estimated value of $Y$ is given by:

$$Y = a + b \times (X - \overline{X})$$

The above brief outline of probability theory and statistical analysis is given as background to the development of the long range daily river flow forecast model which is presented in the following chapter.
3.1 Professor J.F. Muir at the Civil Engineering Department, the University of British Columbia has developed a river flow forecast system using multiple regression analysis, reference 12. However, this forecast system only gives five day forecasts. But, in order to determine an optimum operation policy for a flood control reservoir which would minimize the damage for the entire flood season, a long range flow forecast covering the whole season must be used for the optimization analysis. Since Professor Muir's forecast system could not be used, a simple forecast model was improvised for the purpose of this study.

If the period of interest is from \( m \) to \( n \) as shown in Fig. 3.1,

![Hypothetical Hydrograph](image)

then knowing the river flows on and previous to \( m \), and knowing the snow pack remaining to be melted and other meteorological information, it is conceivable that from historical records, mathematical equations may be established to predict flows on future days \((m + i)\),
where \( i = 1, 2, \ldots, (n - m) \)
The significance of these equations can be determined by statistical tests. For simplicity, it is assumed that there exist linear relationships:

\[
Q(k + i) = A + B \times Q(k) + E \tag{3.1}
\]

where

- \( Q \): daily river flow
- \( k = m, (m + 1), (m + 2), \ldots, (n - 2), (n - 1) \)
- \( i = 1, 2, \ldots, (n - k) \)
- \( E \): a random error component which is normally distributed with mean zero and variance \( \sigma^2 \)

This necessitates knowing the actual river flow \( Q(k) \) on day \( k \), and predicts the flow on day \( (k + i) \), with \( Q(k + i) \) depending only on \( Q(k) \). Equation 3.1 is of the standard simple linear regression model outlined in section 2.12.

In this study, the period of interest is arbitrarily chosen from May 15th to June 30th, since daily flow records at the gauging station located at Hope are readily available for this period from 1953 to 1967. They were used as historical data to determine the sets of constants \( A \)'s and \( B \)'s. Forty-five sets of regression equations were obtained. The first set consists of 46 equations in the form:

\[
QP(M16) = \bar{Q}(M16) + b_{M15,M16} \times (Q(M15) - \bar{Q}(M15)) \\
\vdots \\
QP(J30) = \bar{Q}(J30) + b_{M15,J30} \times (Q(M15) - \bar{Q}(M15))
\]

where

- \( QP \) is the predicted river flow
- \( \bar{Q} \) is the mean value of daily flow
- \( M \) for May
- \( J \) for June
The second set consists of 45 equations relating the actual flow on May 16th to predicted flows on the future days, etc.

3.3 The natural river flow on any day \( k \) is a random variable which follows a certain distribution. The shape of this distribution may be determined, if data are available, by plotting the daily flows as a histogram and deriving an equation for this curve. However, for purposes of this study it is assumed that the daily flow \( Q(k) \) follows a normal distribution with mean \( \mu_k \) and variance \( \sigma_k^2 \) as shown in Fig. 3.2.

![Marginal Density Function of Random Variable \( Q(k) \)](image)

Now consider the river flows on two different days \( k \) and \( (k + i) \). Based on the assumption made in section 3.1, the prediction of the river flow on day \( (k + i) \) will depend only upon the river flow on day \( k \), and since \( Q(k) \) and \( Q(k + i) \) are both random variables, then there exists a joint distribution of \( Q(k) \) and \( Q(k + i) \). The nature of this bivariate distribution may be revealed if available data are plotted on a three dimensional scale and an equation may be derived to describe the curved surface which represents the joint probability density function. In this study, the joint distribution is assumed to be bivariately normally distributed as shown in Fig. 3.3.

With this assumption the conditional probability density function of the
Fig. 3.3  JOINT BIVARIATE DENSITY FUNCTION OF RANDOM VARIABLES \(Q(k)\) AND \(Q(k+i)\).
random variable $Q(k+i)$ given that random variable $Q(k)$ has taken a particular value is easily obtained as outlined in section 2.11. The relationship between the conditional distribution function, the simple linear regression equation, and the joint bivariate distribution is shown in Fig. 3.4. From Fig. 3.4 it is seen that the predicted flow $QP(k+i)$ on day $(k+i)$ by equation 3.1 is nothing else but one of the possible flows on day $(k+i)$, except that $QP(k+i)$, in the forecast model, has the maximum probability of occurrence.

3.4 A sophisticated forecast model, say for the flow on day $(k+i)$ would depend not only upon the flow on day $k$, but also upon all other important factors. If those other factors were also random variables, then they would constitute a multivariate distribution. However, for present purposes our interest is not in the individual distribution of $Q(k+i)$, but in the conditional distribution of $Q(k+i)$. The three assumptions made in this forecast system are:

(1) the river flow on any day $k$ follows a normal distribution
(2) the river flow on day $(k+i)$ depends only on river flow on day $k$, and a linear relationship exists between them
(3) the joint distribution of the river flows on day $(k+i)$ is a bivariate normal distribution

It is realized that these assumptions oversimplify the real situation, but since the emphasis of this study is on the optimization analysis, this forecast system is considered adequate as a source of data to illustrate the optimization process. If a better forecast model were developed, the optimization process would not change at all.
Jooint bivariate density function of random variables $Q(K)$ and $Q(K+i)$.

Conditioned density function of random variable $Q(K+i)$ given $Q(K)$.

Fig. 3.4
CHAPTER 4

DYNAMIC PROGRAMMING

Dynamic programming is one of the optimization methods employed in Operations Research studies. Optimization means finding a best solution among several feasible alternatives. Dynamic programming is based on:

(1) the philosophy of breaking a complex problem into a series of smaller problems, and then combining the results of the solution of the smaller problems to obtain the solution of the whole complex problem.

(2) the principle of optimality, as stated by Bellman: "An optimal policy has the property that whatever the initial state and decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

Some of the concepts of dynamic programming, as used in this study are outlined below. Details can be found in reference 13. The terminology follows normal usage

4.1 A one-stage decision system:
In Fig. 4.1, the box represents a stage and the notations represent the following quantities:

1. an input state variable X, that gives all the relevant information about inputs to the stage; it represents the condition of the system at the beginning of this stage
2. an output state variable Y, that gives all the relevant information about output from the stage; it represents the condition of the system at the end of this stage
3. a decision variable D, that controls the operation of the stage
4. a stage return R, a single-valued function of inputs, decision and outputs. It is a measure of the utility obtained from the stage when the input is X, the decision selected is D, and the resulting output is Y.
   \[ R = r(X,D,Y) \]  \hspace{1cm} \text{(4.1)}
5. a stage transformation t, a single-valued transformation also known as the stage-coupling function, expressing the state at the end of the stage as a function of the decision variable and the state at the beginning of the stage
   \[ Y = t(X,D) \]  \hspace{1cm} \text{(4.2)}

Substitute Eq. 4.2 into Eq. 4.1
\[
R = r(X,D,Y) \\
= r(X,D, t(X,D)) \\
= r(X,D)**
\]

\[**\text{Mathematical Rigor would require } R = r(X, D, t(X, D)) = g(X, D) \text{ to distinguish the return function with two arguments from the function with these arguments, as the functional nature will change. However, to simplify notation, we use } r \text{ to represent the stage return function, thus } r(X, D, Y), \]
\[r(X, D, t(X, D)), \text{ and } r(X, D) \text{ are alternative expressions for the stage return } r.\]
The one-stage initial state optimization problem is to choose D such that the stage return R is maximized (or minimized). Denoting f(X) as the optimal return and D* = D(X) as the optimal decision, we have

\[ r(X, D^*) = r(X, D(X)) = f(X) = \max_D r(X, D) \]

where \( \max_D r(X, D) \) means the maximum value of \( r(X, D) \) among all the possible values the decision variable D can take.

If the initial state X is not fixed, and we are free to choose X, then there may exist a particular value of X such that

\[ f(X^*) = \max_X f(X) = \max_{X,D} r(X, D) \]

By the transformation

\[ Y = t(X, D) \]

the inversion of this transformation t will give

\[ X = t^{-1}(Y, D) \]

then,

\[ R = r(X, D, Y) = r(t(Y, D), D, Y) = r(D, Y) \]

therefore the one-stage final state optimization problem is to choose D as a function of the final state Y, \( D^* = D(Y) = f(Y) = \max_D r(Y, D) \)

4.2 Serial multistage decision system:

A serial multistage system consists of a set of stages joined together in series so that output from one stage becomes the input to the next stage return and transformation as shown in Fig.4.2.
For a general stage \( i \), \((i = 1, 2, \ldots, N)\) of the \( N \)-stage system, the stage transformation is

\[
X_{i-1} = t_i(X_i, D_i)
\]  

(4.3)

and the stage return

\[
R_i = r_i(X_i, D_i)
\]  

(4.4)

Similarly for stage \((i+1)\)

\[
X_i = t_{i+1}(X_{i+1}, D_{i+1})
\]  

(4.5)

Combining Eq. 4.3 and Eq. 4.5

\[
X_{i-1} = t_i(X_{i+1}, D_{i+1}, D_i)
\]

If we trace the stages further up, i.e. in the direction from stage 1 to stage \( N \) in Fig. 4.2, eventually we obtain

\[
X_{i-1} = t_i(X_N, D_N, D_N, \ldots, D_{i+1}, D_i)
\]  

