Chapter 5

Systems Analysis Techniques

The objectives of this chapter are:

- to introduce the systems analysis techniques;
- to explain the optimization techniques that are used in water resources;
- to describe stochastic optimization techniques with special reference to water resources; and
- to explain the simulation tool in detail.

The basic philosophy of an objective in planning or operation is the concept of change for the better; here 'better' is defined by person(s) or society having both the desire and the authority to take action. Two important characteristics of change are the direction the magnitude of desired change. Since the direction of change cannot be easily specified except saying that it should be towards improvement, the word objective is more or less exclusively used to measure the magnitude of the change. The magnitude can be indicated by either some cardinal or an ordinal measure although a cardinal measure is desirable from an analytical point of view. Further, the magnitude of change may be described by a single measure or it may require a set of measures. If there is more than one measure, the level of achievement of any one objective is usually dependent on the level of achievement of the remaining objectives. If there are a number of objectives, there may be some common indices. One can attempt to aggregate all the common indices under different heads till no further aggregation is possible. The objectives left at this stage are termed as non-commensurate objectives. This implies that these objectives cannot be further aggregated without ignoring features that are of concern to the decision makers.

5.1 SYSTEMS ANALYSIS TECHNIQUES

The development of systems analysis techniques can be traced to the seventeenth century when Newton developed the differential calculus method of optimization. The operations research techniques were developed during the second world war when the allied powers had to fight war on many fronts and they were required to deploy limited resources in the

best possible manner. Since the techniques were aimed at getting the best results from military operations, these were known as operations research techniques. Dantzig developed the Simplex method of linear programming in 1947. The work by Kuhn and Tucker in 1951 provided the impetus for development of non-linear programming techniques. In 1957, Bellman developed the principle of optimality which laid the foundation for enormous developments in dynamic programming techniques. The goal programming and multi-objective optimization saw growth and applications during the decade of the 1960's and onwards. A firsthand account of the history of linear programming has been provided by Dantzig and Thapa (1997).

The popular operations research techniques include optimization methods, simulation, queuing theory, network flow theory, and game theory. Among these, optimization and simulation are extensively used in water resources problems. Therefore, only these two techniques will be covered in this chapter.

5.2 OPTIMIZATION

In many engineering problems, there are a number of possible solutions. It is, therefore, required to evaluate each alternative solution and then choose the best from the point of view of interest, say economic or convenience. Optimization is the science of choosing the best amongst a number of possible alternatives. The driving force in the optimization models is the objective function (or functions in multi-objective optimization). The term optimal solution essentially refers to the best from the solution of the mathematical model under all assumptions and constraints, whether explicitly stated or implicitly included in the formulation. Clearly, the optimal solution indicated by the model may be far from the actual system's optimal solution. Dantzig and Thapa (1997) defined mathematical programming (or optimization theory) as "that branch of mathematics dealing with techniques for maximizing or minimizing an objective function subject to linear, non-linear, and integer constraints on the variables". The word programming should not be related to computers; here it means 'scheduling', the setting of an agenda, or creating a plan of activities (ReVelle et al., 1997).

Any "optimal" solution derived is clearly dependent on the assumptions and criteria and their associated uncertainties. Some of these uncertainties might be derived from the selection of model structure, parameters, scope, or focus. Others might be related to data, the optimization techniques used to solve the mathematical models and the inability to account for many non-quantitative and non-tangible considerations in the model.

An optimization problem can be stated as:

$$\text{Maximize (or Minimize)} \quad f(X) \tag{5.1}$$

$$\text{subject to} \quad g_j(X) \geq 0, \quad j = 1,2,\ldots,m \tag{5.2}$$

$$h_j(X) = 0, \quad j = m+1, m+2,\ldots,p \tag{5.3}$$

where X is a vector of n-variables which are known as decision variables, g(X) are the

inequality constraints, and h(X) are the equality constraints. To solve an optimization problem, the value of decision variables is systematically changed. The range over which a decision variable can be changed is known as its feasible range. The decision-maker evaluates the available alternatives on the basis of some prescribed criterion function. This criterion function, denoted by f(X), is known as the objective function. Its choice depends on the problem. While formulating the optimization problem, one should carefully decide the objective function and it should properly reflect the preference of the decision-maker. In the beginning of analysis, objectives are often unclear or loosely stated. Considerable efforts may be needed to clarify them. Typically, the objective function may represent benefits which are maximized or it may represent costs which are to be minimized.

The availability of resources is usually limited and is expressed with the help of constraints. Here, g and h are the inequality and equality constraints. These constraints restrict the range over which the decision variables can change and thus affect the optimum solution. The number of decision variables and the number of constraints depend on the problem. If the number of constraints is zero then the problem is known as the unconstrained optimization problem.

In most practical problems, the surfaces of objective functions have more than one peak or trough. The graph in Fig. 5.1 shows the variation of two objective functions with the decision variable for a problem which has only one decision variable. The objective function shown with solid lines (Z_1) has only one extreme point. If the second objective function shown by dotted lines (Z_2) is to be minimized, points A and B are known as local optimum because the value of the objective function is lowest only in the vicinity of these points. At point C, the value of the objective function is lowest among all the points and hence this point is termed as the global optimum. The value of the objective function at a local optimum is more than the global optimum in case of minimization problem and vice versa for the maximization problem. In problems where the objective function has this type of behavior, the solution algorithm may end up at a local optimum. In an optimization problem, the constraints force the solution to lie within a limited region which is known as the *feasible region.* It is to be noted that usually real-life problems have a large number of decision variables and constraints. Therefore, in such problems, it is helpful to understand the nature of the objective function and constraints.

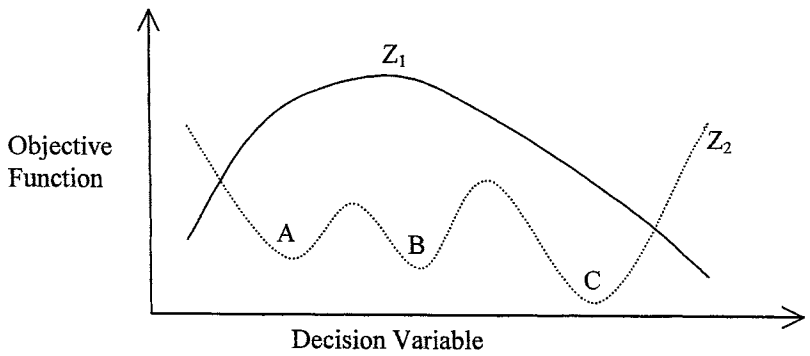

Fig. 5.1 Variation of two objective functions with decision variables.

5.2.1 Classification of Optimization Techniques

Optimization techniques are also known as *mathematical programming* techniques. They can be classified in several ways, such as on the basis of the existence of constraints, the nature of the problem, the nature of the equations involved, the permissible values of the design variables, the deterministic nature of the variables involved, the separability of the functions, the number of objective functions involved, etc. The usual way of classifying optimization techniques is based on the nature of the problem or equations involved. These techniques can be classified as Linear Programming (LP), Nonlinear Programming (NLP), Geometric Programming (GP), Dynamic Programming (DP), etc. This classification is useful from a computational point of view, since many methods have been developed solely for the efficient solution of a particular class of problems.

In the classification which depends on the existence of constraints, the problem can be classified as constrained optimization or unconstrained optimization. The input variables to a problem of water resources could be either deterministic or stochastic and depending upon that, the technique can be classified as deterministic optimization or stochastic optimization. According to Yeh and Becker (1982), stochastic optimization is useful for planning purposes, while deterministic optimization is a viable approach for real-time reservoir operation (see Section 11.7) with frequent updating of streamflow forecasts. Recently, many new optimization techniques have been used in studies dealing with water resources. Genetic Algorithm (GA) is one such approach whose use began in the 1970s (see Goldberg 1989; Dandy et al. 1996; Wardlaw and Sharif 1999). Another technique that has become popular in systems control is fuzzy programming (see Pedrycz 1993; Russell and Campbell 1996).

Although optimization encompasses a very wide range of subjects, keeping in view the current status of the application of optimization techniques in water resources, the discussion in this chapter is limited to LP, NLP, and DP only.

5.3 LINEAR PROGRAMMING

Optimization problems in which the objective function and constraints are linear functions of decision variables and the decision variables are non-negative are termed as linear programming (LP) problems. Dantzig and Thapa (1997) defined LP as a technique that "is concerned with maximization or minimization of a linear objective function in many variables subject to linear equality or inequality constraints." An optimization problem can be classified as an LP problem if it meets the following conditions:

- The decision variables of the problem are non-negative, i.e., positive or zero.
- The criterion function or objective function is described by a linear function of the decision variables, i.e., a mathematical function involving only the first powers of the variables with no cross products.
- The operating rules governing the processes, commonly known as constraints, are expressed as a set of linear equations or linear inequalities.

The LP type of optimization problem was first recognized in the 1930s by economists while developing methods for optimal allocation of resources. During the World War II, the United States Air Force sought more effective procedures to allocate resources and this led to the development of LP. G.B. Dantzig, who was a member of the Air Force Group, formulated the general LP problem and devised the simplex method of solution in 1947. This was a significant step in bringing LP into wider usage. Since then, LP models have been widely used to solve a variety of military, economic, industrial, social, engineering and hydrological problems. The number of applications of linear programming has grown immensely in the past few decades.

The LP models have been extensively used to solve water resources problems. Although the objective function and the constraints are not linearly related with the decision variables in many real-life water resources problems, these can be approximated by linear functions and the LP technique can be used to obtain the solution.

5.3.1 ASSUMPTIONS IN LP

Four basic assumptions are implicitly built into LP models:

a) Proportionality Assumption
The contribution of a decision variable to the objective function and its usage in various resource consuming activities is directly proportional to its values.

b) Additivity Assumption
This assumption indicates that the total usage of resources and contribution to the overall measure of effectiveness are equal to the sum of the corresponding quantities generated by each activity conducted by itself at the given level.

(c) Divisibility Assumption
According to this assumption, the fractional values of the decision variables are permissible.

(d) Deterministic Assumption
The parameters of the LP model are assumed to be known with certainty.

5.3.2 Mathematical Representation of an LP Problem

There are many ways to represent a linear programming problem. A general LP problem can be expressed in the conventional way. A compact way of representation of a linear program is a matrix form. Solutions are obtained by converting the general LP problem to a standard form. Each form is described in what follows.

Conventional Form

An LP problem consists of a linear objective function and a set of linear constraints. The constraints may be expressed in terms of inequalities or equalities. In most cases, especially in real-life water resources problems, constraints appear as inequalities. In many cases, it is

observed that a linear program may consist of both types of constraints, i.e., some constraints may be of equality type and some may be of inequality type.

A general LP problem can be written in conventional form as:

$$\text{Minimize (or maximize): } Z = f(x) \tag{5.4}$$

subject to:

$$g_j(x) \geq 0; \qquad j=1, 2, \ldots, m_1 \tag{5.5}$$
$$h_j(x) \leq 0; \qquad j=1, 2, \ldots, m_2 \tag{5.6}$$
$$l_j(x) = 0; j=1, 2, \ldots, m_3 \tag{5.7}$$

where Z is the objective function; x is an n-dimensional decision vector; $g_j(x)$ and $h_j(x)$ are inequality constraints; $l_j(x)$ are equality constraints; and m_1, m_2 and m_3 denote the number of constraints for these types, respectively. The objective function is a linear function of the decision variables.

Standard Form

An LP problem with m constraints and n variables can be represented in standard form as follows:

$$\text{Minimize (or maximize): } Z = c_1 x_1 + c_2 x_2 + ... + c_n x_n \tag{5.8}$$

subject to:

$$\begin{aligned} &a_{11} x_1 + a_{12} x_2 + ... + a_{1n} x_n = b_1 \\ &a_{21} x_1 + a_{22} x_2 + ... + a_{2n} x_n = b_2 \\ &\cdot \quad \cdot \quad .... \quad \cdot \quad \cdot \\ &\cdot \quad \cdot \quad .... \quad \cdot \quad \cdot \\ &\cdot \quad \cdot \quad .... \quad \cdot \quad \cdot \\ &a_{m1} x_1 + a_{m2} x_2 + ... + a_{mn} x_n = b_m \end{aligned} \tag{5.9}$$

$$x_i \geq 0, \; i = 1, 2, ... n. \tag{5.10}$$
$$b_i \geq 0, \; i = 1, 2, ... m. \tag{5.11}$$

where Z represents the objective function; x_i's are the decision variables and c_i's are the cost (or benefit) coefficients representing the cost (or benefit) incurred by increasing the x_i decision variable by one unit. The right-hand side of constraint equations represents the resource availability. These arise due to limited availability of a particular resource, say, water. The a_{ij} coefficients are called technological coefficients and quantify the amount of a particular resource i required per unit of the activity j.

The standard form of an LP problem is solved algebraically. The main features of the standard form are:

- The objective function is either of the maximization or minimization type.

- All the constraints are expressed as equations, i.e., equality type constraints, except the non-negativity constraints associated with the decision variables.
- All the decision variables are restricted to be nonnegative.
- The RHS constant of each constraint is nonnegative.

Matrix Form

In matrix notation, the standard LP problem can be expressed in a compact form as:

$$\text{Minimize (or maximize)} \quad Z = C^T X \tag{5.12}$$

subject to:

$$A X = b \tag{5.13}$$

$$X \geq 0 \tag{5.14}$$

$$b \geq 0 \tag{5.15}$$

where A is a $m*n$ matrix, X is an $n*1$ column vector, b is an $m*1$ column vector, C is an $n*1$ column vector, and Z represents the objective function. Superscript T in eq. (5.12) refers to the transpose operation. Thus, one can write:

$$X = [x_1, x_2, \ldots\ldots, x_n]^T$$
$$C = [c_1, c_2, \ldots\ldots, c_n]^T$$
$$b = [b_1, b_2, \ldots\ldots, b_m]^T$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}$$

5.3.3 Formulation of an LP Model

The three basic steps in constructing an LP model are:

- Identify the decision variables and represent them in terms of algebraic symbols.
- Identify all the restrictions or constraints in the problem and express them as equations or inequalities which are linear functions of the decision variables.
- Identify the objective or criterion which is to be either maximized or minimized, and represent it as a linear function of the decision variables.

These three basic steps will be clear when one formulates a number of linear programs. Here, one may note that the model building is not a science but is primarily an art and comes mainly by practice. Depending on the experience, skill and scientific knowledge about the system under consideration, the developed model will meet the realism and fulfill the intended objectives. Any discrepancy in the model formulation will yield an erroneous result, and sometimes may even give physically meaningless solution. Hence, it is necessary

to work out many exercises on problem formulation before handling a real-life problem.

5.3.4 Reduction of a General LP Problem to a Standard Form

The simplex method for solving an LP problem requires the problem to be expressed in the standard form. But not all LP problems appear in the standard form. In many cases, some of the constraints are expressed as inequalities rather than equations; at least it is most often true in case of water resources problems. In some problems, all the decision variables may not be even nonnegative. Hence, the first step in solving an LP problem is to convert it to the standard form. The procedure to convert a general program to the standard form is outlined below:

- Convert all inequalities to equalities.
- Convert all decision variables unrestricted in sign to strictly non-negative.
- Make all the right-hand side constants of the constraints nonnegative.

Handling Inequality Constraints

An inequality constraint of the type $\leq$ can be converted to the equality type by introducing a new nonnegative variable called a slack variable. This new variable is added to the left-hand side of the constraint. Hence, the constraint

$$a_{11} x_1 + a_{12} x_2 + ... + a_{1n} x_n \leq b_1 \tag{5.16}$$

can be written as:

$$a_{11} x_1 + a_{12} x_2 + ... + a_{1n} x_n + s_1 = b_1, \qquad s_1 \leq 0 \tag{5.17}$$

Here, s_1 is a slack variable.