(4.6)

i.e. input state variable \( X_{i-1} \) to the \((i-1)\)th stage depends only on the very initial condition of the system and all the decisions made prior to the \((i-1)\)th stage. From Eq. 4.4 it is seen the return of \( i \)th stage is a function of the
input to ith stage and the decision made at the ith stage
\[ R_i = r_i(X_i, D_i) \]

now by Eq. 4.6
\[ X_i = t_{i+1}(X_N, D_N, D_{N-1}, \ldots, D_{i+2}, D_{i+1}) \]

Hence
\[ R_i = r_i(X_N, D_N, D_{N-1}, \ldots, D_{i+2}, D_{i+1}, D_i) \quad (4.7) \]

i.e. the return of the ith stage depends only on the very initial condition of
the system and all the decisions made prior to and during the ith stage. In
other words, the decision made at the ith stage influences only those returns
from stage 1 to stage i. The total return \( R_T \) from stage 1 to stage N is some
function of the individual stage returns
\[
R_T(X_N, X_{N-1}, \ldots, X_1, D_N, D_{N-1}, \ldots, D_1) = G[r_N(X_N, D_N), r_{N-1}(X_{N-1}, D_{N-1}), \ldots, r_1(X_1, D_1)] \quad (4.8)
\]

The N-stage initial state optimization problem is to maximize (or minimize) the
N-stage total return \( R_T \) over the decision variables \( D_1, D_2, \ldots, D_N \), i.e. to
find the optimal return as a function of the initial state \( X_N \).

Denoting \( f_N(X_N) \) as the maximum N-stage return,
\[
D_i^* = D_i(X_N), \text{ the optimal ith stage decision, with} \quad i = 1, 2, \ldots, N
\]
\[
X_i^* = t_i(X_N), \text{ the optimal ith stage input state, with} \quad i = 1, 2, \ldots, (N-1)
\]

Hence we have
\[
f_N(X_N) = G[r_N(X_N, D_N^*), r_{N-1}(X_{N-1}^*, D_{N-1}^*), \ldots, r_1(X_1^*, D_1^*)]
= \max_{D_1^* \ldots D_N^*} G[r_N(X_N, D_N), r_{N-1}(X_{N-1}, D_{N-1}), \ldots, r_1(X_1, D_1)] \quad (4.9)
\]
Subject to

\[ X_{i-1} = t_i(X_i, D_i) \quad i = 1, 2, \ldots, N \]

In order to break the complex optimal N-stage return \( f_N(X_N) \) as expressed in Eq. 4.9 into a series of simpler problems, we must achieve the crucial step of moving the maximization with respect to \( D_{N-1}, D_{N-2}, \ldots, D_1 \), inside the Nth stage return. This is known as 'decomposition'. Sufficient conditions for achieving this important change in the position of maximizations have been given by L.G. Mitten (Reference 14). These conditions are:

1. **Separability**, i.e.

   \[ G[r_N(X_N, D_N), r_{N-1}(X_{N-1}, D_{N-1}), \ldots, r_1(X_1, D_1)] \]

   \[ = G_1(r_N(X_N, D_N), G_2[r_{N-1}(X_{N-1}, D_{N-1}), \ldots, r_1(X_1, D_1)] \}

   where \( G_1 \) and \( G_2 \) are real-valued functions, and

2. **Monotonicity**, i.e.

   \( G_1 \) is a monotonically nondecreasing function of \( G_2 \)

   for every \( r_N \)

When these two conditions are satisfied, the vital decomposition can be achieved, namely,

\[
\max_{D_N, D_{N-1}, \ldots, D_1} G[r_N(X_N, D_N), r_{N-1}(X_{N-1}, D_{N-1}), \ldots, r_1(X_1, D_1)]
\]

\[
= \max_{D_N} G_1(r_N(X_N, D_N), \max_{D_{N-1}, \ldots, D_1} G_2[r_{N-1}(X_{N-1}, D_{N-1}), \ldots, r_1(X_1, D_1)]) \quad (4.10)
\]

It has also been known that if the total N-stage return is either the sum of the N-stages individual returns, i.e.

\[ R_T = r_N(X_N, D_N) + r_{N-1}(X_{N-1}, D_{N-1}) + \ldots + r_1(X_1, D_1) \]
or the product of the N-stages non-negative individual returns, i.e.
\[ R_T = r_N(X_N, D_N) \times r_{N-1}(X_{N-1}, D_{N-1}) \times \ldots \times r_1(X_1, D_1) \]
then the decomposition can always be achieved. Let the operator "o" denote either one of the above mathematical operations, namely addition or multiplication, we have,
\[ f(N) = \max_{D_N, D_{N-1}, \ldots, D_1} \{ r_N(X_N, D_N) \text{ or } r_{N-1}(X_{N-1}, D_{N-1}) \circ \ldots \circ r_1(X_1, D_1) \} \]
by decomposition, we have,
\[ f(N) = \max_{D_N} \{ r_N(X_N, D_N) \circ \max_{D_{N-1}} r_{N-1}(X_{N-1}, D_{N-1}) \circ \ldots \circ r_1(X_1, D_1) \} \]
\[ = \max_{D_N} \{ r_N(X_N, D_N) \circ f_{N-1}(X_{N-1}) \} \]

4.3 Deterministic and stochastic decision making

The recursion equation derived in section 4.2 is applicable to deterministic models where for each stage once a decision is made, the return (optimal or not) from that stage is unambiguously specified, and the state resulting from that decision is also unambiguously specified. In these cases, decision making is under certainty. But there are situations where for each stage, once a decision is made neither the return from that stage, nor the state resulting from that decision are unambiguously specified, but rather there is a set of possible returns and states due to chance occurrences. These cases are known as decision making under risk or stochastic decision making.

4.4 Serial multistage decision system with uncertainty

An N-stage stochastic system is similar to an N-stage deterministic system except that at each stage there is a random variable that affects the
stage return and transformation as shown in Fig. 4.3.

\[ R_i = r_i(X_i, D_i, k_i) \]  

(4.13)

The transformation at each stage is

\[ X_{i-1} = t_i(X_i, D_i, k_i) \]  

(4.14)

The random variable \( k_i \) may take on a series of values which follow a certain probability distribution \( P(k_i) \); hence, given a fixed initial input state \( X_i \) to the ith stage and a fixed decision \( D_i \), \( R_i \) and \( X_{i-1} \) cannot be specified exactly due to the random variable \( k_i \). Similar to Eq. 4.14 we get

\[ X_i = t_{i+1}(X_{i+1}, D_{i+1}, k_{i+1}) \]  

(4.15)

Since \( k_{i+1} \) is another random variable with a probability function \( P(k_{i+1}) \), \( X_i \) cannot be specified exactly either. Hence, for a given initial condition \( X_N \),
the state variables $X_i$

$$i = 1, 2, \ldots, (N-1)$$

the stage returns $R_i$

$$i = 1, 2, \ldots, N$$

are all random variables. Combining Eq. 4.13 and Eq. 4.15

$$R_i = r_i(X_i, D_i, k_i)$$

$$= r_i(t_{i+1}(X_{i+1}, D_{i+1}, k_{i+1}), D_i, k_i)$$

$$= r_i(X_i, D_{i+1}, D_i, k_{i+1}, k_i)$$

If we trace the stages further up, i.e. in the direction from stage 1 to stage N in Fig. 4.3, eventually we obtain

$$R_i = r_1(X_N, D_N, \ldots, D_i, k_N, \ldots, k_i) \quad (4.16)$$

Eq. 4.16 reveals that

(1) the decision made at the $i$th stage influences only those returns from stage 1 to stage $i$ (as in the deterministic model)

(2) the $i$th stage return does not depend on random variables $k_1, k_2, \ldots, k_{i-1}$.

If the total return from the $N$ stages is the sum of the individual stage returns, then

$$R_T(X_N, X_{N-1}, \ldots, X_1, D_N, D_{N-1}, \ldots, D_1, k_N, k_{N-1}, \ldots, k_1)$$

$$= \sum_{i=1}^{N} r_i(X_i, D_i, k_i)$$

$$= r_N(X_N, D_N, k_N) + r_{N-1}(X_{N-1}, D_{N-1}, k_{N-1}) + \ldots + r_1(X_1, D_1, k_1)$$
subject to $x_{i-1} = t_i(x_i, d_i, k_i), i = 1, 2, \ldots N.$

Take the return at stage (N-1) as an example. If $x_{N-1}$ is fixed, the $r_{N-1}$ is a function of the random variable $k_{N-1}$, hence by section 2.3, the expected value of the (N-1)th stage return is

$$E[r_{N-1}(x_{N-1}, d_{N-1}, k_{N-1})] = \sum_{k_{N-1}} [p_{N-1}(k_{N-1}) * r_{N-1}(x_{N-1}, d_{N-1}, k_{N-1})]$$

where $\sum$ means the summation of all the possible values the random variable $k_{N-1}$ may take.

$p_{N-1}(k_{N-1})$ is the probability function of $k_{N-1}$.

But since $x_{N-1} = t_N(x_N, d_N, k_N)$, therefore $x_{N-1}$ is itself a function of another random variable $k_N$. Hence taking consideration of the uncertainty in $x_{N-1}$, resulting from random effects at stage $N$, the expected return from stage (N-1) is

$$E[r_{N-1}(x_{N-1}, d_{N-1}, k_{N-1})]$$

$$= \sum_{k_N} \sum_{k_{N-1}} p_N(x_{N-1}) * \{ \sum_{k_{N-1}} p_{N-1}(k_{N-1}) * r_{N-1}(x_{N-1}, d_{N-1}, k_{N-1}) \}$$

$$= \sum_{k_N} p_N(k_N) * \{ \sum_{k_{N-1}} p_{N-1}(k_{N-1}) * r_{N-1}(x_{N-1}, d_{N-1}, k_{N-1}) \}$$

(4.17)

Therefore, following a similar argument in the derivation of Eq. 4.17, the total expected return from N stages is
\[ \bar{R}_T(X_N, \ldots, X_1, D_N, \ldots, D_1) \]

\[ = \sum_{k_N} P_N(k_N) \times r_N(X_N, D_N, k_N) + \sum_{k_N} P_N(k_N) \times \left[ \sum_{k_{N-1}} P_{N-1}(k_{N-1}) \times r_{N-1}(X_{N-1}, D_{N-1}, k_{N-1}) \right] + \ldots \]

\[ + \sum_{k_N} P_N(k_N) \times \left[ \sum_{k_{N-1}} P_{N-1}(k_{N-1}) \times \ldots \times \left[ \sum_{k_1} P_1(k_1) \times r_1(X_1, D_1, k_1) \right] \right] \]

\[ = \sum_{k_N} P_N(k_N) \times \left( r_N(X_N, D_N, k_N) + \sum_{k_{N-1}} P_{N-1}(k_{N-1}) \times r_{N-1}(X_{N-1}, D_{N-1}, k_{N-1}) \right) \]

\[ + \ldots + \sum_{k_{N-1}} P_{N-1}(k_{N-1}) \times \left[ \sum_{k_{N-2}} P_{N-2}(k_{N-2}) \times \ldots \times \left[ \sum_{k_1} P_1(k_1) \times r_1(X_1, D_1, k_1) \right] \right] + \ldots + \]

\[ \sum_{k_{N-1}} P_{N-1}(k_{N-1}) \times \left[ \sum_{k_{N-2}} P_{N-2}(k_{N-2}) \times \ldots \times \left[ \sum_{k_1} P_1(k_1) \times r_1(X_1, D_1, k_1) \right] \right] \]

Subject to \( X_{i-1} = t_i(X_i, D_i, k_i), i = 1, 2, \ldots, N \)

\( \bar{f}_N(X_N) \), as a function of the initial condition \( X_N \), is defined to be the total minimized expected return from \( N \) stages, and by the principle of optimality,
we obtain,

$$
\bar{f}_N(X) = \max_{D_N, D_{N-1}, \ldots, D_1} \bar{R}_N(X_N, \ldots, X_1, D_N, \ldots, D_1)
$$

$$
= \max_{D_N, \ldots, D_1} \{ \sum_{k_N} P_N(k_N) \cdot r_N(X_N, D_N, k_N) + \sum_{k_{N-1}} P_{N-1}(k_{N-1}) \cdot r_{N-1}(X_{N-1}, D_{N-1}, k_{N-1})

+ \ldots + \sum_{k_{1}} P_1(k_1) \cdot r_1(X_1, D_1, k_1) \} \ldots \} + \ldots +

\sum_{k_{N-1}} P_{N-1}(k_{N-1}) \cdot \{ \sum_{k_{N-2}} P_{N-2}(k_{N-2}) \cdot \ldots \cdot \{ \sum_{k_{1}} P_1(k_1) \cdot r_1(X_1, D_1, k_1) \} \ldots \} } (4.18)

$$

Since by definition
Combining Eq. 4.18 and Eq. 4.19, we obtain,

$$
\bar{f}_{N-1}(X_{N-1}) = \max_{D_{N-1}, \ldots, D_1} \left\{ \sum P_{N-1}(k_{N-1}) \cdot \{ r_{N-1}(X_{N-1}, D_{N-1}, k_{N-1}) + \right.
\left. \sum P_{N-2}(k_{N-2}) \cdot r_{N-2}(X_{N-2}, D_{N-2}, k_{N-2}) + \ldots \right. \\
\left. + \sum P_{N-2}(k_{N-2}) \cdot \left[ \sum P_{N-3}(k_{N-3}) \cdot \ldots \cdot \left[ \sum P_i(k_i) \cdot r_i(X_i, D_i, k_i) \right] \ldots \right] \right\} \\
+ \ldots \sum P_{N-2}(k_{N-2}) \cdot \left[ \sum P_{N-3}(k_{N-3}) \cdot \ldots \cdot \left[ \sum P_i(k_i) \cdot r_i(X_i, D_i, k_i) \right] \ldots \right] \right\}
$$

(4.19)

To generalize this expression, we obtain,

$$
\bar{f}_i(X_i) = \max_{D_i, k_i} \left\{ \sum P_i(k_i) \cdot \left[ r_i(X_i, D_i, k_i) + \bar{f}_{i-1}(X_{i-1}) \right] \right\}
$$

$$
i = 1, 2, \ldots, N
$$

(4.20)

This is the stochastic recursion formula in dynamic programming. As pointed out by A. Kauffman (Reference 15) in the face of an uncertain future, the optimization can only be carried out in one direction: by proceeding from the future towards the past, namely to proceed against the direction of the arrows, i.e. from stage 1 to stage N. This is known as backward computation or upstream flow.

This rule is the consequence of the stochastic multistage method by which this expected value is calculated. However, in a deterministic model, if the stage-coupling functions $t_i$

$$
X_{i-1} = t_i(X_i, D_i) \quad i = 1, 2, \ldots N
$$
have inverse functions $t_i^{-1}$

$$X_i = t_i^{-1}(X_{i-1}, D_i) \quad i = 1, 2, \ldots, N$$

then the optimization can be carried out in either forward or backward sequence. Engineering studies of water resources systems often encounter river flows as one of the input data. Since river flow is a random variable, problems of water resources systems are frequently stochastic by nature.

4.5 Computational Scheme

In the above derivation of the recursive formulae of deterministic and stochastic dynamic programming, block diagram representations of the sequential decision process were shown in Figs. 4.2 and 4.3. The backward numbering of stages is quite standard in dynamic programming, the reason for such practice probably being the sequence of decisions made in solving a multistage decision problem. As in Fig. 4.2, $X_0$ quite often denotes the final system condition and $X_N$ denotes the initial system condition; and frequently in dynamic programming a problem can only be solved if one proceeds from the final condition $X_0$ to determine a set of decisions which will lead to the initial condition $X_N$ optimally. Hence practical computational scheme is also often proceeded backwardly from stage 1 to stage N. The backward approach will be used in the formulation in Chapter 5.
CHAPTER 5

APPLICATION OF DYNAMIC PROGRAMMING TO A SINGLE FLOOD CONTROL RESERVOIR

The fictitious reservoir is assumed to have a storage capacity of 500,000 sfd and for simplicity it is also assumed that the reservoir is able to discharge as much water as desired.

The optimization of the reservoir operating procedure is treated as a sequential decision process. The process of operating a dam is continuous since inflow varies continuously with time, and the reservoir storage function is also continuous. However, the most convenient way to formulate the problem for computation on a digital computer is to consider a series of time increments, and to treat the inflows as discrete variables. The volume of water in storage will thus also appear to change in stepwise manner. The size of the increment in flow and in the volume of water stored in the reservoir depends on the accuracy one wishes to obtain. In this study, the inflow is considered having an increment of 10,000 cfs, and the total reservoir storage is divided into 50 increments, each increment equals to 10,000 sfd.

5.1 Discretization of the inflow to the reservoir.

As described in section 3.3, knowing the actual riverflow on day k, the conditional distributions of flows on future days are all assumed to be normally distributed.

For normal distribution, the probability of $X$ having a value less than or equal to $X_o$ is (section 2.9)

$$ P(X \leq X_o) = \frac{1}{\sqrt{2\pi} \sigma} \int_{-\infty}^{X_o} e^{-\frac{(X - \mu)^2}{2\sigma^2}} dx $$

Let $t = \frac{X - \mu}{\sigma}$

$$ dt = \frac{1}{\sigma} dx $$

therefore,

$$ P(X \leq X_o) = \int_{-\infty}^{\frac{X_o - \mu}{\sigma}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt $$
The following approximation formula for this integration is proposed by C. Hastings Jr. (reference 13)

\[ P(X \leq X_0) = 1.0 - Z(t) \ast (a_1 \ast Y + a_2 \ast Y^2 + a_3 \ast Y^3) \ldots (5.1) \]

where \( X_0 \geq \mu \)
\[ t = \frac{X_0 - \mu}{\sigma} \]
\[ Y = 1.0 / (1.0 + p \ast t) \]
\[ Z(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} \]
\[ p = 0.33267 \]
\[ a_1 = 0.43618 \quad a_2 = -0.12016 \quad a_3 = -0.93729 \]

N.B. If \( X_0 < \mu \), then let \( t = \frac{X_0 - \mu}{\sigma} \) and

\[ P(X \leq X_0) = Z(t) \ast (a_1 \ast Y + a_2 \ast Y^2 + a_3 \ast Y^3) \]

Using Eq. 5.1, the shaded area under a normal distribution curve is readily obtainable, hence converting the continuous conditional distribution to a discrete conditional distribution to calculate the conditional probabilities associated with the possible river flow on any future day. For example, knowing
the actual river flow on May 19th

\[ Q(M19) = 200,000 \text{ cfs} \]

by Eq. 3.1 the mean value of the flow on May 20th is 210,000 cfs with a standard deviation equal to

\[ \sigma_{M20} = \sqrt{1-p^2_{M19,M20}} = 52,000 \times \sqrt{1-(0.9725)^2} \]

\[ = 52,000 \times 0.234 = 12,000 \text{ cfs}. \]