Similarly, an inequality constraint of $\geq$ type can be converted to the equality type by introducing a new nonnegative variable called a surplus variable. This new variable is subtracted from the left-hand side of the constraint. Thus, the constraint

$$a_{11} x_1 + a_{12} x_2 + ... + a_{1n} x_n \geq b_1 \tag{5.18}$$

can be written as:

$$a_{11} x_1 + a_{12} x_2 + ... + a_{1n} x_n - s_2 = b_1 \qquad s_2 \geq 0 \tag{5.19}$$

Here s_2 is a surplus variable. Furthermore, a constraint of $\geq$ type, if desired, can be easily converted to one of $\leq$ type by multiplying by -1 throughout the equation.

Handling Variables Unrestricted in Sign

In some situations, it may become necessary to introduce a variable in the LP model that can have both positive and negative values. Since the standard form requires all the variables to be nonnegative, a variable unrestricted in sign must be transformed. The

unrestricted variable is replaced by the difference of the two nonnegative variables. If x_1 is unrestricted in sign, it can be replaced by $x_1 = x_2 - x_3$, where x_2 and x_3 can have only positive values.

Handling Constraints Having Negative Right-hand Side Constants

Since the right-hand side constant of each constraint must be nonnegative, the constraints having negative right-hand side constants are multiplied by -1 throughout to get the constraint in the standard form. Thus, the constraint

$$3x_1 - x_2 - 2x_3 = -5 \tag{5.20}$$

will take the form

$$-3x_1 + x_2 + 2x_3 = 5 \tag{5.21}$$

in standard form. It is important to note here that if the inequality type constraints are being multiplied by -1, their nature will reverse.

Interchanging the Nature of the Objective Function

The nature of the objective function, if desired, can be changed by putting a negative sign with the prescribed expression for the objective function. That means that a maximization problem is equivalent to a minimization problem with the negative of the objective function, i.e.,

$$\text{Max } [Z] = \text{Min } [-Z] \tag{5.22}$$

5.3.5 Canonical Form of an LP Problem

A system of equations which possesses at least one basic variable in all equations is called a canonical system. A variable is said to be a basic variable in a given equation if it appears with a unit coefficient in that equation and is absent in all other equations. A system of equations given by

$$x_1 - 3x_3 - 2x_4 - 4x_5 = 6 \tag{5.23}$$

$$x_2 - 2x_3 + x_4 - 3x_5 = 2 \tag{5.24}$$

represents a canonical system which has x_1 and x_2 as basic variables. This system is useful to obtain the optimal solution, and forms a basis for the simplex method.

To get a canonical system, a sequence of pivot operations is performed on the original system such that there is at least one basic variable in each equation. The number of basic variables is decided by the number of equations in the system. The variables which are not basic are called nonbasic variables. By applying the elementary row operations, a given variable can be made a basic variable.

The solution obtained from a canonical system by setting the nonbasic variables equal to zero and solving for the basic variables is called a basic solution. A basic feasible

solution is a solution in which the values of the basic variables are nonnegative. The basic feasible solution satisfies all constraints. A basic feasible solution which provides minimum (or maximum) value of the objective function is called an optimum solution. It may be noted that the feasible region of a properly formed LP problem is a convex set. A set is convex if it is not possible to find two points such that not all points on the line joining them belong to the set.

5.3.6 Graphical Solution of a LP Problem

The graphical method is a simple way to solve LP problems. This method is also very useful in conceptual understanding of the solution technique. However, it can be used to solve the LP problems involving at most two decision variables. In the following, an LP problem having two decision variables will be discussed. The feasible region and constraints are graphically shown in Fig. 5.2.

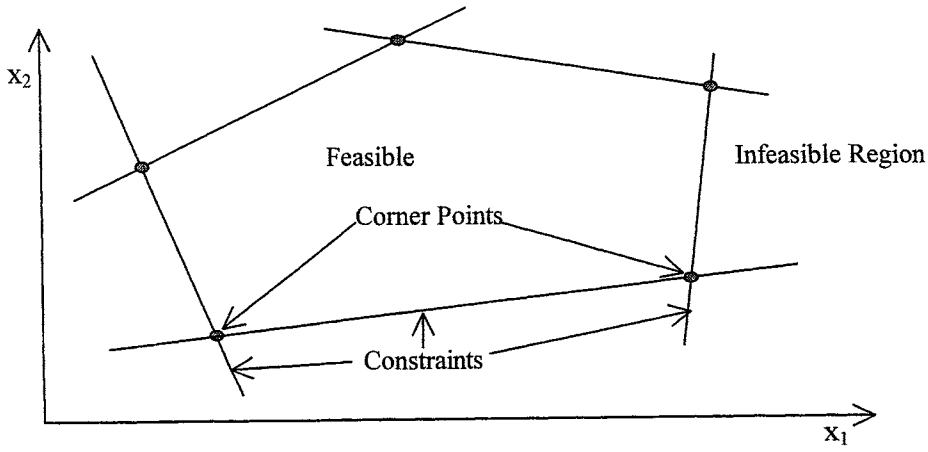

Fig. 5.2 Definition sketch of feasible region and constraints.

Example 5.1: Solve the following problem using graphical method of LP.

$$\text{Max } Z = 2x_1 + x_2 \tag{5.25}$$

subject to:

$$2x_1 - x_2 \leq 8 \tag{5.26}$$

$$x_1 + x_2 \leq 10 \tag{5.27}$$

$$x_2 \leq 7 \tag{5.28}$$

$$x_1, x_2 \leq 0 \tag{5.29}$$

Solution: In Fig. 5.3, the constraints are plotted against the coordinate axes x_1 and x_2. To plot the first constraint, $2x_1 - x_2 \leq 8$, plot a straight line $2x_1 - x_2 = 8$. Similarly, plot lines $x_1 + x_2 = 10$, and $x_2 = 7$ to mark the second and third constraints. The non-negativity constraints are plotted as the axes themselves. The feasible region can be easily delineated, and is

shown in Fig. 5.3 by the bounded pentagonal region formed by the lines of each constraint including non-negativity. The solution begins with an arbitrary value of the objective function, say 6 and the line $2x_1 + x_2 = 6$ is plotted. There are infinite points on the objective function line inside the feasible region and each of these points is a solution to the problem. Since it is a maximization problem, the objective function line is shifted to the right as far as possible while ensuring that at least one point lies in the feasible region. It can be seen that the farthest point up to which we can go is the point (6, 4). Beyond this point, although the value of the objective function increases, there is no feasible solution. Hence, this is the optimum solution of the problem with $x_1 = 6$ and $x_2 = 4$, and the optimal value of the objective function is 16.

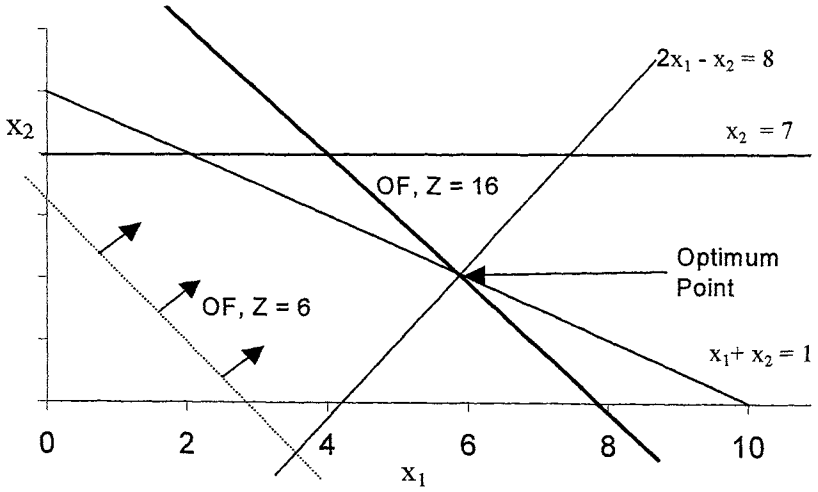

Fig. 5.3 Graphical Solution of Illustrative Example 5.1.

Importance of Corner Points

A closer inspection of Fig. 5.3 shows that the feasible region is compact and continuous and the optimum point is always a corner point. Further, if a constraint passing through this corner point is parallel to the objective function line, all the points falling on this constraint (or objective function) will have the same (optimum) value of the objective function. In this case, the problem will have infinite solutions. Such LP problems are said to have alternative or multiple optimal solutions. Some important properties of the corner points are:

- If there is only one optimal solution to an LP problem, it must be a feasible extreme point. If there are multiple optimal solutions, at least two must be adjacent feasible extreme points.
- In every problem, there are only a finite number of feasible extreme points.
- If a feasible extreme point is better than all its adjacent feasible points, then it is better than all other feasible extreme points. This property holds if the feasible region is convex. Based on this property, one need not enumerate all the extreme points, and

the status of one extreme point can be ascertained to determine whether the optimal solution has been reached or not.

In some problems the feasible region may not be a closed convex polygon, and hence it may be possible to increase the objective function value continuously and still be inside this region. Such types of problems are termed as having an unbounded solution. One may also notice in Fig. 5.3 that the constraint $x_2 \leq 10$ has no influence on the optimal solution. Such constraints are known as non-binding constraints. The constraints which are met with equality at the optimum point (e.g., $x_1 + x_2 \leq 10$) are called binding constraints.

5.3.7 Simplex Method of LP

Depending on the number of decision variables (n) and constraints (m), one of the three cases may arise in an LP problem: (i) $m = n$, (ii) $m > n$, and (iii) $m < n$. In the first case, the problem has a unique solution, if it exists, and there can be no optimization. If $m > n$, there would be $(m-n)$ redundant equations which could be eliminated. If $(m-n)$ equations are not redundant, the problem has a solution only in a least square sense. The case $m < n$ corresponds to an undetermined set of linear equations which, if exist, will have many solutions. The LP problem is to find one of these solutions which satisfies the constraints and yields the optimum value of the objective function. One can set $(n-m)$ variables equal to zero and solve the m equations for m variables. These solutions will be basic solutions as $(n-m)$ variables, which have been set equal to zero, represent non-basic variables. However, there will be ${}^{n}C_{m}$ such solutions.

If in a particular problem the number of decision variables, $n = 20$, and the number of constraints, $m = 10$, then the number of possible basic solutions will be ${}^{20}C_{10} = 20!/[(20-10)!10!] = 184756$. Hence, to solve this problem, one has to obtain 184756 solutions and compare them. This is a formidable task even with the help of a fast computer and hence a systematic and efficient method is necessary.

The simplex method, developed by Dantzig, is an efficient method to solve LP problems. It is an iterative procedure to solve problems which are in the standard form. The simplex method requires that the constraint equations be expressed as a canonical system from which a basic feasible solution can be readily obtained. Once a basic feasible solution is available, attempts are made to improve it untill the optimal solution is obtained.

The equations containing only the slack variables can be automatically considered as canonical system with the slack variables as basic variables. However, in many cases, finding a canonical system with a basic feasible solution is not an easy task. One way to obtain a basic feasible solution is to arbitrarily choose the basic variables and use a technique, such as Gaussian elimination, to get the solution. A systematic approach to get a canonical system starting from a basic feasible solution is the use of artificial variables. The artificial variables are added in those equations in which no basic variables appear by inspection. An auxiliary objective function is formed which is equal to the sum of artificial variables. This method is called two-phase simplex as there are two objective functions. The first phase aims at minimization of the auxiliary objective function. If, as a result of this

phase, this function cannot be made zero then the problem is infeasible and the algorithm is terminated. When the auxiliary objective function becomes zero, the optimization of the main function is taken up.

Computational Steps of the Simplex Method

The computational steps of the simplex method in tableau form are as follows:

1. Express the problem in standard form.
2. Start with an initial basic feasible solution in canonical form and set up the initial tableau.
3. Use the inner product rule to find the relative profit (or cost) coefficients ($\overline{C}_j$). This rule states that the relative profit (or cost) coefficient of a variable x_j ($\overline{C}_j$) is obtained by subtracting the product of the row matrix consisting of profit (or cost) coefficients (C_j) of basic variables and the column matrix consisting of transformation coefficients (a_{ij}) corresponding to x_j in the canonical system from the actual profit (or cost) coefficient corresponding to variable x_j. If all the relative profit coefficients are negative or zero, the current basic feasible solution is optimal for a maximization problem. For a minimization problem, all $\overline{C}_j$ should be positive or zero at the optimal point.
4. If the solution is not optimum, select the non-basic variable with the most positive $\overline{C}_j$ value (highest value) to enter the basis in a maximization problem. In a minimization problem, the non-basic variable with the most negative $\overline{C}_j$ value (lowest value) is selected to enter the basis. The decision is arbitrary in case of a tie. Let this variable be x_r. The value of the objective function can be also computed by multiplying the row matrix consisting of profit (or cost) coefficients of basic variables and the column matrix consisting of right-hand side constants.
5. Apply the minimum ratio rule to determine the basic variable to leave the basis. The minimum ratio rule is that for this variable (x_r), take b_i/a_{ir} ratio for each constraint row i (for those constraints only which have +ve a_{ir} values), and the minimum ratio determines the row in which the basic variable will have unit coefficient. The corresponding variable from this row (which was a basic variable) will leave the basis. The constraint row corresponding to the entering basic variable is known as pivot equation and the element located at the intersection of the entering column and pivoting row is known as the pivot element.
6. Perform the pivot operation to get the new tableau in canonical form, and get a new basic feasible solution.
7. Go to step 3 and repeat the steps until an optimal solution is found.

If, during the simplex iterations, the value of one or more basic variables becomes zero, it is termed as a degenerate solution. In such an event, there is no assurance that the solution will improve further. Sometimes, the degeneracy is temporary, and the solution improves after a few iterations.

The simplex method, when used for a large problem requires considerable computer time and storage. Some techniques have been developed which require lesser time

and storage. Among these techniques, the revised simplex method is the most popular in which the time and storage are saved by manipulating only selected entries of the simplex tableau. A new algorithm has been recently developed by Karmarkar (1984) which moves through the interior of the feasible region to attain the optimum.

Example 5.2: Solve the problem of Example 5.1 using the simplex method.

Solution: The problem is written in canonical form by introducing slack variables, x_3, x_4 and x_5:

$$\text{Max } Z = 2x_1 + x_2 \tag{5.30}$$

subject to:

$$2x_1 - x_2 + x_3 = 8 \tag{5.31}$$

$$x_1 + x_2 + x_4 = 10 \tag{5.32}$$

$$x_2 + x_5 = 7 \tag{5.33}$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0 \tag{5.34}$$

Here, the values of n and m are 5 and 3, respectively. Hence, there will be three basic variables which can be chosen arbitrarily. If the variables x_3, x_4 and x_5 are considered basic variables, the problem is in the canonical form. This finishes Step 1 and paves the way to Step 2. The initial basic feasible solution will be $x_1 = 0$, $x_2 = 0$, $x_3 = 8$, $x_4 = 10$ and $x_5 = 7$. The initial simplex tableau is formed as follows:

Tableau 1

C_j	2	1	0	0	0	
	x_1	x_2	x_3	x_4	x_5	RHS
	2	-1	1	0	0	8
	1	1	0	1	0	10
	0	1	0	0	1	7
$\overline{C_j}$	2	1	0	0	0	Z = 0

Here, one should note that the contents below the dotted line are not a part of the initial simplex tableau. The bottom row separated by the dotted line shows the value of the relative profit or cost coefficients corresponding to the variable x_j ($\overline{C}_j$). It is computed as stated in Step 3. Since two values of relative profit coefficients are not either negative or zero, this is not the optimal solution and hence this solution is to be improved. For this, we see that the value of the relative profit coefficient is maximum for x_1 variable, thus x_1 will enter the basis. This completes Step 4. Moving to Step 5, since row 1 gives the minimum (b_i/a_{ir}) ratio, the variable x_3 will leave the basis. Thus, the row 1 is the pivot row and the number 2 in this row is the pivot element. This concludes Step 5.