The conditional continuous normal distribution of flow on May 20th is discretized to give:

<table>
<thead>
<tr>
<th>Possible river flow on May 20(10^2 cfs)</th>
<th>Probability of occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>170</td>
<td>.001</td>
</tr>
<tr>
<td>180</td>
<td>.017</td>
</tr>
<tr>
<td>190</td>
<td>.087</td>
</tr>
<tr>
<td>200</td>
<td>.233</td>
</tr>
<tr>
<td>210</td>
<td>.323</td>
</tr>
<tr>
<td>220</td>
<td>.233</td>
</tr>
<tr>
<td>230</td>
<td>.087</td>
</tr>
<tr>
<td>240</td>
<td>.017</td>
</tr>
<tr>
<td>250</td>
<td>.001</td>
</tr>
</tbody>
</table>

5.2 Flood damage function.

Flood damage is usually measured in terms of money, and the actual determination of the damage function is quite a formidable task. Since the object of this study is to demonstrate the application of dynamic programming to determine an optimal reservoir operation policy, the specific nature of the damage function is not examined in detail. It is assumed that when the river discharge downstream from the reservoir exceeds \( Q_L = 200,000 \) cfs, damage will begin to occur and damage will reach a maximum when the flow exceeds \( Q_U = 500,000 \) cfs. The values of \( Q_U \) and \( Q_L \) used in this study are arbitrarily chosen, as is the damage function itself. Since the actual form of the damage function is

* The probability of occurrence of a flow of say 180,000 cfs is taken as the probability that the flow will fall between 175,000 and 185,000 cfs.
immaterial as far as the approach to optimization analysis is concerned, for simplicity it is assumed here that the larger the river flow is the greater is the flood (i.e. the damage done by river flow $Q_1$ is greater than the damage done by river flows $Q_2$, $Q_3$, ..., $Q_n$, if $Q_1 \geq Q_2 + Q_3 + ... + Q_n$, and $Q_1 > Q_L$, for $i = 2, 3, ..., n$). Hence the flood damage function is of the shape shown in Fig. 5.2.

![Fig. 5.2](image)

Two types of damage are considered in the study, namely:

1. non-recursive damage, which is defined as follows: unless the river discharge downstream from the reservoir exceeds the maximum discharge experienced so far in that year, no extra damage will be done even if the flow exceeds $Q_L$.

2. recursive damage, which is defined as follows: as long as the discharge downstream of the reservoir exceeds $Q_L$, damage will be caused in accordance with the given flood damage function.

5.3 General description of a flood control reservoir operation policy.

The purpose of a flood control reservoir is to retain all the excess water which would cause damage (if the storage capacity is large enough), or to store part of the excess water to minimize the inevitable damage (if the storage is not great enough).
If the inflows during the entire flood season are known with certainty, then the reservoir operating policy is simple to form, namely to begin closing the gates at point c, and maintain a constant discharge such that the reservoir will be full at point d, as shown in Fig.5.3. Although damage is done during period ab, the maximum downstream discharge is only $Q_0$ rather than $Q_{\text{max}}$; therefore the inevitable damage is minimized.

When the inflows for the entire flood season are not known with certainty, then it is a difficult task to decide when to store water and how much excess water to store, since an immediate avoidance of a small damage may result in a much larger damage in the future. Therefore the reservoir operation policy must have the entire flood season in mind, and the policy should be such that the total seasonal damage is as small as possible.

Since the future river inflow is uncertain, although the range of possible values the future inflow may take on any particular day is known from the forecast, it is impossible to know beforehand how much damage will result due to a particular decision. An example is given below to illustrate this point.

If a forecast system indicates that the inflow on a particular day may take any one of the following values with the associated probabilities

\[
\begin{array}{cccccccccccc}
Q(10^3 \text{ cfs}) & 170 & 180 & 190 & 200 & 210 & 220 & 230 & 240 & 250 \\
\hline
P(Q) & .001 & .017 & .087 & .233 & .323 & .323 & .087 & .017 & .001
\end{array}
\]

and suppose the reservoir to be half full at the beginning of that day, and a decision made on that day not to store any inflow. Hence, due to this decision, the reservoir will still be half full at the end of that day. But what will be the river flow downstream of the reservoir (and the corresponding damage) resulting from that decision? Two possible ways of
assessing this situation are:

(1) expected value approach:

Since the river flow on that day may take any value from 170,000 cfs to 250,000 cfs with associated probabilities of occurrence (thus the outflow is of the same nature), the expected value of the river flow on that day is

\[
\sum_{Q=170,000}^{250,000} \text{P}(Q) \cdot Q = 10^3 \times [170 \times 0.001 + 180 \times 0.017 + \ldots + 240 \times 0.017 + 250 \times 0.001]
\]

\[= 210,000 \text{ cfs.}\]

The expected value has a statistical significance, in that it represents an estimate of the average value of river flows likely to occur on that day, even though the actual river flow on that day may not coincide with the expected value.

(2) probability approach:

Since the probability of flow being equal to 170,000 cfs is 0.001 and the probability of flow being equal to 180,000 cfs is 0.017, one can say that the probability of flow being not greater than 180,000 cfs is 0.018.

When one deals with probability, unless the value is 1.0, one can never be certain whether an event will take place or not.

From this example, it is seen that the definite river flow on that day cannot be pin-pointed. Since the flood damage is dependent on the outflow from reservoir, which in turn depends on the inflow to the reservoir, the damage itself is a random variable. Therefore, the damage resulting from any particular decision cannot be pin-pointed either. The probabilistic nature of the answer to such problems may seem rather inprecise to a deterministic


frame of mind, but this is a typical characteristic of any stochastic process.

As discussed in the following sections, dynamic programming as an optimization technique can be used to find optimal reservoir operating policy which is either to minimize the total expected damage, or to maximize the probability of a damage not exceeding a certain amount for the entire flood season.

5.4 Non-recursive type of damage.

Since the maximum possible damage done on a certain day is given in Fig.5.2, and since the damages are non-recursive, the maximum possible damage that might result for the entire flood season is also known.

A valid criterion for the reservoir operating policy would be to guarantee that the damage for the whole flood season does not exceed a certain percentage of the maximum possible damage. This means that our desire is to operate the reservoir in such a manner that the inevitable damage is contained within a% of the maximum possible damage. Hence the objective is to find a reservoir operating policy which will maximize the probability of occurrence of an event which we wish to occur, i.e. maximize the probability of limiting the damage to some figure which has been decided upon.

In the following formulation, a stage refers to a day. The variables are defined as follows:

state variable $X_i$: level of water in reservoir at the beginning of stage $i$
decision variable $D_i$: decision made at stage $i$ to operate the reservoir such that the aim is to let the water level in reservoir at the end of stage $i$ be at $Y_{i-1}$
random variable $Q_i$: river inflow to the reservoir during stage $i$
state variable $X_{i-1}$: the actual water level in reservoir at the end of stage $i$, although the decision $D_i$ is made to aim for a level $Y_{i-1}$
\[ X_{i-1} = t_i (X_i, D_i, Q_i) \]

\( \alpha \) : percentage of maximum possible damage

\[ P(R_i \leq \alpha / X_i, D_i) \] : probability of causing damage not greater than \( \alpha \% \) of the maximum possible damage during stage \( i \) only, given that at the beginning of stage \( i \) the water level in reservoir is at \( X_i \), and a decision \( D_i \) is made to aim for a water level \( Y_{i-1} \) at the end of the stage \( i \)

\( f_i(X_i, \alpha) \) : total maximized probability of damage not exceeding \( \alpha \% \) of the maximum possible damage from stage \( i \) to stage \( 1 \), where stage \( 1 \) is the end of the flood season.

Return \( R_i \) : the flood damage suffered at stage \( i \) only

By definition of \( f_i(X, \alpha) \):

\[ f_i(X_i, \alpha) = \max_{D_i, \ldots, D_1} P(R_1, R_{i-1}, \ldots, R_i \leq \alpha \mid X_1, D_1, D_{i-1}, \ldots, D_1) \]

As pointed out in section 4.4, the decision made at state \( i \) influences only those returns from stage \( i \) to stage \( 1 \), or to state it in reverse, the decisions made during stage \( 1 \) to stage \( (i-1) \) have no influence whatsoever on the return from stage \( i \).
Hence,

\[ P(R_1, R_{1-1}, \ldots, R_1 \leq \alpha \mid X_1, D_1, D_{1-1}, \ldots, D_1) \]

\[ = P(R_1 \leq \alpha \mid X_1, D_1) \times P(R_{1-1}, \ldots, R_1 \leq \alpha \mid R_1 \leq \alpha, X_1, D_1, \ldots, D_1) \]

\[ = P(R_1 \leq \alpha \mid X_1, D_1) \times \sum_{X_{i-1}} P(X_{i-1}, R_{i-1}, \ldots, R_1 \leq \alpha \mid R_1 \leq \alpha, X_1, D_1, \ldots, D_1) \]

\[ = P(R_1 \leq \alpha \mid X_1, D_1) \times \sum_{X_{i-1}} [P(X_{i-1} \mid R_1 \leq \alpha, X_1, D_1) \times P(R_{i-1}, \ldots, R_1 \leq \alpha \mid X_{i-1}, D_{i-1}, \ldots, D_1)] \]

therefore,

\[ f_i(X_1, \alpha) = \max_{D_1, \ldots, D_1} \{ P(R_1 \leq \alpha \mid X_1, D_1) \times P(R_{i-1}, \ldots, R_1 \leq \alpha \mid R_1 \leq \alpha, X_1, D_1, \ldots, D_1) \} \]

\[ = \max_{D_1} \{ P(R_1 \leq \alpha \mid X_1, D_1) \times \sum_{X_{i-1}} [P(X_{i-1} \mid R_1 \leq \alpha, X_1, D_1) \times \max_{D_{i-1}, \ldots, D_1} P(R_{i-1}, \ldots, R_1 \leq \alpha \mid X_{i-1}, D_{i-1}, \ldots, D_1)] \} \]

\[ = \max_{D_1} \{ P(R_1 \leq \alpha \mid X_1, D_1) \times \sum_{X_{i-1}} [P(X_{i-1} \mid R_1 \leq \alpha, X_1, D_1) \times f_{i-1}(X_{i-1}, \alpha)] \} \]

\[ \ldots (5.1) \]

where \( \sum_{X_{i-1}} \) means summing through all the possible states \( X_{i-1} \), resulting from the random inflow \( Q_i \) and the decision \( D_1 \) to aim for \( Y_{i-1} \) at the end of stage \( i \)

\[ \max_{D_1} \] means to seek an optimal decision \( D_1^* \) among all the possible decisions \( D_1 \) at stage \( i \)

The random inflow \( Q_i \), according to the forecast may have value \( Q_{i,j} \), where \( j = 1, 2, \ldots, m \) (\( Q_{i,j-1} < Q_{i,j} \)), with associated probability \( P(Q_{i,j}) \). If at the beginning of stage \( i \), \( X_i \) is known and a decision \( D_i \) is made to operate the reservoir on that day such that the aim is to have the water level in the reservoir at the end of stage \( i \) at \( Y_{i-1} \), then three types of decision \( D_i \) are possible:
Type (1): decision $D_1$ such that $Y_{i-1} < X_i$, i.e. the decision is to lower the water level in the reservoir no matter what the random inflow $Q_i$ is. Such a decision is always attainable, i.e. $X_{i-1} = Y_{i-1}$ is always true, and therefore

$$\text{Prob}(X_{i-1} = Y_{i-1} / R_i < a, X_i, D_1) = 1.0$$

$$\text{Prob}(X_{i-1} = Y_{i-1} / R_i < a, X_i, D_1) = 0.0$$

Hence Eq.5.1 becomes

$$f_1(X_i, a) = \max \{P(R_i < a / X_i, D_1) \ast f_{i-1}(X_{i-1}, a)\} \ldots (5.2)$$

Type (2): decision $D_1$ such that $Y_{i-1} = X_i$, i.e. the decision is to maintain the water level in reservoir no matter what the random inflow $Q_i$ is. Since such a decision is also always attainable, $X_{i-1} = Y_{i-1}$ is always true, and therefore Eq.5.1 becomes Eq.5.2.

Type (3): decision $D_1$ such that $Y_{i-1} > X_i$, i.e. the decision is to store more water up to the level $Y_{i-1}$. There are three possibilities associated with such a decision, depending on the actual inflow value of the random variable $Q_i$

(3a): $Q_{i,1} > (Y_{i-1} - X_i)$ i.e. the minimum possible value of inflow $Q_i$ is sufficient to fill up the volume; then such a decision is always attainable, $X_{i-1} = Y_{i-1}$ is always true and Eq.5.1 becomes Eq. 5.2.

(3b): $Q_{i,1} < (Y_{i-1} - X_i) < Q_{i,m}$, i.e. the decision aimed at level $Y_{i-1}$ may only be attainable if inflow exceeds a certain value $Q_{i,k}$, $1 < k < m$
An example of such a case is shown in Fig. 5.5.

\[ k=4, \ m=11, \ \text{namely the random inflow} \ Q_i \ \text{may take any value} \ Q_{i,j}, \ j=1,2,\ldots,11. \]

If \( Q_i = Q_{i,1} \), then instead of reaching the aimed level \( Y_{i-1} \), at the end of stage \( i \), it will be at state \( t \). Therefore \( Y_{i-1} = X_{i-1} = t \).

In short, \( X_{i-1} \) may be any one of the 4 possible states, \( t, u, v, \) and \( Y_{i-1} \). Hence the summation consists of 4 terms: \( P(X_{i-1} = t/R_i < \alpha, \ X_i, \ D_i) = P(Q_{i,1}), \ P(X_{i-1} = u/R_i < \alpha, \ X_i, \ D_i) = P(Q_{i,2}), \ P(X_{i-1} = v/R_i < \alpha, \ X_i, \ D_i) = P(Q_{i,3}), \) and \( P(X_{i-1} = Y_{i-1}/R_i < \alpha, \ X_i, \ D_i) = \sum_{j=4}^{11} P(Q_{i,j}) \)

If when \( Q_i = Q_{i,8} \), the damage done will be \( \alpha \% \) of the maximum possible damage, then

\[ P(R_i < \alpha / X_i, \ D_i) = \sum_{j=1}^{8} P(Q_{i,j}) \]

(3c): \( Y_{i-1} - X_i > Q_{i,m} \), i.e. the maximum possible value of inflow \( Q_i \) is still not sufficient to fill up the volume, then the decision aimed
at level $Y_{i-1}$ is not attainable. Therefore $P(X_{i-1} = Y_{i-1} | R_{i-1} < \alpha, X_i, D_i) = 0.0$ and no damage will be possible. Hence $P(R_{i-1} < \alpha | X_i, D_i) = 1.0$.

Reference 15 gives good graphical illustrations of similar stochastic processes.

5.5 A digital computer programme has been written (appendix 1) for the computation of Eq.5.1 with the initial condition that the reservoir is empty and the final condition that the reservoir is full.

Typical final output is shown in Fig.5.6. Typical final output is shown in Fig.5.6, plotted according to optimal solutions tabulated in Table 5.1.

### Table 5.1

<table>
<thead>
<tr>
<th>Percentage of damage $\alpha$ (Max. Prob.)</th>
<th>0</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Decision: % of reservoir to fill)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.10</td>
<td>0.57</td>
<td>0.94</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
</tbody>
</table>

The result shows that according to the forecast future inflows the probability of having damage not greater than 30% of the maximum possible damage level is practically zero. However, there is 0.1 probability of having damage not exceeding 40%, if the reservoir is operated according to a certain policy, which involves aiming for the reservoir being 8% full the very first day. There is 0.57 probability of having damage not exceeding 50% of the maximum possible damage, if the reservoir is operated according to another policy i.e. aiming for the reservoir being 34% full the very first day. This in fact shows that the result of dynamic programming to determine optimal reservoir
Fig. 5.6

Maximum probability of being able to contain flood damage less than α% of the maximum possible damage.

Percent of maximum possible damage
operation policy by a probability approach offers a set of policies, each of which is optimized. It also provides a basis to evaluate these policies.

5.6 Recursive type of damage: expected value approach.

The stochastic recursive formula developed in section 4.4 is most readily applicable in this case to find an optimal reservoir operating strategy such that the total expected damage for the entire flood season is minimized.

\[
\bar{f}_i(X_i) = \min_{D_i} \left\{ \sum_{Q_i} P(Q_i) \left( R_i(Q_i/X_i, D_i) + \bar{f}_{i-1}(X_{i-1}) \right) \right\}
\]

\[
= \min_{D_i} \left\{ \sum_{Q_i} P(Q_i) \left( R_i(Q_i/X_i, D_i) \right) + \sum_{X_{i-1}} P(X_{i-1}/X_i, D_i) \cdot \bar{f}_{i-1}(X_{i-1}) \right\}
\]

where \(\bar{f}_i(X_i)\) is the minimum expected damage over all periods starting from stage \(i\) to the end of the flood season, namely stage 1.

\(R(Q_i/X_i, D_i)\) is the damage done at stage \(i\) only, if the random inflow \(Q_i\) takes on a particular value and given that at the beginning of stage \(i\) the water level is at \(X_i\) and a decision \(D_i\) is made.

\(P(Q_i)\) is the probability of \(Q_i\) taking a particular value, thus causing the corresponding amount of damage.

\(P(X_{i-1}/X_i, D_i)\) is the probability of reaching state \(X_{i-1}\) at the end of stage \(i\), given that the water level at beginning of stage \(i\) is at \(X_i\) and a decision \(D_i\) is made.

A computer programme has also been written (appendix 2) for this optimization. Given the forecasts, as outlined in Chapter 3, the possible decisions on the very first day and the probable results from each decision are shown on table 5-2.
Table 5-2

Possible decisions: Minimized expected 
reservoir percentage damage from 2nd day 
to the end of the flood season. 
Minimized expected Total 
damage done in the minimized damage for the whole 
1st day. expected season.