Coming to Step 6, all the coefficients in the pivot row are divided by the pivot element, and x_r is eliminated from all rows except row 1. To make the coefficient of x_1 unity, row 1 is divided by 2. To eliminate x_1 from the second row, row 1 is multiplied by -

1/2 and is added to the second row. In row 3, already x_1 variable does not appear. After performing these operations, a new simplex tableau is formed as follows:

Tableau 2

C_j	2	1	0	0	0	
	x_1	x_2	x_3	x_4	x_5	RHS
	1	-1/2	½	0	0	4
	0	3/2	-1/2	1	0	6
	0	1	0	0	1	7
$\overline{C}_j$	0	2	-1	0	0	Z = 8

This new table shows the constraints in canonical form, and thus the improved basic feasible solution is $x_1 = 4$, $x_2 = 0$, $x_3 = 0$, $x_4 = 6$ and $x_5 = 7$. This completes Step 6, and now we move to Step 7. Again, from tableau 2, it is clear that this solution is not optimal because all $\overline{C}_j$ are not either negative or zero. Thus, we again repeat the process. A close inspection of Tableau 2 shows that variable x_2 will enter the basis and variable x_4 will leave the basis. Therefore, a new tableau is again formed with row 2 as the pivot row and the number 3/2 in this row as the pivot element. The new tableau is shown below:

Tableau 3

C_j	2	1	0	0	0	
	x_1	x_2	x_3	x_4	x_5	RHS
	1	0	1/3	1/3	0	6
	0	1	-1/3	2/3	0	4
	0	0	1/3	-2/3	1	3
$\overline{C}_j$	0	0	-1/3	-4/3	0	Z = 16

Tableau 3 shows that all relative profit coefficients are either negative or zero. Thus, the optimal point has been reached and the computations are terminated. The optimal value of the objective function is 16 and the optimal solution is $x_1 = 6$, $x_2 = 4$, $x_3 = 0$, $x_4 = 0$, $x_5 = 3$ and $Z = 16$.

Interpreting Simplex Tableau

The final simplex tableau, besides giving information about the optimal solution, also contains other useful information. From the simplex Tableau 3 above, one can readily determine that the values of basic variables are $x_1 = 6$, $x_2 = 4$ and the value of the objective function is 16. The value of non-basic decision variables at the optimum point is zero, except x_5 which is basic. The values of optimum slack variables are ignored because they do not affect the decision. However, if a slack variable is a basic variable at an optimal stage,

the corresponding constraint is non-binding and the corresponding resource is abundant. Otherwise, the constraint is binding and the resource is scarce. Note that the RHS coefficients can be viewed as resource constraints.

The simplex tableau also contains information about the per unit worth of a resource, which is also known as its shadow price. This information is useful while fixing priorities about allocation of funds for various resources. The shadow price of a non-binding resource is zero while it is non-zero for a binding constraint. The per-unit worth of a resource is given by $\partial Z/\partial b_i$, i=1, 2, ... , m. Any change in the availability of the resource corresponding to the binding constraints will change the optimum solution. The value of per-unit worth of a resource can be obtained from the final tableau in the objective function row under the starting basic feasible variables. In the example above, the per-unit worth of resources for the constraints number 1, 2, and 3 are 1/3, 4/3, and 0, respectively. This implies that an increase of availability of resource 1 by one unit will lead to an increase in the objective function value by 1/3 units.

5.3.8 Duality in LP

Associated with every LP problem (termed as the primal problem), there exists another problem known as the dual problem. The dual problem is formulated by transposing the rows and columns of the primal problem including the right-hand side and the objective function, reversing the inequalities and maximizing the objective function instead of minimizing.

Consider the following problem (primal problem):

$$\text{Minimize } Z = C^T x \tag{5.35}$$

subject to:

$$A x \geq b \tag{5.36}$$

$$x \geq 0 \tag{5.37}$$

Then, its dual problem can be stated as

$$\text{Maximize } Z_1 = b^T y \tag{5.38}$$

subject to:

$$A^T y \leq C \tag{5.39}$$

$$y \geq 0 \tag{5.40}$$

Here y is a column vector (m x 1). To write a dual problem, it is necessary to write the primal problem in a particular way. In a minimization problem, all the constraints must be written in $\geq$ form and all the constraints of a maximization problem must be written in $\leq$ form. For example, let the primal be:

$$\text{Min} \quad z = x_1 + x_2$$

$$\text{subject to:} \quad x_1 + 2x_2 \geq 5$$

$$2x_1 + x_2 \geq 4$$
$$x_1, x_2 \geq 0$$

then the dual is

$$\text{Max} \quad z_1 = 5y_1 + 4y_2$$
$$\text{subject to} \quad y_1 + 2y_2 \leq 1$$
$$2y_1 + y_2 \leq 1$$
$$y_1, y_2 \leq 0$$

Relationship between Primal and Dual Problems

Some interesting relations exist between a primal problem and its dual. These are:

1. The dual of the dual is the primal.
2. If the primal is a minimization problem then the dual is a maximization problem.
3. If dual has a finite solution, then the primal also has a finite solution.
4. For each variable in the primal, there exists a constraint in the dual and vice versa.
5. If any variable in the primal is unrestricted in sign, then the corresponding constraint in the dual is an equality constraint and vice-versa.
6. If the primal has an unbounded solution, the dual has either an unbounded solution or is infeasible.
7. In the final solution (if it exists), if any constraint in the primal problem is satisfied as equality, the corresponding dual variable will have a value greater than zero, and vice versa.

The dual variables, y, are also termed as simplex multipliers, Lagrange multipliers, shadow prices, marginal costs or opportunity costs. If a constraint is viewed as a resources constraint, the dual variable gives the marginal value of relaxing the constraint. It shows the change in the objective function per unit change in the RHS at the optimum, all other things remaining the same. Furthermore, if the problem contains a large number of constraints, it is more efficient to solve the dual since, in general, an additional constraint requires more computational effort than an additional variable. The solution procedure for a dual problem is on similar lines as for a primal. However, in the case of primal simplex, the feasibility is maintained throughout while the dual simplex starts with an infeasible solution while maintaining optimality.

The dual problem of cost minimization may be viewed as a product maximization problem where the product is maximized by varying the y variables, the imputed cost of each constraint. These implicit values are the shadow prices of constraints. They define the marginal value of the contribution of each constraint to the objective function, say how much more output can be obtained by relaxing a constraint by one unit. If the price of a resource is less than its shadow price, it is desirable to buy that resource and expand the production. The value of the slack variable at the optimum solution indicates cost (in terms of lowering the output) of using any activity which is not included in the optimum solution.

5.3.9 Post Optimality Analysis

Many times it is necessary to study the variation in the optimal solution resulting from the change in the various parameters, such as the cost coefficients, technological constants, or due to addition or deletion of variables or constraints etc. The study of the change in optimal solution due to these changes is known as the post optimality analysis. The following changes affect the optimal solution:

- Changes in the RHS constants of constraint equations,
- changes in the objective function coefficients,
- changes in the coefficients in the constraints,
- addition of new variables, and
- addition of new constraints.

Due to these changes, the optimal solution may change in the following ways:

- The optimal solution may remain unchanged,
- the basic variables remain the same but their values change, or
- the basic variables as well as their values change.

In most cases, it is not necessary to solve the problem from the beginning, and the final simplex tableau can be used to get the required answer.

5.3.10 Important Classes of LP Problems

Many day-to-day LP problems have some unique features which allow the use of special techniques for solution. Some of these problems are briefly discussed below:

Transportation Problem

In many real-life situations, a product is manufactured at a number of locations and it is required to transport it to a number of destinations. For example, a big company may have a number of production and demand centers, spread out geographically. The objective of a transportation problem is to devise a schedule of movement which minimizes the cost of transportation. This problem can also be formulated as a regular LP problem and solved using the simplex method. However, its special structure allows a more efficient and convenient procedure for its solution.

Network Flow Problems

A network is a configuration which consists of nodes joined by directed arcs. Each arc can support a flow. Associated with each arc are three parameters: a lower bound on the flow, an upper bound on the flow, and a cost value which represents the expenses of moving one unit of flow along the arc. A circulation is a set of flows in the network which preserves conservation of flow at each node. This means that the total flow into a node must equal the total flow out of the node.

The objective of a network flow optimization algorithm is to generate a circulation in the network which minimizes the total system cost (defined as the sum of the flows in each arc times the arc expense) subject to the capacity restrictions on the flow in each arc. Thus, the problem can be written as:

$$\text{Minimize } \Sigma C_i x_i \tag{5.41}$$

$$\text{subject to} \quad l_i \le x_i \le u_i \tag{5.42}$$

$$x_{in} = x_{out} \tag{5.43}$$

where x_i = flow in arc i, C_i = cost value for arc i, l_i = lower bound on arc i, u_i = upper bound on arc i, x_{in} = sum of flows into node j, and x_{out} = sum of flows out of node j.

The out-of-kilter algorithm (OKA) is a general method to generate the optimum circulation in a capacitated cost network. OKA begins with a circulation in the network. Now, maintaining a circulation, the flows are changed to achieve the objective without violating any of the continuity and limit constraints. OKA establishes a pricing system which assigns a price to each node. This technique has been described in detail in many books, such as Jensen and Barnes (1980). The Surface Water Allocation Model (AL-V) by Martin (1981) makes use of network flow optimization approach.

Other topics of interest in LP include the transportation problem and the assignment problem. Very efficient computer packages are available to solve LP problems which make use of this technique very attractive. For more on the linear programming and other optimization techniques, the reader may refer Rao (1979) and Taha (1982).

5.4 NONLINEAR PROGRAMMING

An optimization problem in which either the objective function and/or one or more constraints are nonlinear functions of decision variables is termed a Non-Linear Programming (NLP) problem. An NLP problem can be stated in the general form as:

$$\text{Minimize (or maximize) } f(x) \tag{5.44}$$

subject to

$$h_j(x) = 0, \; j = 1, 2, \ldots, m \tag{5.45}$$

$$g_j(x) \le 0, \; j = (m+1), \ldots, p \tag{5.46}$$

where $f(x)$ denotes the objective function, $h_j(x)$ are the equality constraints, $g_j(x)$ represent inequality constraints, and $x = [x_1, x_2 \ldots, x_n]$ is a vector of decision variables. Since $g_j(x) \ge 0$ can be written as $-g_j(x) \le 0$, inequality constraints can be denoted by $g_j(x) \le 0$. Here m and p are non-negative integers. If $m = p = 0$, then the problem is said to be unconstrained. The problem reduces to an LP problem if $h_j(x)$, $g_j(x)$, and $f(x)$ are all linear functions of decision variables.

Before proceeding further, let us define some properties of $f(x)$. Let $f(x)$ be a

continuous and continuously differentiable function. The vector of first partial derivatives of the function with respect to various variables is called the gradient of the function:

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right)^T \tag{5.47}$$

The gradient vector at a given point represents the direction along which the function values change at the maximum rate. The matrix of second partial derivatives of the function (if it exists) is known as the Hessian matrix. This is a square and symmetric matrix.

$$H(x) = \nabla^2 f(x) = \begin{vmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{vmatrix} \tag{5.48}$$

The solution of an NLP problem will be global if the objective function and feasible region are convex. In a convex region, a line joining any two points within the region will always be in the domain of the function. Mathematically, a function is said to be convex if for any two points x_1 and x_2, and for all α, $0 \leq \alpha \leq 1$:

$$F[\alpha x_1 + (1-\alpha)\, x_2] < [\alpha F(x_1) + (1-\alpha) F(x_2)] \tag{5.49}$$

The concept of a convex function is geometrically representation in Fig. 5.4. The convexity or concavity of a function $f(x)$ can also be ascertained by the following:

Function $f(x)$ is concave	if $H(x)$ is negative semidefinite
Function $f(x)$ is strictly concave	if $H(x)$ is negative definite
Function $f(x)$ is convex	if $H(x)$ is positive semidefinite
Function $f(x)$ is strictly convex	if $H(x)$ is positive definite.

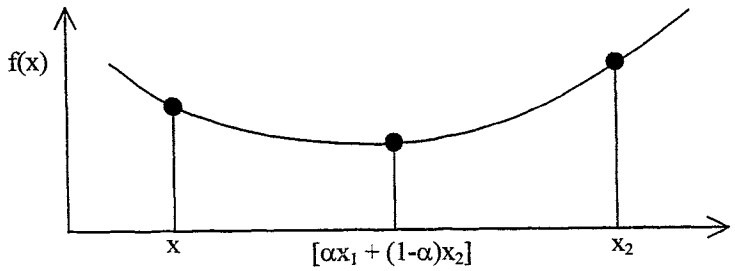

Fig. 5.4 Geometric representation of a convex function.

The convexity and concavity of function *f(x)* depends on the properties of the Hessian matrix. The Hessian matrix can be classified into following types:

Type of Hessian matrix	Condition to be satisfied
Positive definite	$x^T Hx > 0$ for all $x \neq 0$
Negative definite	$x^T Hx < 0$ for all $x \neq 0$
Indefinite	$x^T Hx > 0$ for some x and > 0 for some other x
Positive semidefinite	$x^T Hx \geq 0$ for all x
Negative semidefinite	$x^T Hx \leq 0$ for all x

A function of two variables may have partial derivatives of both variables zero at a point and in this case, the Hessian matrix will be neither positive nor negative definite. Such a point is called a saddle point.

5.4.1 Lagrange Multipliers and Kuhn-Tucker Conditions

The concepts of Lagrange multipliers and Kuhn-Tucker conditions are important and useful for constrained NLP problems. The Lagrange multiplier method converts an NLP problem with equality constraints to an unconstrained problem by developing an augmented objective function. For the constrained problem given by eq. (5.44) with equality constraints given by eq. (5.45), the Lagrange function (minimization problem) is:

$$L(x, \lambda) = f(x) + \lambda\, g(x) \tag{5.50}$$

where λ is the vector of Lagrange multipliers. If the original problem had *n* variables and *m* constraints, the augmented objective function will have *(n + m)* variables. These variables can be obtained by setting the partial derivative of $L(x, \lambda)$ to zero:

$$\frac{\partial f}{\partial x_i} + \sum_{j=1}^{m} \lambda_j \frac{\partial g_j}{\partial x_i} = 0, \qquad i = 1,\ldots n \tag{5.51}$$

$$\lambda_j g_j = 0, \quad j = 1, 2, \ldots m \tag{5.52}$$

The Lagrange multipliers are also known as dual variables (see Section 5.3.8), shadow prices, opportunity costs, etc. The j^{th} Lagrange multiplier represents a marginal change in the value of the objective function in the vicinity of the optimal solution with respect to the right-hand side of the j^{th} constraint. The Lagrange multipliers indicate how much the value of the objective function at the optimal point will change for a small change in the right hand side of the constraint.

A necessary condition for a local optimum is that the first derivative of the function is zero; this is also a sufficient condition for a convex or concave function. These conditions need extension when there are inequality constraints. Consider the minimization problem given by eq. (5.44) with m inequality constraints given by $g_i(x)$ which is differentiable and let $x \geq 0$. According to the Kuhn – Tucker conditions, an optimal solution x^* to this problem will exist only if there exist $\lambda_1, \lambda_2, \ldots, \lambda_m$ such that the following

conditions are satisfied:

1. If $x_j^* > 0$ then $\dfrac{\partial F}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \left.\dfrac{\partial g_i}{\partial x_j}\right|_{x_j^*} = 0$; j = 1,2,..., n (5.53a)

2. If $x_j^* = 0$ then $\dfrac{\partial F}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \left.\dfrac{\partial g_i}{\partial x_j}\right|_{x_j^*} \geq 0$; j = 1,2,..., n (5.53b)

3. If $\lambda_i > 0$ then $g_i(x_1^*, x_2^*, ..., x_n^*) = b_i$; j=1,2,...,m (5.53c)

4. If $\lambda_i = 0$ then $g_i(x_1^*, x_2^*, ..., x_n^*) \leq b_i$; j=1,2,...,m (5.53d)

5. $x_j^* \geq 0$; j=1,2,...,n (5.53e)

6. $\lambda_i \geq 0$; i=1,2,...m (5.53f)

The condition given by eq. (5.53a) is the necessary condition for an optimum provided the stationary point is not at the boundary. The condition specified by eq. (5.53b) is the supplementary condition when the optimum may be at the boundary. The condition given by eq. (5.53c) suggests that the inequality constraints introduced into the Lagrangian is binding (i.e., it suggests equality). According to eq. (5.53d), the Lagrangian multipliers of the constraints that are not binding vanish. The conditions given by eqs. (5.53e) and (5.53f) ensure non-negativity of the decision variables and Lagrangian multipliers.

5.4.2 Classification of Nonlinear Programming Methods

Depending on the nature of the problem, the sub-classes into which the solution techniques can be divided are the unidirectional search methods, the unconstrained optimization techniques, and the constrained optimization techniques. A brief discussion of these follows.

5.4.3 Unconstrained Nonlinear Programming Methods

An optimization problem without any constraint is called an unconstrained optimization problem. Although unconstrained optimization problems are rare in real life, in several nonlinear optimization techniques, the constrained problem is converted into an unconstrained problem which is subsequently solved. All the unconstrained optimization algorithms are iterative in nature. The computation is started at an initial point and advancement is made towards the optimum point in a systematic manner using some property of the function. A suitable criterion is chosen to terminate the computation when no further improvement is possible. These methods can be further classified into two categories, depending on whether the derivatives of the objective function are used or not.

Unidirectional Search Methods

In these methods, the objective function is optimized with respect to one variable only.

These techniques may not be useful for real life problems because mostly such problems have more than one variable. However, in some multidimensional problems, optimization is performed by systematically conducting unidirectional searches until the optimum is found.

The unidirectional search techniques are especially suitable for functions which are unimodal (having only one extreme) in the specified interval of uncertainty. The objective function is evaluated at selected points in the feasible region and then a part of it is discarded using the unimodality assumption. Clearly, the technique which requires a minimum number of function evaluations to reduce the feasible region to the required degree will be computationally most efficient. Some of the techniques in this category are the exhaustive search, the dichotomous search, the Fibonacci method, and the Golden Section method. Of these, the last two methods are most popular. In the Fibonacci method, the number of function evaluations depends on the accuracy desired and has to be specified beforehand. The placing of experimental points for function evaluations is determined using a sequence of numbers called Fibonacci numbers. In this sequence, the first two numbers are unity and thereafter each number is the sum of two previous numbers. Thus, the sequence is 1, 1, 2, 3, 5, 8, 13, 21, 34, 55... A portion of the search region is discarded at each stage until the solution with the desired accuracy is obtained.

Constrained problems involve the optimization of an objective function subject to one or more constraints and are much harder to solve than unconstrained problems with a comparable number of independent variables and the degree of non-linearity, because of the additional requirement that the solution must satisfy the constraints.

Direct Search Methods

The methods that use only function values to guide the search for the optimum value are either search methods constructed from geometric intuition or theoretically based techniques which have a mathematical foundation. These methods are also known as pattern search methods and do not require the gradient of the objective function. Let $f(x)$ be the function to be optimized. It is assumed that $f(x)$ is continuous, and $\nabla f(x)$ may or may not exist but is not available, and $f(x)$ is unimodal in the domain of interest. In case of multimodal functions, the solution may terminate at a local minimum.

Among the methods which do not need the derivative of the objective function, the most frequently used techniques are the pattern search methods (such as Powell's or Hooke and Jeeves' method), Rosenbrock's method of rotating coordinates, and the simplex method. In pattern search methods, starting from an initial point, several favourable unidirectional moves are made and the local behavior of the function is established. Using these, the pattern direction is determined as the most favourable direction of movement and a base point is established by moving in this direction. These two steps are repeated until the optimum value is found. The main motivation of moving in the pattern direction is that convergence can be considerably slow if movements are made only in coordinate directions. The cyclic use of searches in coordinate or any fixed set of directions is inefficient and may fail to converge to a local optimum. The Hooke & Jeeves method is a combination of one-

at-a-time exploratory moves (to understand the local behavior of the objective function) and the pattern moves (to take advantage of the pattern direction) are based on some heuristic rules. Powell's method is a theoretically based method that uses the history of iterations to build up directions for accelerations. The algorithm was devised assuming a quadratic objective function and it will converge in a finite number of iterations for such functions. A quadratic function in an n-dimensional real space is defined by:

$$F(x) = a + b^T x + x^T Q x \tag{5.54}$$

Another commonly used method is the Rosenbrock's method of rotating coordinates. Here, the coordinate system is rotated at each stage of optimization in such a manner that the first axis is oriented towards the locally estimated direction of the valley and all other axes are made mutually orthogonal and normal to the first axis. Because the coordinate system can be rotated depending on the need, this method can follow curved and steep valleys.

Computationally, direct search methods are relatively uncomplicated, hence the algorithm is easy to implement. On the other hand, they can be, and often are, slower than the derivative-based methods. Often, the objective function is the only reliable information that is available in a practical engineering problem and therefore, the direct methods are important.

Gradient-Based Methods

A major drawback of direct methods is that they require an excessive number of function evaluations to locate the solution. Then there may be a need to seek stationary points (the points where the first derivative of the objective function is zero) and thus a motivation to use the methods that employ gradient information. The gradient of a function can be computed using eq. (5.47). The gradient-based methods are iterative since the elements of the gradient are, in general, nonlinear functions of the decision variables.

The function value changes fastest by moving in the direction of the gradient. Hence, this direction is also called steepest ascent direction and the maximization methods which use the first derivative are also known as steepest ascent methods. The unconstrained minimization methods which use the derivatives of the objective function are also called steepest descent methods. However, the steepest ascent direction is a local property and it changes as one moves along the objective function surface. It is, therefore, necessary to evaluate the gradient at many points and thus the methods are iterative.

It is assumed here that the function *f(x)*, and its first and second derivatives $\nabla f(x)$, $\nabla^2 f(x)$ exist and are continuous. Some methods use the first derivative and some require both first and second derivatives. It is assumed that the elements of the gradient are available in closed form or can be reliably approximated numerically. The gradient-based methods employ an iterative procedure:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} s\{x^{(k)}\} \tag{5.55}$$

where $x^{(k)}$ is the solution at k^{th} step, $\alpha^{(k)}$ is the step-length parameter and $s\{x^{(k)}\}$ is the search direction in the N-dimensional space of the decision variables. The various methods differ in the manner in which *s(x)* and α are determined at each iteration. Usually $\alpha^{(k)}$ is selected so as to minimize *f(x)* in the $s\{x^{(k)}\}$ direction. Therefore, efficient single-variable minimization algorithms are required to implement these methods. Since the gradient is the direction of the steepest descent, a simple gradient search algorithm is:

$$x^{(k+1)} = x^{(k)} - \alpha \nabla f\{x^{(k)}\} \tag{5.56}$$

It remains to find a suitable value of α. The algorithm becomes slow near the minimum point as $\nabla f(x)$ tends to zero. The Cauchy method determines the step length α such that the function value $f\{x^{(k+1)}\}$ is minimum along the gradient direction. The Newton method that makes use of a second-order gradient is an improvement over this method. The iterative scheme of Newton's method is

$$x^{(k+1)} = x^{(k)} - \alpha \nabla^2 f\{x^{(k)}\}^{-1} \nabla f\{x^{(k)}\} \tag{5.57}$$

Marquardt Algorithm

This procedure was proposed by Marquardt (1963) by using the strengths of both Cauchy's and Newton's methods. Also known as Levenberg-Marquardt method, it requires second order information and allows for convergence with relatively poor starting guesses for the unknown variables. In general, a steepest descent procedure would be expected to converge for poor starting values but requires a lengthy solution time. The Gauss Newton method, on the other hand, will converge rapidly for good starting estimates. In this method, a least-square objective function is utilized. The search direction is given by

$$s\{x^{(k)}\} = -[H^{(k)} + \lambda^{(k)} I]^{-1} \nabla f\{x^{(k)}\} \tag{5.58}$$

where *I* is the identity matrix and λ is used to control the search direction as well as the step-length. When λ is very large, the search is in the gradient direction. When λ equals zero, the algorithm reduces to the Newton method. In the Marquardt algorithm, the initial values of λ are large and decrease towards zero as the optimum is approached.

Conjugate Gradient Methods

The conjugate gradient methods exploit the conjugacy concept by using gradient information. To understand the conjugate property of ellipses, consider an ellipse shown in Fig. 5.5. According to the conjugate property, the line AB joining the points of contact of two parallel tangents to an ellipse must pass through the center of the ellipse.

In Fig. 5.5, directions AB and AC are conjugate directions. Mathematically, for an *n*-dimensional case, if A is an *n*n* symmetric positive definite matrix, then a set of directions $\{S_i\}$ is said to be conjugate if

$$S_i A S_j = 0 \text{ for all } i \neq j \quad \text{and } i = 1,2, \ldots, n; \quad j = 1,2, \ldots n \tag{5.59}$$

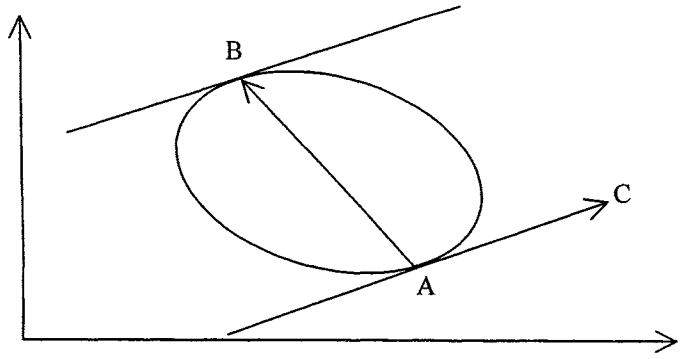

Fig. 5.5 The concept of conjugate directions.

The methods that locate the minimum of a quadratic function of n variables in steps whose number is related to n, are known as quadratically convergent. The convergence of a steepest descent method can be significantly improved by changing it to a conjugate gradient method. To complete the function search along n conjugate directions, n minimization cycles are necessary. After these n cycles, all the search directions are mutually conjugate (search has also been done along coordinate or non-conjugate directions) and the minimum of a quadratic would have been found. The Fletcher-Reeves method is a conjugate gradient method. In this method, the search direction is established as a linear combination of all the previous search directions and newly determined gradient.

A class of methods uses the following scheme to generate search directions at the k^{th} iteration

$$s\{x^{(k)}\} = -H^{(k)} \nabla f\{x^{(k)}\} \tag{5.60}$$

where $H^{(k)}$ is an $n*n$ matrix, termed as metric. When H changes at each iteration, it is called as variable metric. This is a positive-definite matrix and at the first iteration, it can be an identity matrix. The best variable metric method is the Davidon-Fletcher-Powell (DFP) method. In this, the first order derivatives are used to get the approximation of the Hessian matrix of the function. If the optimal step length is $\lambda^{(k)}$ then the iterations in the DFP method proceed as follows

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)} H^{(k)} \nabla f\{x^{(k)}\} \tag{5.61}$$

This method is stable and quadratically convergent and is used extensively. Rao (1979) found it to be the best general-purpose unconstrained optimization technique that use derivatives.

5.4.4 Constrained Nonlinear Programming Methods

The methods of constrained nonlinear optimization can be classified into two groups: direct

methods and indirect methods. In direct methods, the constraints are handled explicitly. Most popular in this category are the methods of feasible directions. In these methods, the current solution is improved by moving in usable feasible directions.

$$x^{(k+1)} = x^{(k)} + \lambda S \tag{5.62}$$

Here S is the direction along which a small step of size λ can be taken without leaving the feasible domain and at the same time improving the objective function value. There are two important steps at each stage of iteration: finding a usable feasible direction at the given point and determining the length of step along this direction. Except for convex problems, the algorithm may terminate at a local optimum. The Zoutendijk's method and Rosen's Gradient Projection method come under this category. Basically, the methods differ in the ways in which the usable feasible directions are found.

In indirect methods, the constrained problem is solved by solving a sequence of unconstrained problems. Penalty function methods, which come under this category, follow this strategy. Following Rao (1979), the problem given by eqs. (5.44) and (5.45) is converted to the following unconstrained minimization problem:

$$F = f(x) + r_k \sum_{j=1}^{p} G_j[g_j(x)] \tag{5.63}$$

where G_j is some function of the constraint g_j and r_k is the penalty parameter. The second term on the right hand side is termed as penalty term. Two variations of penalty function methods are used. In the interior penalty function method, a feasible starting point is obtained and the sequence of iterations coverage to the constrained minimum. In the exterior penalty function method, a slightly different form of eq. (5.63) is used and the function F increases as some power of the amount by which the constraints are violated. The sequence of computations converges to the desired solution from the exterior of the feasible region.

The Generalized Reduced Gradient (GRG) method is similar to the simplex method of LP in the sense that the n decision variables are partitioned into m basic (x_B) and (n-m) non-basic (x_N) variables. The problem is then expressed as:

$$\text{Min (or max) } f(x_B, x_N) \tag{5.64}$$

subject to

$$g(x_B, x_N) = 0 \tag{5.65}$$

and bounds on the variables. The basic variables can be expressed in terms of non-basic variables as $x_B(x_N)$. The objective function, expressed in terms of non-basic variables, is known as the reduced objective and its gradient, the reduced gradient. In the GRG method, a sequence of reduced problems is solved by a gradient search method. These methods are especially very successful in problems where the constraints are nearly linear.

Another powerful technique is the Successive Linear Programming (SLP)

technique. The SLP algorithm also solves nonlinear programming problems by solving a sequence of linear programs. These algorithms are attractive for large sparse nonlinear optimization problems where usually only some variables appear nonlinearly in the objective function and/or constraints.

5.4.5 Some Common Problems in NLP Applications

While solving an NLP problem, several things require attention. Firstly, an algorithm may fail because the problem was wrongly formulated. Unfortunately, this error may not be revealed directly, but rather through the failure of some portion of the algorithm.

Overflow in the user-defined function is a common problem. Overflow may occur when the optimization problem has an unbounded solution since in this case the function value gradually becomes large. An unbounded problem is also indicated in a minimization problem when there is a consistent decrease in the function with no sign of convergence. The appropriate remedy depends on the reason for the unboundedness. If the original unconstrained problem was incorrectly formulated, the user must redefine it. The user may indeed wish to find a particular local minimum of an unbounded function. This can be achieved by imposing a small value of the maximum step allowed during each iteration or by adding bounds on the variables to keep the iteration within the desired region.