<table>
<thead>
<tr>
<th></th>
<th>Minimized expected</th>
<th>Minimized expected</th>
<th>Total minimized</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>damage from 2nd day</td>
<td>damage done in the</td>
<td>expected damage</td>
</tr>
<tr>
<td></td>
<td>to the end of the</td>
<td>1st day.</td>
<td>for the whole</td>
</tr>
<tr>
<td></td>
<td>flood season.</td>
<td></td>
<td>season.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>135.2</td>
<td>1.2</td>
<td>137.4</td>
</tr>
<tr>
<td>4</td>
<td>135.5</td>
<td>0.5</td>
<td>136.0</td>
</tr>
<tr>
<td>6</td>
<td>135.9</td>
<td>0.1</td>
<td>136.0</td>
</tr>
<tr>
<td>8</td>
<td>136.2</td>
<td>0.0</td>
<td>136.2</td>
</tr>
<tr>
<td>10</td>
<td>136.5</td>
<td>0.0</td>
<td>136.5</td>
</tr>
<tr>
<td>12</td>
<td>136.9</td>
<td>0.0</td>
<td>136.9</td>
</tr>
<tr>
<td>14</td>
<td>137.3</td>
<td>0.0</td>
<td>137.3</td>
</tr>
<tr>
<td>16</td>
<td>137.7</td>
<td>0.0</td>
<td>137.7</td>
</tr>
<tr>
<td>18</td>
<td>137.9</td>
<td>0.0</td>
<td>137.9</td>
</tr>
<tr>
<td>20</td>
<td>138.0</td>
<td>0.0</td>
<td>138.0</td>
</tr>
<tr>
<td>22</td>
<td>138.2</td>
<td>0.0</td>
<td>138.2</td>
</tr>
<tr>
<td>24</td>
<td>138.4</td>
<td>0.0</td>
<td>138.4</td>
</tr>
<tr>
<td>26</td>
<td>138.6</td>
<td>0.0</td>
<td>138.6</td>
</tr>
<tr>
<td>28</td>
<td>138.9</td>
<td>0.0</td>
<td>138.9</td>
</tr>
<tr>
<td>30</td>
<td>139.1</td>
<td>0.0</td>
<td>139.1</td>
</tr>
<tr>
<td>32</td>
<td>139.1</td>
<td>0.0</td>
<td>139.1</td>
</tr>
<tr>
<td>34</td>
<td>139.1</td>
<td>0.0</td>
<td>139.1</td>
</tr>
<tr>
<td>36</td>
<td>139.1</td>
<td>0.0</td>
<td>139.1</td>
</tr>
<tr>
<td>38</td>
<td>139.2</td>
<td>0.0</td>
<td>139.2</td>
</tr>
<tr>
<td>40</td>
<td>139.2</td>
<td>0.0</td>
<td>139.2</td>
</tr>
<tr>
<td>42</td>
<td>139.3</td>
<td>0.0</td>
<td>139.3</td>
</tr>
<tr>
<td>44</td>
<td>137.4</td>
<td>0.0</td>
<td>137.4</td>
</tr>
<tr>
<td>46</td>
<td>137.7</td>
<td>0.0</td>
<td>137.7</td>
</tr>
<tr>
<td>48</td>
<td>137.9</td>
<td>0.0</td>
<td>137.9</td>
</tr>
<tr>
<td>50</td>
<td>138.2</td>
<td>0.0</td>
<td>138.2</td>
</tr>
<tr>
<td>52</td>
<td>138.6</td>
<td></td>
<td>138.6</td>
</tr>
</tbody>
</table>
From the above table 5-2 it can be seen that the optimal operating strategy for the entire flood season requires that for the first day, the aim should be to let the reservoir fill to 4% full and this will give a minimum total expected seasonal damage of 136 units of value. Although the expected value approach is statistically meaningful, the disadvantage of finding an optimal operating strategy based on the minimization of the expected damage is that no indication is given regarding the risk involved in this optimal strategy, or the possibility of actually achieving this minimum total expected damage, when the optimal strategy is carried out.

5.7 Recursive type of damage: suggested probability approach.

An alternative method to determine the optimal operating policy with a damage function of the recursive type would be to follow a probability approach, similar to that described in section 5.4 but modify the recursion equations as follows:

State variable $X_i$: system condition at the beginning of stage $i$

Decision variable $D_i$: decision made at stage $i$

State variable $X_{i-1}$: system condition at the end of stage $i$

Return $R_i$ : damage suffered at stage $i$ only

$a$ : total damage suffered from stage $i$ to stage 1, the end of the flood season

$$
\text{Prob}(R_i + R_{i-1} + \ldots + R_1 < a/X_i, D_i, D_{i-1}, \ldots, D_1) \\
= \sum_{\beta=0}^{a} \text{Prob}(R_i < \beta/X_i, D_i) \times \text{Prob}(R_{i-1} + R_{i-2} + \ldots + R_1 < a-\beta/X_i, D_i, \ldots, D_1) \\
= \sum_{\beta=0}^{a} \text{Prob}(R_i < \beta/X_i, D_i) \times \left[ \sum_{X_{i-1}} \text{Prob}(X_{i-1} | X_i, D_i) \times \text{Prob}(R_{i-1} + R_{i-2} + \ldots + R_1 < a-\beta | X_{i-1}, D_{i-1}, \ldots, D_1) \right]
$$
Let \( f_1(X_1, \alpha) \) be the optimum probability that total damages done from stage 1 till stage 1 are not greater than \( \alpha \), then

\[
f_1(X_1, \alpha) = \max_{D_1, D_{1-1}, \ldots, D_1} \text{Prob}(R_1 + R_{1-1} + \ldots + R_{1-i} < \alpha / X_1, D_1, D_{1-1}, \ldots, D_1)
\]

\[
= \max \{ \sum_{R_{1-1}} \text{Prob}(X_{1-1} / X_1, D_1) \times f_{i-1}(X_{i-1}, (\alpha - \beta)) \}
\]

5.8 Summary

In both the expected value and the probability approaches to finding optimal reservoir operation policies, the optimizations are all based upon the probability distributions of future river flows. In this study the assumptions of:

(1) Single purpose flood control reservoir, and
(2) Unlimited reservoir outlet capacity

were made to simplify the analysis of the main problem, which was to develop a policy for decision making under uncertainty. However, a real reservoir with a limited outlet capacity would merely add a constraint, in that the outflow from the reservoir could not exceed the discharge capacity. It is one of the most attractive features of dynamic programming that such constraints can be either linear or non-linear, and they will not limit the use of dynamic programming. In fact, constraints usually shorten the computing time since they reduce the number of available alternatives. For a multipurpose reservoir, the return from any stage, \( R_i \), will not consist of only one term as in this study, i.e. flood damage, but rather a composite of all the returns from irrigation, water supply, hydropower, flood control, etc. But again this can be readily handled in dynamic programming. In summary, the basic concepts and formulations developed in this study could be readily adapted to multiple
purpose reservoir with limited outlet capacity.

A reservoir daily operating policy based on analysis of the "critical period" of records is inadequate, since it is a deterministic model and the operation policy so formed is optimal only for that definite sequence of river flows. The advantage of using the output from a daily flow forecast system as the input to the optimization analysis is that the operation policy can be constantly revised to take into account the actual flows which occur and the most up to date forecasts. To clarify this point, let us imagine that on May 15th the reservoir is empty, the forecast system predicts the daily flows from May 16th to the end of the flood season, and with this information an optimal policy for the reservoir operation is determined. The reservoir operation on May 16th is carried out according to this optimal policy. When the flow on May 16th, a random variable originally, reveals itself by assuming a particular value and as a result of this flow and the decision executed on May 16th, the reservoir will change from being empty to a new state (either remain empty, or partly filled). From the new information available to the forecast system namely the actual value of the flow on May 16th instead of the originally random flow on that day, the forecast system can give predictions of future flows from May 17th to the end of the period. This prediction should be more accurate since more information is available to the forecast system. With the new set of predicted values and the new initial condition, i.e. the actual water level in the reservoir on May 16th, dynamic programming can again determine a new optimal daily operation strategy. Likewise the whole process can be repeated till the very end of the flood season. In this way, the reservoir daily operating strategy makes the best possible use of the information available on each day.
From the above analysis, it is seen that dynamic programming as an optimization technique is extremely powerful and readily applicable to Water Resources systems. However, meaningful results from any mathematical method basically rely upon the validity of input data. Thus a realistic forecast model and realistic utility functions such as flood damage, irrigation benefit, etc. are required before a dynamic programming approach such as those developed in this study can be very useful in the determination of reservoir operating policies.

Conclusions

Two different approaches to determining the optimal daily operating strategy for a reservoir have been developed in this study, and applied to the case of operating a single purpose flood control reservoir when the inflows can be forecast with some accuracy. Both approaches use dynamic programming, but one involves the minimization of the expected value of the "return" in this case flood damage, and the other involves maximizing the probability of occurrence of a desirable event, namely restricting the inevitable damage to a certain amount.

The probability approach offers a set of alternative optimal policies, each one based on maximizing the probability of restricting the damage to a certain selected figure, and it also indicates the probability of being able to achieve each of the selected objectives which give a basis for comparing and evaluating the alternative policies. Thus it gives more information to the decision maker than the expected value approach, which simply offers one policy to minimize the expected value of damage but gives no indication of the risks involved.