If the program is self-developed, there might be errors in programming, for example, in computation of step-length or the gradient or Hessian of the objective function (inaccurate finite-difference approximation). Computation of the gradient of the function is necessary in most NLP techniques. Numerically, the gradient can be approximated up to sufficient accuracy by the central difference or the forward difference formula. A poor scaling of variables can also cause instability in numerical procedures. There may be an imbalance between the values of the function and changes in decision variable x; the function values may change little even though x changes significantly. Conversely, the function may change extremely rapidly even though x changes hardly at all.

Overly stringent accuracy requirements may also cause a failure of the algorithm. In some instances, an algorithm may indicate that it has been unable to terminate successfully, whereas, in fact, the solution has already been found. A good programming practice is to input the maximum number of times the problem functions are evaluated, and/or an upper bound on the number of iterations. Such bounds are useful in several situations, and serve as a protection against errors in formulation that would otherwise not be revealed. In particular, the upper bound on the number of function evaluations may be reached when an optimization problem has an unbounded solution. This failure will also occur if a large number of iterations are performed without any significant improvement in the objective function.

A detailed discussion of these techniques is available in many text books, such as Rao (1979), Wagner & Himmelblau (1972), and Reklaitis et al. (1983). Many books contain steps of the various solution algorithms. One can develop a working, if not efficient, program by carefully following these steps.

5.5 DYNAMIC PROGRAMMING

Dynamic Programming (DP) is an enumerative technique developed by Richard Bellman in 1953. This technique is used to get the optimum solution to a problem which can be represented as a multistage decision process. The entire DP formulation is based on the Bellman principle of optimality. According to this principle, an optimal policy has the property that whatever the initial state and decisions are, the remaining decisions must constitute an optimal policy with respect to the state resulting from the first decision. The proof of this theorem can be obtained by contradiction. In Fig. 5.6, let the optimal path for going from A to D be ABCD. According to Bellman's theorem, the optimal path from B to D will be BCD and not BED. If the optimal path from B to D is BED then the optimal path from A to D will be ABED and not ABCD.

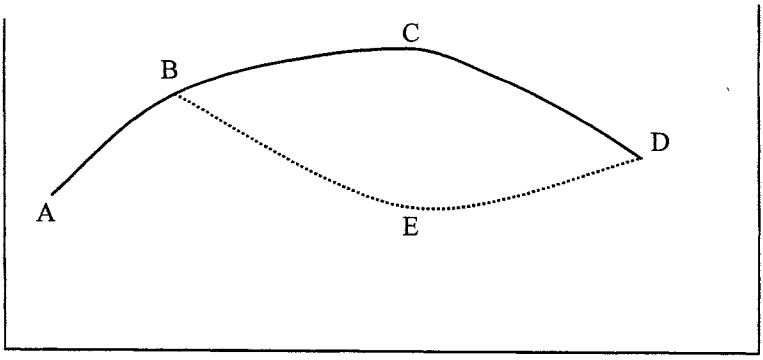

Fig. 5.6 Illustration of the Principle of Optimality.

Dynamic programming is not a class of optimization techniques, but as an algorithm it is a powerful procedure to solve sequential decision problems. Many problems in water resources involve a sequence of decisions from one period to the next period and are known as sequential decision problems. Such problems can be decomposed into a series of smaller and easily solvable problems that can be conveniently handled by DP. For example, the operation of a reservoir proceeds in a sequential manner from one time period to another. An important feature of DP is that non-linearities and constraints can be readily accommodated. In fact, constraints serve to reduce the region to be covered in computations and are helpful in that sense. In a DP problem formulation, the dynamic behavior of the system is expressed by using three types of variables:

State variables - define the condition of the system. For example, the amount of water stored in the reservoir may represent its state. If a problem has one state variable per stage, it is called a one-dimensional problem; a multi-dimensional problem has more than one state variable per stage. Thus, the optimization of operation of a system of two reservoirs will have two state variables, one for each reservoir.
Stage variables - define the order in which events occur in the system. Most commonly, time is the stage variable. There must be a finite number of possible states at each stage.
Control variables - represent the controls applied at a particular stage and transform the

state of the system. For a reservoir operation problem, the release of water from the reservoir is a typical control variable.

The dynamic behavior of the system is expressed by an equation known as the system equation. It can be written in discrete form as:

$$s(t+1) = f[(s(t), u(t),t] \qquad t = 1, 2,...,N \tag{5.66}$$

where *s(t)* is the state variable at time *t*, *u(t)* is the control applied at time instant *t*, which lasts for a finite duration and f[.] is the given functional form. The state of the system at any stage should lie in the domain of admissible states at that stage; the controls should also lie in the admissible domain at that stage:

$$s(t) \varepsilon S(t), \qquad u(t) \varepsilon U(t) \tag{5.67}$$

where *S(t)* and *U(t)* are the domains of admissible states and controls at stage *t*. The function f[.] should be invertible, i.e., it must be possible to express the decision variable as an explicit function of state variables:

$$u(t) = f^{-1}[s(t+1), s(t), t] \tag{5.68}$$

For an invertible system, the order of the state vector is equal to the order of the control vector. Thus, the knowledge of stage variables enables one to compute the decision variables. For instance, in reservoir regulation problems, the mass balance equation (which is also the state equation) is invertible.

With each state transformation, a return is associated which may either represent benefits or costs. Typically the benefits are maximized and the costs are minimized. The optimal decision made at a particular stage is independent of decisions made at the previous stage, given the current state of the system. It is necessary that the objective function of a DP problem should be separable. It should be possible to write individual objective functions at each stage as functions of state and/or decision variables at that stage. Likewise, the constraints should also be separable or each constraint should be associated with an individual stage only. For a multi-dimensional problem, it would be necessary to evaluate the objective function for all discrete combinations of state variables.

A set of decisions for each time period is called a policy and the policy which optimizes the objective function is called the optimal policy. The set of states resulting from an application of the policy is called the state trajectory. For example, the volume of water stored in a reservoir can be considered to be its state. The state of a reservoir is transformed due to inflows and can be controlled by releasing water from the storage. This water can be used for some useful purpose (e.g., irrigation) to yield monetary returns or it may also cause flood damage downstream and a cost is associated with this damage. A problem of optimizing the operation of a reservoir could be to find the releases (controls) which yield the best returns.

5.5.1 Recursive Equation of DP

Let R[s(t), u(t), t] be the return from operating a system which is at state *s(t)* and the control *u(t)* is applied at stage *t*. Further, let F[s(N),N] be the sum of returns from application of controls from the initial stage at $t = 0$ to the final stage at $t = N$. The objective of maximizing the sum of returns from the system can be expressed as

$$\text{Max } F[s(N), N] \tag{5.69}$$

Let the state of the system at t = 0, s(0) ε S(0) be known and the returns F[s(0),0] be also known. Let $F^*[s(0),0]$ be the optimum value of these returns. Now, consider the first stage (of duration Δt). The optimal return for this period is given by

$$F^*[s(1),1] = \underset{u(0)\ \varepsilon\ U(0)}{\text{Max}}\ R[s(0),u(0),0] + F^*[s(0),0] \tag{5.70}$$

Here, R[s(0),u(0),0] indicates the returns that are obtained when at stage 0, the system is in state *s(0)* and control *u(0)* is applied [U(0) is the domain of admissible states]. This equation is solved for each discrete level of state at $t = 0$ as a function of control variables *u(0)*. To do this, the state is discretized into a number of discrete levels. Now a particular lattice point is chosen and all the admissible levels of decision variables which lead to this state are chosen. For each of these decision variables, the return F[s(1),1] is calculated. The maximum among these returns gives the value of $F^*[s(1),1]$. This computation is repeated for each discrete value of *s(1)* and the results are stored. The progress of computations in a forward algorithm is shown in Fig. 5.7.

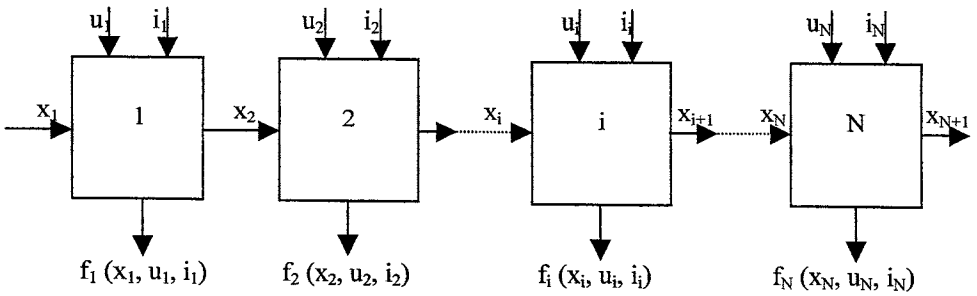

Fig. 5.7 Progress of DP computations in a forward algorithm.

Computations are performed in a similar fashion for stage 2, 3.... , N. Note that there can be more than one decision variables at a given state. The recursive equation for any stage *t* can be written as:

$$F^*[s(t),t] = \underset{u(t-1)\ \varepsilon\ U(t-1)}{\text{Max}}\ R[s(t-1),u(t-1),t-1] + F^*[s(t-1),t-1] \tag{5.71}$$

Finally, at the end of stage N, the values of $F^*[s(t),t]$, $t = 1,2 \ldots N$, are available. The optimal value of control variables or the optimal policy is obtained by tracing back the values of returns, corresponding to those states which satisfy the initial and final values and the constraints. The optimal state trajectory can be determined by using the system equation, once the optimal policy is known. This is the method of enumeration or brute-force optimization. The tree of computations generated in enumeration is shown in Fig. 5.8.

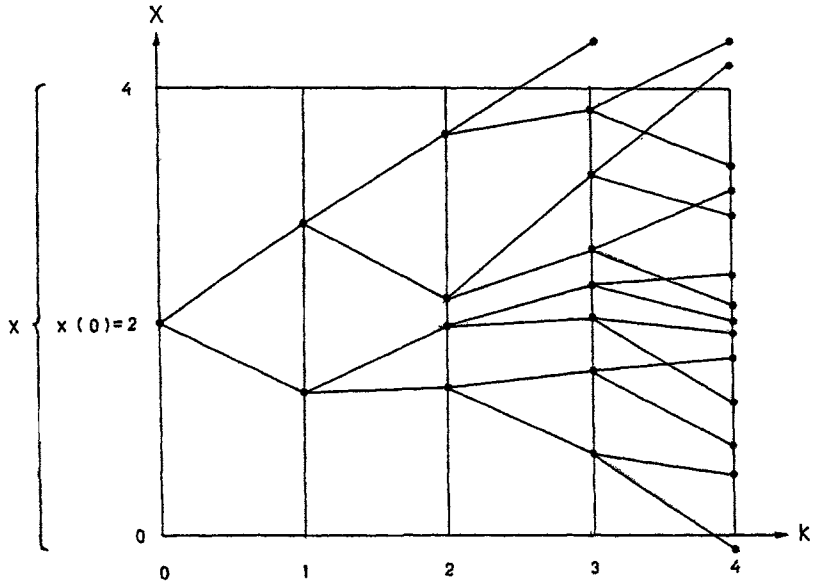

Fig. 5.8 Tree generated by enumeration in DP.

The above computational scheme of dynamic programming is known as the forward algorithm, since the computations start at the initial value of the state variable at stage 1 and move forward stage-by-stage. In contrast with this, computations can also commence at the final value of the state variable at the last stage and can move backwards. The optimal policy is retrieved by tracing forward from the returns. This algorithm is termed as the backward algorithm.

Example 5.3: A system of three reservoirs is to be constructed. The yield versus cost at the reservoir sites is given in the following table.

Yield	Cost		
	Reservoir 1	Reservoir 2	Reservoir 3
0	0	0	0
20	15	10	20
40	30	35	40

Find the minimum cost combination to get a total system yield of 60 and 80.

Solution: This problem can be solved using enumeration method of DP by carrying out computations in various stages. Here, stage refers to the number of reservoirs in the system at any time. At stage 1, only reservoir 1 is considered. Therefore, the cost of providing various yields (in the range 0 to 40) at stage 1 will be same as the cost of providing yields by reservoir 1. At stage 2, the reservoir 2 is added and the combination of reservoirs 1 and 2 is considered. The computation of cost to get various yields in the range of 0 to 80 from reservoir 1 and reservoir 2 is given in the following table:

Total Yield	Yield from		Cost of		Total cost at stage 2
	Reservoir 1	Reservoir 2	Reservoir 1	Reservoir 2	
0	0	0	0	0	0
20	20	0	15	0	15
	0	20	0	10	10*
40	40	0	30	0	30
	20	20	15	10	25*
	0	40	0	35	35
60	20	40	15	35	50
	40	20	30	10	40*
80	40	40	30	35	65*

* indicates optimal solution for that yield.

Proceeding to stage 3, all the three reservoirs are considered. Now, for a total system yield of 60 and 80, the possible combinations and corresponding costs are given in the following table:

Total Yield	Yield from		Cost of		Total Cost
	Reservoir 3	Stage 2	Reservoir 3	Stage 2	
60	40	20	40	10	50
	20	40	20	25	45
	0	60	0	40	40*
80	20	60	20	40	60*
	40	40	25	40	65

* indicates optimal solution for that yield.

The last column of the above table gives the total cost of providing various yields. It can be seen from the table that for a total system yield of 60, the minimum cost is 40 units. Now, the optimum solution can be traced backward. To get a yield of 60, reservoir 3 should not be constructed and a yield of 60 units should be obtained from stage 2. From computations of stage 2, one can note that reservoir 1 should give a yield of 40 and a yield of 20 units must be obtained from reservoir 2.

Similarly, for a total system yield of 80, a yield of 20 must be planned from reservoir 3, and a yield of 60 from stage 2. The table for stage 2 shows that a yield of 40 should be obtained from reservoir 1 and reservoir 2 must provide a yield of 20.

It is clear from this example that computations are quite simple in the enumeration method. However, as the number of variables and their discretisation increase, one encounters a major computational problem because the computer memory and time requirements increase exponentially. For example, consider a two-reservoir problem. If reservoir 1 takes on 40 feasible states and reservoir 2 takes on 20 feasible states, the DP recursive equation would have to be evaluated at 800 points in each period. In general, if there are n state variables at each stage and each state variable has m discrete values then one needs to evaluate the objective function at m^n points. Each of these values and equal number of values of state variables need to be stored in computer memory at each stage. One can easily imagine the consequences as the number of state variables increases or as a finer discretisation is used. This problem arising due to storage and comparison of abnormally large number of variables was termed by Bellman as the *curse of dimensionality* (see Bellman and Dreyfus, 1962; Buras, 1966).

Several procedures have been developed to overcome the curse of dimensionality. Intuitively, the number of variables to be stored can be reduced by adopting a coarser grid for initial computations. After the optimal solution is located, a finer grid can be constructed in the vicinity of this solution. However, in this scheme, one may miss the global optimum and the solution may converge to a local optimum. Another attractive procedure to alleviate the curse of dimensionality is the Discrete Differential DP (DDDP).

4.5.2 Discrete Differential DP

The technique which uses the concept of increments for state variables was introduced by Larson (1968) and termed as state increment DP (SIDP). Heidari et al. (1971) used this concept for reservoir operation studies and referred it as discrete differential dynamic programming (DDDP). The major difference between Larson's SIDP and DDDP is the time interval used in computations, which is variable in the former and fixed in the latter. In fact, DDDP is a generalization of SIDP.