Given a daily forecast system, and a realistic utility function for flood damage, an approach such as one of those developed in this study has the
advantage over the traditional rule curve approach (which is usually based on the critical period of record), that the policy so formed is optimal given the information available to that time, and can be continuously revised as new information becomes available.
References


References - cont:


(16) M. Abramowitz and I.A. Stegun (edited), "Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables", Dover, 1965.
APPENDIX 1

COMPUTER PROGRAMME FOR THE PROBABILISTIC APPROACH TO NON-RECURSIVE TYPE OF FLOOD DAMAGE
$SIG TSOU

**LAST SIGNON WAS: 12:50:33 08-14-70
USER "TSOU" SIGNED ON AT 12:50:41 ON 08-14-70
$LIST TAX(1,155)

1 DIMENSION QP(42), R(42), ISTATE(51), D(42,51,11)
2 COMMON QMID(60), P(60), F(43,51,11)
3 INTEGER D, Q, DQ, QMAX, QMID
4 READ(5,101) QP
5 101 FORMAT(13F6.0)
6 READ(5,102) R
7 102 FORMAT(13F6.3)
7.1 III=0
8 DO 1 I = 1, 42
9 DO 1 J = 1, 51
10 DO 1 K = 1, 11
11 D(I,J,K) = 0
12 DO 2 I = 1, 43
13 DO 2 J = 1, 51
14 DO 2 K = 1, 11
15 F(I, J, K) = -1.0
15.1 A = -1.0
16 DO 3 K = 1, 11
17 F(43,51,K) = 1.0
18 DO 18 I = 1, 51
19 18 ISTATE(I) = I
20 L = 41
21 DO 4 I = 1, 42
22 II = L + I
23 SEGMA = 52.0 * SQRT( 1.0 - R(II) ** 2 )
24 CALL FLOW(QP(II), SEGMA, IMAX)
25 DO 6 INT = 1, 51
26 DO 7 IP = 1, 11
27 QMAX = 200 + 30 * ( IP - 1 )
28 DO 8 INTPRE = 1, 51
29 IF ( F(II+1, INTPRE, IP) .EQ. A ) GO TO 8
30 IF ( INTPRE - INT ) .LT. 9, 10, 11
31 DQ = ( INT - INTPRE ) * 10
32 Q = DQ + QMID(1)
33 IF ( Q .GT. QMAX ) GO TO 12
34 14 SUM = 0.0
35 15 IQ = 1, IMAX
35.1 Q = DQ + QMID(IQ)
35.2 IF ( Q .GT. QMAX .AND. IP .NE. 11 ) GO TO 16
35.3 15 SUM = SUM + P(IQ)
35.4 16 FTEMP = SUM * F(II+1, INTPRE, IP)
36 GO TO 13
37 12 FTEMP = 0.0
38 GO TO 13
39 10 IF ( QMID(1) .GT. QMAX ) GO TO 12
40 DQ = 0
41 GO TO 14
45 11 DQ = ( INT - INTPRE ) * 10
45.1 IDQ = ( INTPRE - INT ) * 10
46 IF ( QMID(IMAX) .LT. IDQ ) GO TO 7
55 CALL FLOOD(DQ, QMAX, IMAX, IP, FTEMP, INT, INTPRE, II)
56 13 IF ( FTEMP .LT. F(II, INT, IP) ) GO TO 8
57  F(I1, INT, IP) = FTEMP
58  D(I1, INT, IP) = INTPRE
59  8 CONTINUE
60  7 CONTINUE
61  IF (II .EQ. 1) GO TO 17
62  6 CONTINUE
63  WRITE(6,201) II
64  201 FORMAT(1HI, 'STAGE=', 12)
65  DO 5 IP = 1, 11, 2
66  5 IP = IP - 1
67  WRITE(6,202) IP, (ISTATE(J), J=1, 17), (D(I1, J, IP), J=1, 17),
68  L(I1, J, IP), J=1, 17
69  202 FORMAT(1X, 12, 3X, 'STATE', 17I6, /, 10X, 'DECISION', 17(4X, I2),
70  1/, 10X, 'OPT. PRO.', 17F6.2, /)
71  WRITE(6,203) (ISTATE(J), J=18, 34), (D(I1, J, IP), J=18, 34),
72  1(F(I1, J, IP), J=18, 34)
73  203 FORMAT(13X, 'STATE', 17I6, /, 10X, 'DECISION', 17(4X, I2), /, 10X, 'OPT. PRO.
74  1', 17F6.2, /)
75  WRITE(6,204) II, (ISTATE(J), J=35, 51), (D(I1, J, IP), J=35, 51),
76  1(F(I1, J, IP), J=35, 51)
77  4 L = L - 2
78  WRITE(6,204) II, (ISTATE(J), J=1, 10), (D(1, 1, K), K=1, 11),
79  204 FORMAT(1HI, /, 'STAGE=1', 5X, 'INITIAL CONDITION: RESERVOIR EMPTY', 5
80  1X, 'FINAL CONDITION: RESERVOIR FULL', /, 12X, 'PERCENTAGE DAMAGE', 1
81  2(X5, I2, '0'), /, 21X, 'DECISION', 6I6, 'MAXIMUM PROBABILITY', 11
82  3B11F8.2)
83  STOP
84  END
85  SUBROUTINE FLOOD(DQ, QMAX, IMAX, IP, FTEMP, INT, INTPRE, II)
86  COMMON QMID(60), P(60), F(43, 51, 11)
87  INTEGER DQ, QMAX, QMID, Q
88  SUM = 0.0
89  SUMPRE = 0.0
90  A = -1.0
91  Q = DQ + QMID(1)
92  IF (Q .LE. 0) GO TO 5
93  1 DO 4 IQ = 1, IMAX
94  Q = DQ + QMID(IQ)
95  IF (Q .GT. QMAX .AND. IP .NE. 11) GO TO 9
96  4 CONTINUE
97  SUM = SUM + P(IQ)
98  12 SUM = SUM + P(IQ)
99  5 IF (Q .GT. QMAX .AND. IP .NE. 11) GO TO 6
100  SUMPRE = SUMPRE + P(IQ) * F(I1+1, J, IP)
101  SUM = SUM + P(IQ)
102  102 4 CONTINUE
103  FTEMP = SUM * SUMPRE
104  103.1 IF (II .EQ. 42) FTEMP = SUM
105  GO TO 7
106  2 DO 8 IQ = 1, IMAX
107  Q = DQ + QMID(IQ)
108  IF (Q .GT. QMAX .AND. IP .NE. 11) GO TO 9
109  SUM = SUM + P(IQ)
110  9 FTEMP = SUM * F(I1+1, INTPRE, IP)
111  108.2 GO TO 7
112  3 Q = DQ + QMID(1)
113  108.31 IF (Q .GT. QMAX .AND. IP .NE. 11) GO TO 13
114  108.3
GO TO 3
146  2 K = 1
147  3 L = J
148  4 P(L) = P(K)
149  5 M = J - 1
150  6 P(L) = PTEMP
151  7 QMID(L) = QTEMP
152  8 QMID(M) = 2 * QMID(J) - QMID(L)
153  9 P(M) = P(L)
154 10 L = L - 1
155 11 IMAX = JJ
156 12 RETURN
157 13 END
158 14 FUNCTION HAST(T)
159 15 X = 1.0 - 0.39894 * EXP(-0.5*T**2) * (0.43618*X - 0.12016*X**2 + 0.93729*X**3)
160 16 RETURN
161 17 END
162 18 SUBROUTINE FLOW(QP,SEGMA,IP)
163 19 COMMON QMID(60), P(60), F(43,51,11)
164 20 INTEGER QMID, QTEMP
165 21 PTOTAL = 0.0
166 22 DO 1 I = 1, 30
167 23 A(I) = I
168 24 QMID(I) = QP + 10.0 * (AI - 1.0)
169 25 QDUMMY = QMID(I)
170 26 TU = (QDUMMY + 5.0 - QP) / SEGMA
171 27 TL = (QDUMMY - 5.0 - QP) / SEGMA
172 28 IF (I .NE. 1) GO TO 6
173 29 P(I) = (HAST(TU) - 0.5) * 2.0
174 30 GO TO 7
175 31 6 P(I) = HAST(TU) - HAST(TL)
176 32 PTOTAL = PTOTAL + 2.0 * P(I)
177 33 GO TO 8
178 34 7 PTOTAL = PTOTAL + P(I)
179 35 8 IF (PTOTAL .LT. 0.99) GO TO 1
180 36 QMID(I+1) = QMID(I) + 10
181 37 P(I+1) = 0.5 * (1.0 - PTOTAL)
182 38 JJ = 2 * (I + 1) - 1
183 39 J = JJ / 2 + 1
184 40 PTEMP = P(I+1)
185 41 QTEMP = QMID(I+1)
186 42 GO TO 2
187 43 1 CONTINUE
188 44 2 K = 1
189 45 3 L = J
190 46 4 P(L) = P(K)
191 47 5 M = L + 1
192 48 6 P(L) = PTEMP
193 49 7 QMID(L) = QTEMP
194 50 8 QMID(M) = 2 * QMID(J) - QMID(L)
195 51 9 P(M) = P(L)
196 52 10 L = L - 1
197 53 11 IMAX = JJ
198 54 12 RETURN
199 55 END
APPENDIX 2

COMPUTER PROGRAMME FOR THE EXPECTED VALUE APPROACH TO
RECURSIVE TYPE OF FLOOD DAMAGE
****** PLEASE RETURN TO COMPUTING CENTRE STAFF **********

$SIG TSOU

**LAST SIGNON WAS: 16:26:35 08-13-70

USER "TSOU" SIGNED ON AT 12:49:39 ON 08-14-70

$LIST ST0(1,230)

1 DIMENSION QMID(60), P(60), QP(42), R(42), F(43,51), D(42,51)

2 1, NO(42,51), ISTATE(51)

3 INTEGER D

4 QO = 200.0

5 QU = 400.0

6 DOOM = 39.0

7 DO 30 J = 1, 42

8 DO 30 J = 1, 51

9 D(I,J) = 0

10 30 NO(I,J) = 0

10.02 602 FORMAT(1X,'ISTATE=',12)

11 DO 31 I = 1, 51

12 31 ISTATE(I) = I

13 READ(5,101) QP

14 101 FORMAT (13F6.0)

15 READ(5,102) R

16 102 FORMAT (13F6.3)

17 DO 10 I = 1, 43

18 DO 10 J = 1, 51

19 10 F(I,J) = 10.0 ** 10

20 F(43,51) = 0.0

21 L = 41

22 DO 6 I = 1, 42

23 II = I + L

24 IF (II .NE. 1) GO TO 28

25 WRITE(6,201) II

26 201 FORMAT(1X,'STAGE=',12,/,10X,'POS. DEC.',5X,'STAGE*2 OPT.',5X,'EXP. R

27 LET.',5X,'EXP. TOTAL')

28 SEGMA = 52.0 * SQRT(1.0 - R(I)**2)

29 CALL FLOWQP(I), SEGMA, P, QMID, IMAX)

30 DO 11 INT = 1, 51

31 DO 12 INTPRE = 1, 51

32 IF (F(I+1,INTPRE) .EQ. 10.0**10) GO TO 12

33 SUM = 0.0

34 SP = 0.0

35 DUMMY = 0.0

36 IF (INTPRE - INT) 13, 14, 15

37 13 PTEMP = 1.0

38 DQ = (INT - INTPRE) * 10

39 IF (DQ .GT. QQ .AND. DQ .LT. QU) GO TO 16

40 IF (DQ .LE. QQ) GO TO 36

41 DEFRET = DOOM

42 GO TO 35

43 36 DO 17 IQ = 1, IMAX

44 Q = DQ + QMID(IQ)

45 IF (Q .LE. QQ) GO TO 17

46 IF (Q .GT. QQ .AND. Q .LT. QU) GO TO 24

47 SUM = SUM + P(IQ) * DOOM

48 GO TO 17

49 24 CALL FLOOD(Q,CAM)

50 SUM = DAM * P(IQ)

51 17 CONTINUE
FTEMP = SUM + F(I1+1,INTPRE)
DTEMP = SUM
IF ( I1 .NE. 1 ) GO TO 18
WRITE(6,204) INTPRE,F(2,INTPRE),DTEMP,FTEMP
 204 FORMAT(10X,18,5X,F12.1,5X,F8.1,5X,F9.1)
GO TO 18

16 CALL FLOOD(DQ,DEFRET)
35 DO 19. IQ = 1, IMAX
  19 IF ( QMID(IQ) .LE. QO ) GO TO 19
  32 CALL FLOOD(QMID(IQ),DAM)
 33 SUM = SUM + DAM * P(IQ)
19 CONTINUE
 14 PTEMP = 1.0
20 DO 21 IQ = 1, IMAX
  22 SP = SP + P(IQ)
  23 CALL FLOOD(Q,DAM)
  24 SUM = SUM + SP * ( DAM + F(I1+1,INTPRE) )
  25 DUMMY = DUMMY + DAM * P(IQ)
21 CONTINUE
  15 DQ = ( INTPRE - INT ) * 10
  16 CALL FLOOD(DQ,DEFRET)
  17 DO 19. IQ = 1, IMAX
  18 IF ( QMID(IQ) .LE. QO ) GO TO 19
  20. IF ( QMID(IQ) .LT. QU ) GO TO 20
  21 SUM = SUM + P(IQ) * DOOM
  22 GO TO 19
  23 CALL FLOOD(QMID,IQ),DAM)
  24 SUM = SUM + DAM * P(IQ)
  25 GO TO 19

14 PTEMP = 1.0
15 DQ = ( INTPRE - INT ) * 10
16 CALL FLOOD(DQ,DEFRET)
17 DO 19. IQ = 1, IMAX
18 IF ( QMID(IQ) .LE. QO ) GO TO 19
19 IF ( QMID(IQ) .LT. QU ) GO TO 33
20 SUM = SUM + P(IQ) * DOOM
21 GO TO 19
22 SP = SP + P(IQ)
23 CALL FLOOD(Q,DAM)
24 SUM = SUM + SP * ( DAM + F(I1+1,INTPRE) )
25 DUMMY = DUMMY + DAM * P(IQ)
21 CONTINUE
22 SP = SP + P(IQ)
23 CALL FLOOD(Q,DAM)
24 SUM = SUM + SP * ( DAM + F(I1+1,INTPRE) )
25 DUMMY = DUMMY + DAM * P(IQ)
21 CONTINUE
22 SP = SP + P(IQ)
23 CALL FLOOD(Q,DAM)
24 SUM = SUM + SP * ( DAM + F(I1+1,INTPRE) )
25 DUMMY = DUMMY + DAM * P(IQ)
21 CONTINUE
22 SP = SP + P(IQ)
23 CALL FLOOD(Q,DAM)
24 SUM = SUM + SP * ( DAM + F(I1+1,INTPRE) )
25 DUMMY = DUMMY + DAM * P(IQ)
112  18 IF ( FTEMP - F(I,INT) ) 26, 27, 12
113  26 F(I,INT) = FTEMP
114  27 D(I,INT) = INTPRE
115  12 PDEC = PTEMP
116  25 DAMEXP = DTEMP
117  11 NO(I,INT) = 1
118  20 GO TO 12
119  27 NO(I,INT) = NO(I,INT) + 1
120  11 CONTINUE
121  25 IF ( II .EQ. 1 ) GO TO 29
122  11 CONTINUE
123  20 WRITE(6,202) II,(ISTATE(J),J=1,25),(D(I, J),J=1,25),(NO(I, J),
124  1J=1,25)
125  202 FORMAT(1X,'STAGE=',12,'/10X,'STATE',3X,25I4,'/10X,'DICISION',25I4,
126  1/10X,'CHOICE',2X,25I4)
127  206 WRITE(6,206) (ISTATE(J),J=26,51),(D(I, J),J=26,51),(NO(I, J),J=26,
128  151).
129  206 FORMAT(10X,'STATE',3X,25I4,'/10X,'DICISION',25I4,'/10X,'CHOICE',2X
130  1,25I4)
131  6 L = L - 2
132  29 K = ( D(I,1) - 1 ) * 2
133  25 WRITE(6,205) (QMI.D(I),I=1,IMAX),(P(I),I=1,IMAX),K,F(I,1)
134  205 FORMAT(//,'INFLOW DISTRIBUTION',//,20X,9F10.1,//,20X,9F10.3,//,10X,
135  1'THE BEST STRATEGY IS TO ALLOW RESERVOIR',13,'% FULL. THIS GIVES
136  2THE MINIMUM EXPECTED DAMAGE OF',F10.1)
137  1 STOP
138  1 END
139  1 FUNCTION HAST(T)
140  1 X = 1.0 / (1.0 + 0.33267 * T )
141  1 HAST = 1.0 - 0.39894 * EXP(-0.5*T**2) * (0.43618*X - 0.12016*X**2
142  1+ 0.93729*X**3)
143  1 RETURN
144  1 END
145  1 SUBROUTINE FLOOD(Q,DAM)
146  1 IQ = ( Q - 200.0 ) / 10.0
147  1 GO TO (51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,
148  170), IQ
149  51 DAM = 1.0
150  51 GO TO 9
151  52 DAM = 2.1
152  52 GO TO 9
153  53 DAM = 3.3
154  53 GO TO 9
155  54 DAM = 4.6
156  54 GO TO 9
157  55 DAM = 6.0
158  55 GO TO 9
159  56 DAM = 7.5
160  56 GO TO 9
161  57 DAM = 9.1
162  57 GO TO 9
163  58 DAM = 10.8
164  58 GO TO 9
165  59 DAM = 12.6
166  59 GO TO 9
167  60 DAM = 14.5
168  60 GO TO 9
169  61 DAM = 16.5
170  61 GO TO 9
171  62 DAM = 18.6
SUBROUTINE FLOW(QP,SEGMA,P,QMID,IMAX)

DIMENSION QMID(60), P(60)

PTOTAL = 0.0

DO 1 I = 1, 30

AI = I

QMID(I) = QP + 10.0 * ( AI - 1.0 )

TU = ( QMID(I) + 5.0 - QP ) / SEGMA

TL = ( QMID(I) - 5.0 - QP ) / SEGMA

P(I) = HAST(TU) - HAST(TL)

IF ( I .EQ. 1 ) GO TO 7

PTOTAL = PTOTAL + 2.0 * P(I)

GO TO 8

7 PTOTAL = PTOTAL + P(I)

IF ( PTOTAL .LT. 0.99 ) GO TO 1

Qomid(I+1) = QMID(I) + 10.0

P(I+1) = 0.5 * ( 1.0 - PTOTAL )

JJ = 2 * ( I + 1 ) - 1

J = JJ / 2 + 1

PTEMP = P(I+1)

QTEMP = QMID(I+1)

GO TO 2

1 CONTINUE

2 K = 1

L = J

4 P(L) = P(K)

QMID(L) = QMID(K)

IF ( L .EQ. JJ-1) GO TO 3

K = K + 1

L = L + 1

GO TO 4

N = J - 1

L = L + 1

P(L) = PTEMP

QMID(L) = QTEMP

DO: 5 M = 1, N

QMID(M) = 2.0 * QMID(J) - QMID(L)

5 L = L - 1

IMAX = JJ

RETURN

END