The DDDP procedure starts with an assumed trial state trajectory, which is a sequence of feasible state vectors resulting in a corresponding initial policy, and an initial value of the objective function. The DP recursive equation is then used to examine a restricted set of values of the state variables or the neighboring states that are one small increment above and below the trial state trajectory. This subdomain is called a corridor (see Fig. 5.9) and the trial trajectory lies at the center of the corridor, although this is not a necessary condition. More than one discrete states on either side of the trajectory may be chosen but the choice of three quantized states at each stage is most suitable for computational efficiency. Now, DP computations are performed within this restricted corridor and a neighboring trajectory that gives a better value of the objective function is found. This new trajectory replaces the trial state trajectory and the procedure continues. The procedure is assumed to have converged to a local optimum when the trajectories in two successive iterations are the same and a better value of the objective function cannot be found. This can be interpreted as a sort of successive approximation scheme. An initial estimate of the policy is made and this is used to construct an improved estimate. The scheme cannot assure the global optimum and may converge to a local optimum. However,

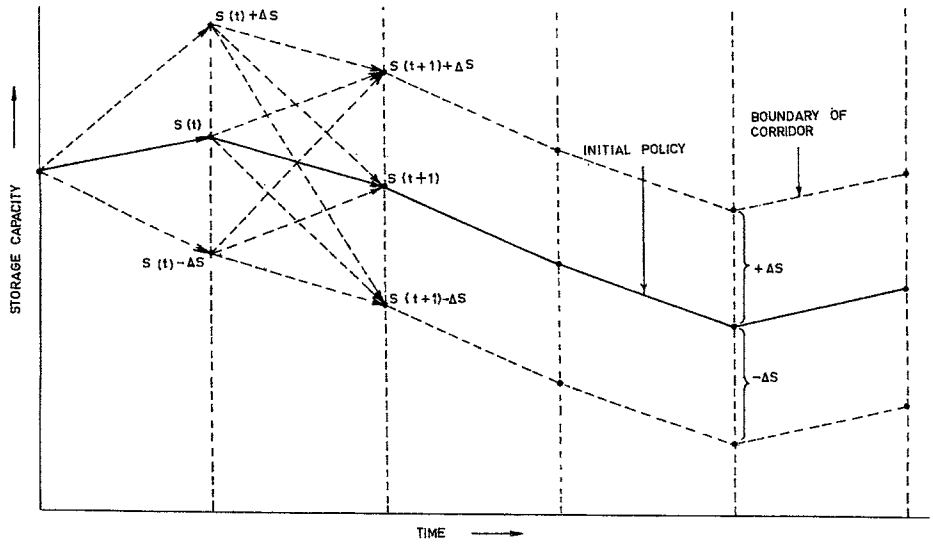

Fig. 5.9 Sub-domain of DDDP computations.

by starting from different initial solutions, the possibility of finding the global optimum is increased. This technique is particularly suitable for invertible systems. The water resources systems are mostly invertible. For example, assuming that the inflows to a reservoir are known, the releases from it can be determined if the states of the reservoir at different times are known.

When a real-life problem is solved by DDDP, the number of state transitions is considerably reduced. Computer time and storage requirements are drastically reduced. To obtain quick convergence, two procedures have been suggested to compute the increments of state variables. The first is to keep the increments small and constant throughout an iteration. The second is to reduce the size of increments as the iterations proceed. Goulter and Tai (1985) recommend that at least 5 to 10 discrete states should be used at each stage. In general, strong correlation is found between the number of iterations required for convergence and the size of increments used at each iteration. Furthermore, several iterations with a small increment of state variables should be allowed at the end of computation to improve the value of the objective function.

5.5.3 Advantages and Disadvantages of DP

DP is essentially an enumerative technique which is specially suited to multistage decision problems. Some of the advantages of using this technique for problems of water resources systems are:

(i) The DP formulation is the same for linear as well as nonlinear problems. Thus, no extra effort is required for nonlinear problems. This property is very useful since many problems cannot be realistically linearized.

(ii) The incorporation of constraints in linear and nonlinear programming problems is more difficult than in DP problems. In DP, the constraints serve a useful purpose. They limit the feasible region and thus may lead to reduction in the computational time requirement.

(iii) The stochastic nature of a problem can be easily considered in the DP formulation. The algorithm developed for a deterministic problem does not have to be significantly changed to incorporate stochasticity. This is in contrast with other techniques where incorporation of stochasticity requires too much change in the algorithm and significant increase in computational time.

Along with the above advantages, there are also some disadvantages. The major disadvantage is because DP is not basically tailored in such a fashion that generalized programs can be written using it. Thus, a new computer program has to be developed or an existing program has to be significantly modified and tested for each new application of the technique. On the other hand, standard computer programs are widely available for LP.

5.6 STOCHASTIC OPTIMIZATION

The system behavior and input of water resources projects display stochasticity which needs to be appropriately accounted for. Depending on the way the stochastic nature of the system and inputs are treated in an optimization formulation, the solution techniques are classified as implicit stochastic optimization (ISO) or explicit stochastic optimization (ESO). In ISO, the system and the stochastic nature of the input are represented by statistical models and these models are used to generate realizations of the inputs time series over the operation horizon. A suitable deterministic optimization technique is applied to find the optimum decision variables for each input realization. Since data generation techniques and simulation are used, the problem need not be solved analytically. This approach is known as Monte Carlo technique.

In this approach, inputs are represented by a time series model or probability distribution and the system behavior is modeled. Inputs are transformed into system outputs and statistical characteristics of the outputs are gathered. This enables estimates of various output probabilities to be made which can be related to risk. A regression analysis is carried out to establish relationship among the system inputs, state variables, outputs, and optimum decisions for all of the generated sequences. This relationship can be used to take operation decisions when the future is unknown. This strategy was used by Young (1967) to derive reservoir operation rules. Since deterministic DP was used, the procedure was termed as Monte Carlo Dynamic Programming (MCDP).

A drawback of ISO is that it requires considerable computational time and efforts to generate a large number of synthetic input sequences, solve a large number of optimization problems, and do multivariate regression analysis. Furthermore, one may not always get a good relationship between the variables involved.

In contrast to ISO, ESO directly uses the probability distributions of inputs in optimization. The objective function is the sum of benefits over all states, stages, inputs,

and decisions multiplied by the probability that these conditions occur. Thus, the objective function represents the expected total benefit for the system. Computations are performed to find that set of probabilities which maximizes the expected total benefit. These probabilities are then used to calculate the conditional probabilities of making a decision given that the system is in a certain state at a given stage and it receives certain inputs. Ideally, the solution should yield a pure strategy, i.e., one decision should have a probability of unity and all other decisions should have zero probability. Unfortunately, one does not always obtain pure strategies; "mixed" strategies are occasionally obtained and are suitably used to arrive at the decisions. In the ESO, if the optimization technique used is LP, this problem is known as stochastic LP. The linear decision rules are used in LP to disallow the possibility of the mixed strategy.

The advantage of ESO procedures over ISO is that the results obtained from ESO are based on a conditional probability distribution at each stage. Therefore, more information is utilized for the choice of a decision at each stage. Instead of just a single estimate, the probability distribution of inputs is used. The method involves a great deal of computation time and storage so that its application to complex systems is severely limited. The common assumption is that the inputs at each stage have a steady-stage probability distribution and the system can be represented by a cyclic (repetitive) operation. Thus, only one cycle needs to be analyzed, and the system properties can be represented using a small number of discrete values for states, inputs, and decisions.

5.6.1 Chance-Constrained Linear Programming

The future inputs to a water project, e.g., a reservoir, are random and hence the resulting states and decisions, such as storage and releases, will also be random. In some optimization problems, the constraints that limit the range of values of these specify the percentage of time that these ranges can be violated. The constraints that explicitly define these limits are termed as chance-constraints. Chance constraints are used in mathematical programming to constrain the optimization to those decisions that represent a failure probability smaller than the constraint value. Basically, a constraint on the probability of failure is transformed into its deterministic equivalent. A chance-constraint that ensures that some variable $\hat{x}$ is no greater than the value of a random variable X at least some fraction α of time is written as [Loucks and Dorfman, 1975]:

$$\text{prob}[\hat{x} \leq X] \geq \alpha \tag{5.72}$$

where *prob* denotes probability.

In reality, many problems, such as reservoir sizing and operation, require a non-linear optimization formulation but the use of linear decision rules permits their framing as a LP problem. ReVelle et al. (1969) were the first to present a chance-constrained formulation to solve reservoir design and operation problem. They used linear decision rule which allows simple formulation of chance constraints for problems dealing with reservoirs and the probability distribution of inflows can be easily considered. The linear decision rule relates release from a reservoir to storage:

$$R_t = S_t - b_t \tag{5.73}$$

where R_t is the release from the reservoir, S_t is the initial storage, and b_t is a decision variable, all for period t. The objective of their study was to minimize the capacity of the reservoir and to determine optimum coefficients in the decision rule, subject to chance constraints on freeboard, storage, and releases. The freeboard consideration requires that at least a volume V be available at the end of the period t for temporary storage of flood peaks. In other words, the storage S_t at the end of the period should not be greater than the reservoir capacity C minus the freeboard requirement V, at least 100α % of the time:

$$P(S_t \leq C - V) \geq \alpha \tag{5.74}$$

By substitution of the linear decision rule from eq. (5.73) and by using continuity equation:

$$S_t = S_{t-1} + I_t - R_t = I_t + b_t \tag{5.75}$$

The chance-constraint can be formulated as

$$P[I_t + b_t \leq C - V] \geq \alpha \tag{5.76}$$

$$F_{It}(C - V - b) \geq \alpha \tag{5.77}$$

or

$$C - V - b \geq i_\alpha \tag{5.78}$$

where i_α is the α quantile point from the distribution of I_t, $F_{It}(i_\alpha) = \alpha$. The above use of the linear decision rule, chance constraints, and optimization in a LP formulation yields chance-constrained LP. Though the use of the rule results in conservative designs, the model is useful in multireservoir studies and preliminary screening studies. It enables simpler application of chance-constraints in a LP problem.

5.6.2 Stochastic Dynamic Programming

The DP formulation where the stochastic nature of the variables is not considered is known as deterministic DP. Since many water resources variables are stochastic in nature, the DP approach is frequently modified to account for this stochasticity. The DP formulation which takes into account the stochastic nature of variables is known as Stochastic Dynamic programming (SDP). DP can be adapted in two ways to handle stochastic input data. The first of these is called Monte Carlo Dynamic Programming. The basic idea of this procedure (Monte Carlo techniques are discussed later in Section 5.9.2) is to generate a number of synthetic streamflow sequences which match the properties of observed inflow series. For each of these series, a DP formulation is used to get the optimum policy and so there will be as many policies as the number of synthetic sequences. These optimum policies are then used in a regression analysis to determine the causal factors influencing the optimal policy.

An alternative to this procedure is to formulate the problem as a true stochastic dynamic programming problem and use the policy iteration and the policy improvement

routines. The SDP approach for development of an optimal operation policy of water resources systems was first used in the 1950s. In this method, it is necessary to analyze streamflows on a time period (usually monthly) basis and express the relation between these as transition or conditional probabilities of period-to-period flows. The computation of transition probability matrix has been described in Chapter 4.

Where the probability of various values of a variable are dependent on the value of that variable in a previous time period, the sequence of events so described is called a Markov Chain. When the probability of being in a given state after another given state is a fixed quantity, it is termed as constant or stationary conditional probability. Many hydrologic variables display this property.

An examination of monthly river flow data at a number of sites shows that the conditional probability connecting monthly flows is not a stationary quantity. Therefore, the sequence of monthly flows can be regarded as connected by twelve sets of different conditional (or transitional) probabilities to form a non-stationary (cyclic) Markov Chain. The derivation of the optimal policy is based on the assumption that the system described is ergodic. For an ergodic system, the final system state is independent of the starting state. For example, in a reservoir operation problem, this is equivalent to stating that no matter what the state of the reservoir at the start of computations is the steady state of the system will be independent of that starting state.

Consider that a system is to be managed for one time period only and at the end of this time period, its state is of no value. Let this be the end point of the study period. The calculations step backward in time. The optimum decision r for this last time period is obtained by the following equation:

$$f_1(s_1, I_2) = \max_r R(r) \tag{5.79}$$

where $f_1(s_1, I_2)$ is the expected return from the optimal operation of a system which has 1 time period to the end of the operation horizon; s_1 is the state of the system at the start of the 1st calculation time period; I_2 is the input to the system in the 1st time period; and $R(r)$ is the return obtained consequent to applying controls in this period. The decision should satisfy the applicable constraints. Thus, for each discrete value of state variable and for each discrete value of input, there will be a value of r which will give the maximum $R(r)$.

Now, consider the period before the last time period (which is the calculation time period number 2, as the counting is backward). According to Bellman's Principle of Optimality, an optimal policy for these last two time periods must include one of the policies already determined for time period number 1, relating to the state that the system attains when the optimal decision is made for time period number 2. Often, the next state that will be occupied is not a deterministic function of the current state and decision and it may depend on uncertain events, such as rainfall, streamflow, or political decisions. Transition probabilities are used to account for such situations. One can then write

$$f_2(s_2, I_3) = \max_r \left[R(r) + \sum_{I_3=0}^{I_3=I_{3\max}} P(I_2 \mid I_3)\, f_1(s_1, I_2) \right] \tag{5.80}$$

where $P(I_i|I_{i+1})$ is the transition probabilities connecting the input in the i^{th} time period I_i with input in the $(i+1)^{th}$ time period I_{i+1}. According to this equation, for all values of s_2, the state at the start of the second computational time period (or the second last in the study period) and all possible values of the input during the preceding period (I_3), there exists a value of the control (r) which maximizes the objective function. The value of r is chosen from its feasible domain. The right hand side of this equation expresses that r will be chosen such that the return from the decision r in the current (second) time period, together with the worth of the system state at the start of the succeeding time period, will be a maximum. Note that the value of the system state at the start of the next time period for all possible states of that system will be known from the preceding calculations. Eq. (5.80) uses the fact that one period's input is related to the preceding period's input by the conditional probabilities $P(I_2|I_3)$. Thus, for each value of I_3, it is possible to assign a probability to I_2 so that the probability of a situation which derives from I_2 (such as s_1) can similarly be assigned that probability. Thus it is possible with given values of s_2 and I_3 to range over all possible values of r and determine both the current period returns and the expected value of the resulting state of the system. The value of r is chosen to maximize the sum of these returns.

This procedure is repeated for all possible values of s_2 and I_3 and function f_2 is completely evaluated. Similarly, f_3 is evaluated from f_2 and so on, leading to the general formulation:

$$f_i(s_i, I_{i+1}) = \max_r \left[R(r) + \sum_{I_{i+1}=0}^{I_{i+1}=I_{i+1\max}} P(I_i \mid I_{i+1}) \, f_{i-1}(s_{i-1}, I_i) \right] \tag{5.81}$$

In table below, the temporal progress of the various computations is presented.

Time period index for computations	i+1	i	i-1	...	2	1	End of study period \ Start of computations
Input I	I_{i+1}	I_i	I_{i-1}	...	I_2	I_1	
System state at the beginning of the period	s_{i+1}	s_i	s_{i-1}	...	s_2	S_1	
Return function	$f_{i+1}(s_{i+1}, I_{i+2})$	$f_i(s_i, I_{i+1})$	$f_{i-1}(s_{i-1}, I_i)$	...	$f_2(s_2, I_3)$	$f_1(s_1, I_2)$	
		Progress of time			→		

Starting at some time in the future and using the transition probabilities between the input in one time period and that in the adjacent time period, it is possible to calculate the values of r for each time period as a function of state variables s_i and I_{i+1}. Put together, these r's form an optimal policy for control of the system. Under certain circumstances, this policy converges when the values of r that are used to evaluate the function $f_i(s_i, I_{i+1})$ repeat for all values of i, as i becomes larger. This method of determination of an optimal policy is termed as policy-iteration routine and it is an effective way to develop an optimal policy.

By starting this procedure at an arbitrary future time period, and stepping backward one period at a time, it is easy to find optimum control at any state of the system.

The decision to be made in early time periods will be affected by the conditions specified at the end of the study period, which is the start of the calculation procedure. However, by carrying the calculations sufficiently far from the end, the results will be free from the influence of the starting conditions, and an optimum policy can be established. The result of this calculation is a set of matrices of decisions to be made under all states of the system, i.e., for all values of s_i and I_{i+1} in each period, say a month, of the year.

A major advantage of this formulation is that the developed policy can be used for design as well as for actual operation. The state of the system is described by two variables, the input in the preceding period I_{i+1} and the system state s_i. These quantities are available at the time when the decision is to be taken. This formulation can also be used to assess a given design. A model of this kind also allows exploration the sensitivity of the return and the uncertainties associated with it to variation in input parameters (correlation, uncertainty, etc.), constraints on the system operation, and return functions.

5.7 MULTI-OBJECTIVE OPTIMIZATION

Water resources systems are usually characterized by multiple objectives, multiple decision-makers, and multiple constituencies. Before discussing this topic, it is useful to differentiate between multi-purpose and multi-objective. According to Major (1977), the term multi-objective refers to multiple economic, social, environmental, and other objectives of water development; and multipurpose refers to multiple functions like navigation, flood control, water supply, recreation, etc. Clearly, these two terms are not synonymous -- purposes can vary and still be aimed at the same objective, and one purpose can fulfill more than one objective.

A fundamental characteristic of multi-objective water-resources problems is that the various objectives are often non-commensurate and may be in conflict. In view of this, a multi-objective analysis has a central role in water resources planning and management. In fact, the area of water resource planning and management is responsible for many developments in the field of multi-objective optimization.

Trade-offs are an inherent part of negotiation, of reaching consensus, and of compromise solutions. Thus, if two solutions are compared, may be that the first solution achieves higher levels of certain objectives and lower levels of other objectives when compared to the second solution. For example, the use of a reservoir for flood control purposes may be achieved at the expense of reducing benefits from conservation operation. Moreover, as environmental and other socioeconomic aspects now dominate and influence policy decisions, the importance and the need for a multi-objective analysis has become more critical and evident. The multi-objective analysis should be viewed not only as a methodological approach but also as a philosophy. However, the analysts and decision-makers should be aware of the properties, efficacy, and limitations of the multi-objective analysis.

While formulating and screening alternative plans, the objectives of the project should be given explicit and quantitative consideration. This is especially important in the

planning of river basins, where there are likely to be several conflicting and non-commensurate objectives. The objectives that can't be expressed in a common unit are non-commensurate. For example, one may want to maximize irrigation benefits (measured in monetary units) and environmental quality (measured in units of pollutant concentration). Traditionally, only one objective (economic efficiency) is considered and the other objectives are included either as constraints or as commensurate with the primary objective in some way. This limitation can be overcome in multi-objective analysis.

A multi-objective analysis is 'vector optimization'. The essential difference between a scalar optimization problem and a vector optimization problem is that the objective function is a vector instead of a scalar. To some persons, the term vector optimization is contradictory in itself since one cannot optimize a vector. Fundamental to multi-objective analysis is the concept of Pareto optimum, which is also known as the non-inferior solution. Qualitatively, a non-inferior solution of a multi-objective problem is one where any improvement of one objective function can be achieved only at the expense of degrading another. The Pareto optimal solutions and the associated trade-off values help decision-makers select an acceptable level of assurance and the corresponding cost. The decision-makers can make known their preferences with respect to the level of assurance against uncertainties in the model's prediction at the expense of a degradation (reduction) in the model's optimal solution. Some authors prefer the term 'best-compromise solution' to suggest that a non-inferior solution so identified is optimal only in terms of a particular set of value judgments.

In a single-objective model, the decision-makers' preferences are assumed to be known and expressed as a single-objective function $F(X)$ whose value for a particular solution vector X gives the utility of X for the decision maker. A general multi-objective analysis problem involves the selection of a k-dimensional vector of decision variables, $X = (x_1, x_2, \ldots x_k)$, which may represent project outputs, the allocations of scarce resources as inputs, and operation policies. There might be some relationships among variables which are expressed through constraints, $g_i(X) \leq 0$, $i = 1, 2, \ldots, m$. Each quantifiable objective can be described by a function $F_j(X)$ that should be either maximized or minimized. Using these notations, the vector optimization problem can be stated as:

$$\text{Maximize } Z(x) = \{f_1(x), f_2(x), \ldots, f_p(x)\} \qquad (5.83)$$
$$x \in X$$

subject to

$$g_i(x) \leq 0 \quad i=1,2,\ldots m \qquad (5.84)$$

where $Z(x)$ is the p-dimensional objective function, i.e., there are p objectives; and $g_i(x)$ are the constraints. The set of all feasible solutions X is defined as:

$$X = \{x \mid g_i(x) \leq 0, \; i = 1,2,\ldots, m\} \qquad (5.85)$$

Every feasible solution to the problem implies a value for each objective.

It is to be noted that without information about preferences which provide a rule for combining, the objectives may be incomparable. Such an incomplete ordering, which is characteristic of multi-objective planning problems, implies that in the absence of preference information an optimal solution cannot be found to the problem since all feasible solutions are not comparable. A complete ordering, which is characteristic of scalar (single-objective) optimization problems, can be obtained for a vector optimization only by introducing value judgments into the solution process.

Non-inferior Solutions

The trade-offs between separate and non-comparable objectives can be explored by following anyone of several approaches. If there are only a few objective functions and decision variables in the vector X, it may be easy to enumerate all feasible combinations of decision variables. If there are two objective functions and two decision variables, the values of decision variables and objective functions can be visualized by plotting them in the decision and objective space. An enveloping curve of these values can be easily drawn and this would indicate the efficient vectors X and the trade-offs that are possible among these efficient combinations. This concept is shown in Fig. 5.10.

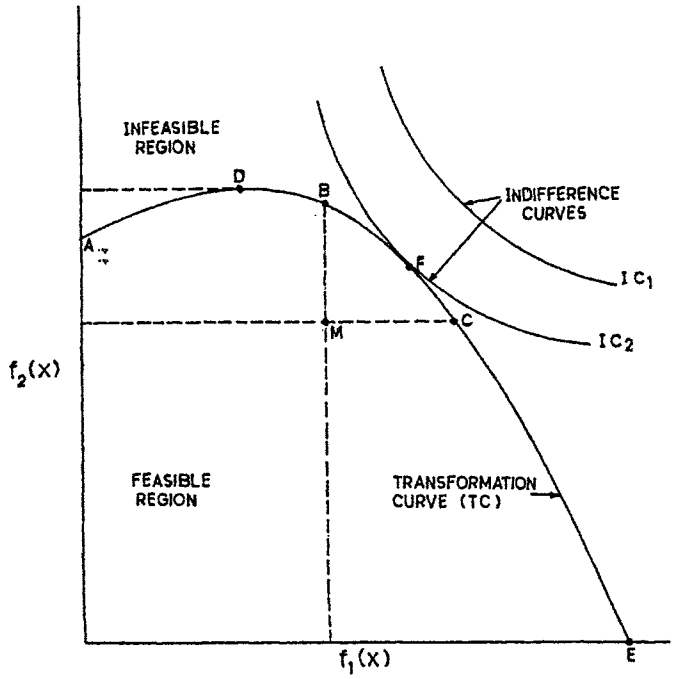

Fig. 5.10 Transformation curve and indifference curves for two-objective case.

A solution x^* of a multi objective problem is termed as non-inferior if and only if there does not exist another x' such that $f_i(x') \geq f_i(x^*)$, i=1,2,...n, with strict inequality holding for at least one *i*. The collection of all non-inferior solutions is referred to as the set of non-inferior solutions.

Each point within the feasible region in Fig. 5.10 represents a particular set of values for the decision variables in the vector X that satisfy the constraints. Each point on the transformation curve gives the maximum value of objective $F_1(X)$ given a particular value of objective function $F_2(X)$. Note that a point M located in the feasible region represents a solution that is not preferred because one can either get a better value of the objective function $F_2(X)$ by moving to point B or a better value of the objective function $F_1(X)$ by moving to point C. The trade-offs defined by certain combinations of feasible and efficient decision vectors are of interest in decision making. Each decision vector on the DBCE portion of the feasibility frontier is efficient because in that segment of the curve, there can be no increase in the value of one objective without a decrease in the value of the other objective. The solutions on the AD segment of the curve are inferior solutions because one can obtain a better combination of values of both objective functions (assuming that the values are to be maximized) elsewhere on the curve. Inferior solution vectors are of interest only if some of the objectives are to be minimized. When all the objectives are to be maximized, only the efficient solutions need be considered.

The goal in the planning stage is identification of plans which lie on the transformation curve since this is the set of efficient trade-offs. Any point on the transformation curve corresponds to a specific trade-off or marginal rate of substitution between the objectives. This rate equals the slope of the curve at that point. For example, the solutions D, B, and C in Fig. 5.10 correspond to three different marginal rates of substitution between objective functions $F_1(X)$ and $F_2(X)$.

The indifference curves IC_1, and IC_2 are the curves of equal preference of a decision-maker. It is possible that different individuals or groups may have different sets of indifference curves. The optimal plan for any particular policy maker is the plan on the objective transformation curve which achieves the highest level of preference of the decision-maker, for instance, point F in Fig. 5.10. The optimal trade-off or marginal rate of substitution is given by the slope of the transformation curve at that point F.

Example 5.4: The following two-objective two-decision variable problem by Cohon and Marks (1975) is used to illustrate the concepts of multi-objective optimization.

$$\text{Max } Z(x) = [Z_1(x), Z_2(x)] \tag{5.86}$$

$$Z_1(x) = 5x_1 - 2x_2 \tag{5.87}$$
$$Z_2(x) = -x_1 + 4x_2 \tag{5.88}$$

subject to

$$g_1(x): -x_1 + x_2 - 3 \leq 0 \tag{5.89a}$$
$$g_2(x): x_1 + x_2 - 8 \leq 0 \tag{5.89b}$$
$$g_3(x): x_1 - 6 \leq 0 \tag{5.89c}$$
$$g_4(x): x_2 - 4 \leq 0 \tag{5.89d}$$
$$g_5(x): x_1 \geq 0 \tag{5.89e}$$
$$g_6(x): x_2 \geq 0 \tag{5.89f}$$

Solution: In this example with two decision variables and two objectives, the feasible region in the objective space can be found by enumeration of all extreme points and computation of the values of each objective at each of these corner solutions. These points and the values of the objective functions are listed in the Table 5.1.

Table 5.1 Extreme points and values of objective functions.

$\bar{X}$	x_1	x_2	Z_1	Z_2
1	1	4	-3	15
2	4	4	12	12
3	6	2	26	2
4	6	0	30	-6
5	0	0	0	0
6	0	3	-6	12

The non-inferior set Z(X*) can be found by applying the definition of non-inferiority. The set of non-inferior solutions contains four extreme points: $Z(x^1)$, $Z(x^2)$, $Z(x^3)$ and $Z(x^4)$. The point $\bar{X}$ 2 (4,4) is the best-compromise solution. Note that this enumeration procedure is computationally feasible only for very small problems. The feasible region in the decision space and the set of non-inferior solutions are shown in Fig. 5.11. Fig. 5.12 shows the feasible region and the non-inferior set Z(X*) in the objective space Z(X).

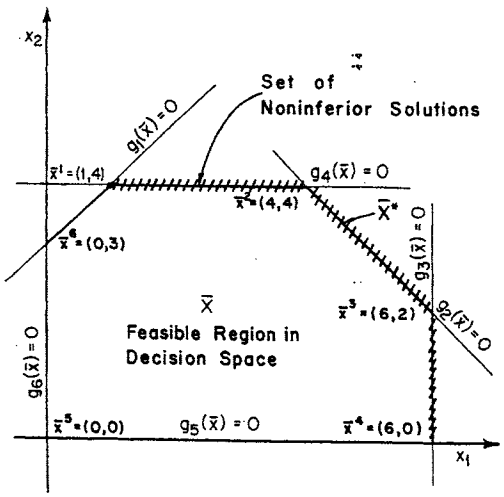

Fig. 5.11 The feasible region in decision space X and the set of non-inferior solutions X^* (after Cohon and Marks, 1975).

In the absence of preference information, no particular non-inferior solution can be identified as preferable to any other non-inferior solution. Of course, if preferences are

known as represented by an indifference surface, then one of the non-inferior solutions can be identified as the best-compromise solution. The term 'best-compromise solution' indicates that a non-inferior solution so identified is optimal only in terms of a particular set of value judgments.

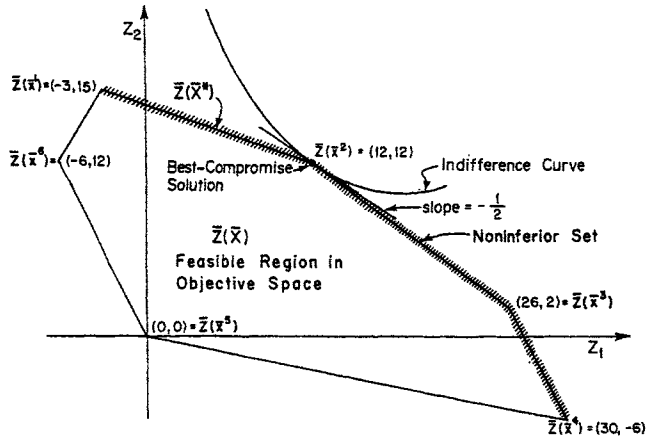

Fig. 5.12 The feasible region in objective space Z(X), the non-inferior set $Z(X^*)$, and the best-compromise solutions (after Cohon and Marks, 1975).

The use of enumeration method as a means of defining the feasibility frontier and the corresponding trade-offs between each efficient decision vector cannot be adopted as the number of variables and objectives increases. For these reasons, optimization techniques are usually suggested as a means of estimating feasible and efficient decision vectors X.

5.7.1 Classification of Multi-objective Optimization Techniques

The multi-objective analysis techniques can be classified in three groups (Cohon and Marks, 1975):

a. Generating techniques,
b. techniques based on prior articulation of preferences, and
c. techniques based on progressive articulation of preferences.

5.7.2 Generating Techniques

These were among the first techniques that were developed to solve multi-objective optimization problems. The purpose of these techniques is to identify the set of non-inferior solutions in the decision space as well as objective space within which the best-compromise solution will lie. The solution contains the maximum information from the model without preference information from the decision-maker. Two popular methods in this category are the weighting method and the constraint method.

Weighting Method

The weighting method was the first technique developed to solve multi-objective optimization problems. It is based on the premise that non-inferior solutions can be obtained by solving a scalar optimization problem in which the objective function is a weighted sum of the components of the original vector-valued objective function. Cohon and Marks (1975) found that the solution to the following problem is, in general, non-inferior:

$$Max \quad \sum_{k=1}^{p} w_k Z_k(x) \tag{5.90}$$

subject to

$$x \in X \tag{5.91}$$

where $w_k \geq 0$ for all k and strictly positive for at least one objective. Thus, the non-inferior solutions can be obtained through the use of an LP package (if all functions and constraints are linear or can be linearized) by parametrically varying the weight w_k in the objective function.

Constraint Approach

This method replaces (p – 1) objective functions with (p – 1) lower bound constraints:

$$\text{Maximize } Z_r(x) \tag{5.92}$$

subject to

$$Z_k(x) \geq L_k \quad \text{for all } k \neq r \tag{5.93}$$

where L_k is a lower bound on objective k; its value can be varied parametrically to evaluate the impact on the single objective function $Z_r(x)$. The non-inferior solution can be identified by solving p problems and taking a different objective function each time.

In the generating techniques, the trade-offs among the objectives are explicitly considered. These methods are intuitively appealing since results can be graphically presented. However, the essence can not be graphically captured for larger problems. The computational burden also grows tremendously with increase in the number of objective functions. Therefore, these techniques remain attractive only for small problems with two or three objective functions.

Techniques based on progressive articulation of preferences

As the name suggests, these techniques consist of finding a non-inferior solution, getting decision- maker's reaction to this solution, modify the problem and repeat the steps until a satisfactory solution is attained. The STEp Method (STEM) is one such method. The method requires construction of a pay off table. This method is not very suitable for water resources problems.

Techniques based on prior articulation of preferences
In these techniques, the information about preferences of the objectives is used to do partial ordering of the objectives and eliminate some non-inferior solutions. The aim is also to reduce the intensive computational load of generating techniques. The goal programming is one such technique. It is discussed in detail in Section 5.8.

The *Electre* method attempts to structure a partial ordering of alternatives which is stronger than the incomplete ordering implied by non-inferiority. In this method, a specific outranking relationship is developed for the set of non-inferior solutions. This method is not applicable to water resources problems since it is not computational attractive and tradeoffs are obscured by the analysis.

5.7.3 Surrogate Worth Trade-off Method

All the multi-objective optimization methods are directed towards evaluating, in some order of magnitude, the relative worth or utility of each of the objectives so that they can be treated as if there was only one composite objective. It might be called the commensurate approach since it is directed to the measurement and summation of the worth of all objectives in a common set of units. In case of water resources systems, the decision-maker may not be interested in evaluating the relative true worth of all the combinations of objectives, but rather in the evaluation of the relative true worth of changes that might be incurred in these objectives due to changes in the set of decisions. The interest is also in the values of those incremental changes for the decision sets that are already at a Pareto optimum. The order of magnitude of values upon which the trade-off of the decision-maker depends can be used as a substitute for the unknown true worth ratios of the marginal gain and losses between objectives. For this reason, these are called as surrogate worth. The term surrogate worth is defined to be a positive number whenever the true worth of Δf of the numerator in the trade-off ratio between any two objectives is considered to be higher than the true worth of the denominator of the ratio. It will be a negative number when the opposite is true and will be zero when the decision maker cannot distinguish between their relative worths.

The Surrogate Worth Trade-off (SWT) method was developed by Haimes and Hall (1974) who proposed that the choice of optimal weights should be made with the knowledge that trade-offs depend on the levels of objectives. The SWT method recognizes that the optimization theory is usually more concerned with the relative value of additional increments of the various non-commensurate objectives, at a given value of each objective function, than it is with their absolute values. Further, when the values of objective levels attained is known, it is easier for decision makers to assess the relative value of the trade-off of marginal increases and decreases between any two objectives than it is to assess their absolute average values. The distinguishing feature of the SWT method is the generation of "trade-off functions" which show the relationship between a weight on one objective (when another objective is the numeraire) and the values of that objective. A set of trade-off functions may be interpreted as a disaggregated non-inferior set, in which the objectives are considered in pairs.

If one objective function is considered primary and all others at minimum satisfying levels are considered constraints, the Lagrange multipliers related with the (p–1) objectives as constraints will be zero or nonzero. The Lagrange multiplier for a constraint that limits the optimum is nonzero. The nonzero Lagrange multipliers correspond to the non-inferior set of solutions whereas the zero Lagrange multipliers correspond to the inferior set of solutions. Furthermore, the set of nonzero Lagrange multipliers represents the set of trade-off ratios between the principal objective and each of the constraining objectives. These Lagrange multipliers are functions of the optimal level attained by the principal objective function as well as the level of all other objectives satisfied as equality (binding) constraints. Consequently, these Lagrange multipliers form a matrix of trade-off functions. It is assumed that the objective functions are differentiable functions of the right-hand-side levels of the constraints ($\in_j$).

Next, the worth ratios are selected. Since the worth ratios only represent relative worth (not the absolute level of worth of objectives), any surrogate ratio that varies monotonically with correct ratios will suffice. The distinguishing feature of the SWT method is the generation of 'trade-off functions' which show the relationship between a weight on one objective (when another objective is the numeraire) and the values of that objective. A set of trade-off functions may be interpreted as a disaggregated non-inferior set, in which the objectives are considered in pairs. The computational procedure is first to transfer the multiobjective problem into

$$\text{Maximize } Z_r(x) \tag{5.94}$$

subject to

$$x \in X \tag{5.95}$$

$$Z_k(x) \geq L_k \tag{5.96}$$

where L_k is the lower bound on the k^{th} objective for all $k \neq r$. One of the objectives expressed as a constraint, say, objective s, is then varied over k values of L_s, keeping the other objectives, all $k \neq r$, s, fixed at L_k. The problem in eqs. (5.87) to (5.89) is solved for each value of L_s, producing at most k non-inferior solutions. The dual variable associated with the constraint for the s^{th} objective when the r^{th} objective is in the objective function is T_{rs} which is the trade-off between objectives r and s:

$$T_{rs}(x) = df_r(x) / df_s(x) \tag{5.97}$$

where

$$df_r(x) = \sum_{k=1}^{p} \frac{\partial f_r(x)}{\partial x_k} dx_k \tag{5.98}$$

There are p values of T_{rs} generated by solving the modified problem with k values of L_s. The trade-off T_{rs}, taken as a function of $Z_s(x)$, is the previously referred trade-off function.

With generated values of T_{rs} and $Z_s(x)$, regression analysis is used to get the function $T_{rs}[Z_s(x)]$. Next, another of the $k \neq r$ objectives is selected as objective s. This

process is repeated until $T_{rk}[Z_k(x)]$ is generated for all $k \neq r$. The next step is to replace the r^{th} objective and repeat the procedure until all $T_{jk}[Z_k(x)]$ are generated for all $j = 1, 2, \ldots, p$, and all $k = 1, 2, \ldots, j-1, j+1, \ldots, p$. The result is a set of functions which relate the weights to the levels of the objectives (these can be displayed graphically). The number of trade-off functions is in general equal to p^2. However, since $T_{kk} = 1$ and $T_{jk} = 1/T_{kj}$, the number of trade-off functions is little less.

The trade-off functions give the analyst the required information to extract 'surrogate worth functions' W_{jk} from the decision maker. There is one surrogate worth function for every trade-off function; thus the intent of constructing W_{jk} is to attach values to the previously computed trade-offs. The W_{jk} functions are ordinal, varying between -10 and $+10$, with some arbitrary but predetermined value which indicates an acceptable ('optimal') trade-off. The set of optimal trade-offs or weights found by this method are then used to identify the best-compromise solution.

The SWT method provides more information than other methods, although less than the maximum information associated with the generating methods is available. The information supplied is not complete in the sense that the trade-off functions are generated between two objectives, assuming fixed values for all of the remaining objectives. Thus, the variation of trade-offs with the level of objectives is captured in only a limited sense.

The SWT method can be a powerful tool when there are difficulties in evaluating trade-offs. Its greatest utility is for problems with several objectives ($p > 3$), since it leads decision makers through a systematic comparison of objectives, two at a time. This approach may decrease the confusion associated with high dimensionality in the objective space when it is administered properly. Unfortunately, this method is vulnerable to its computational sensitivity to the number of objectives which is a generic characteristic of multi-objective solution techniques.

Cohon and Marks (1975) evaluated various multi-objective optimization techniques and suggested that when there are less than four objectives, a generating technique, such as the weighting method or constraint method should be used. When there are four or more objectives, a technique which restricts the size of the feasible region, such as the SWT method may be more suitable to use.

The purpose of multi-objective techniques is to assist the decision maker in estimating efficient alternative solutions and trade-offs that may be required to obtain an acceptable solution. The iterative process of proposing a solution and having it accepted or rejected by the decision maker is one means of focussing on trade-offs that are considered acceptable (need not be optimal) by the decision maker. But sometimes, the decision maker may not be adequately aware of the issues and implications and may not be able to clearly and consistently state his preferences.

5.8 GOAL PROGRAMMING

Most real world decision problems involve multiple and often conflicting goals which

cannot be maximized or minimized simultaneously within the constraints. In such a context, one has to think in terms of deviations from the optimum of individual objectives and thus try to achieve a balance between various objectives. When LP is used to solve decision problems, all constraints have equal importance and the optimum solution must satisfy all constraints. However, this assumption is not realistic and all constraints may not have equal importance. Such problems can be efficiently solved using the Goal Programming (GP) technique which can solve problems with a single or multiple goals. These goals may be non-commensurate meaning that they cannot be measured on the same-unit basis. Thus, there is a need to establish a hierarchy of importance among these conflicting goals so that low-order goals are considered only after the higher-order goals are satisfied or have reached the point beyond which no further improvements are desirable.

In many cases, the management does not try to 'optimize', instead it tries to 'satisfice'. An optimizer usually seeks the best possible outcome for a given objective, such as profit maximization in LP. A satisficer, on the other hand, attempts to achieve a satisfactory level of multiple objectives. GP is an appropriate technique for decision analysis. If management can provide an ordinal ranking of goals in terms of their contributions or importance to the organization and all relationships of model are linear, GP can be used to solve the problem. GP was devised for situations wherein the decision maker proposes to seek maximization or minimization of the weighted absolute deviations or departures from the individual optimum. The main aim of GP is to establish a specific numerical goal for each objective, formulate an objective function for each goal, and then seek a solution that minimizes the (weighted) sum of deviations of these objective functions from their respective goals.

There are three possible types of goals:

- A lower, one-sided goal sets a lower limit that should not be under-achieved (but exceeding the limit is acceptable).
- An upper, one-sided goal sets an upper limit that should not be exceeded (but falling under the limit is acceptable).
- A two-sided goal sets a specific target that should not be missed on either side.

In GP, instead of trying to maximize the objective criterion directly, the deviations among goals and what can be achieved within the given set of constraints are to be minimized. Such type of variable is represented in two dimensions, positive and negative deviations from each goal. The objective function becomes the minimization of these deviations based on the relative importance or priority assigned to them.

To understand the problem formulation of a goal program, it is necessary to define some notations. Any variable x_j, positive, zero or negative, can be expressed as the difference of two positive variables, i.e.,

$$x_j = x^+_j - x_j^- \tag{5.99}$$

where

$$x^{+}_{j} = x_j \text{ if } x_j \geq 0 \quad (5.100)$$
$$x^{+}_{j} = 0 \text{ if } x_j \leq 0 \quad (5.101)$$

Similarly,

$$x^{-}_{j} = 0 \text{ if } x_j \geq 0 \quad (5.102)$$
$$x^{-}_{j} = -x_j \text{ if } x_j \leq 0 \quad (5.103)$$

Thus, x^{+}_{j} and x^{-}_{j} are both positive and represent the positive and negative components (only components without sign). Clearly, the product $x^{+}_{j} * x^{-}_{j} = 0$ is always satisfied with one of the components or both components equal to zero. Furthermore,

$$|x_j| = x^{+}_{j} + x^{-}_{j} \quad (5.104)$$

The matrix used in GP is composed of two types of constraints: goal and non-goal. Each goal constraint may be assigned a positive or negative deviational variable or both. These variables are shown in Fig. 5.13. In this figure, the line labelled, "Goal", indicates the complete goal attainment. If more than the desired goal level is achieved, there is positive deviation from the goal (d^{+}). Under-achievement (d^{-}) means there will be a negative deviation (d^{-}) from the goal. An optimal solution is obtained when the sum of non-attainment of goals is minimized according to the priority structure.

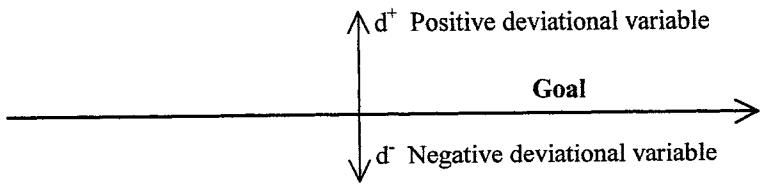

Fig. 5.13 A two dimensional goal space.

5.8.1 Goal Programming Model

Goal programming is a linear mathematical model in which the optimum attainment of multiple goals is sought within the given decision environment. The objective function is composed of either a pair or a single deviational variable for each goal constraint. If overachievement is acceptable, the positive deviation (d_i^{+}) can be eliminated from the objective function. On the other hand, if underachievement is satisfactory the negative deviation (d_i^{-}) should not be included. The exact achievement of a goal requires both negative and positive deviations to be represented in the objective function to achieve the ordinal solution.

An optimal solution is obtained when the sum of non-attainment of goals is minimized according to the priority structure established by the decision-maker. To achieve the goals according to their importance, GP provides a means by which the negative or

positive deviations about the goal may be ranked according to an ordinal priority ranking scale in order of preference of each goal level. Weights are assigned to priority factors for minimizing the deviational variables. They are only assigned to deviational variables which have been assigned the same priority levels. The deviational variables and the ordinal priority factors are always present in each objective function.

A GP problem in standard format is

$$\text{Minimize} \quad Z = d^- + d^+ \tag{5.105}$$

$$\text{subject to} \quad BX + d^- - d^+ = h \tag{5.106}$$

$$AX \leq b \tag{5.107}$$

$$X, d^-, d^+ \geq 0 \tag{5.108}$$

where B is a (1 x n) row vector of objective function coefficients, X is an (n x 1) column vector of real variables, b is an (m x 1) column vector of right hand side constant, A is an (m x n) matrix of technological coefficients, d^+, d^- are deviational variables in positive and negative directions, and h is the goal level set by the decision maker. The left hand side of a goal constraint can be less than, greater than, or equal to type. Table 5.2 shows how these three possibilities and how they are handled in GP formulations.

Table 5.2 GP Model formulations and inclusion of the deviational variable in the objective function.

S.N.	Goal or constraint type	Processed goal or constraint	Deviational variable to be minimized in the objective function
1.	$f_i(X) \leq b_i$	$f_i(X) + d_i^- - d_i^+ = b_i$	d_i^+
2.	$f_i(X) \geq b_i$	$f_i(X) + d_i^- - d_i^+ = b_i$	d_i^-
3.	$f_i(X) = b_i$	$f_i(X) + d_i^- - d_i^+ = b_i$	$d_i^- + d_i^+$

While solving the GP problems, the following points must be considered:

(i) In GP, the objective is to minimize the total non-attainment of goals. This is achieved by minimizing deviational variables through the use of preemptive priority factors and differential weights. There is no profit maximization or cost minimization per se in the objective function. Therefore, preemptive factors and differential weights take the place of C_j used in LP.

(ii) The objective function is expressed by assigning priority factors to certain variables. These pre-emptive priority factors are mutli-dimensional, since they are ordinal rather than cardinal values.

(iii) The GP solution will allow some lower priority goals to go unsatisfied in order that higher priority goals, which may conflict with lower priority ones, achieve the targets. The ranking of deviational variables is the most important step in formulating a GP problem. The highest priority factor is assigned to deviational

variables of the most important goal. The lowest priority factor is assigned to deviational variable of the least important goal. Thus, the low order goals are considered only after higher order goals are achieved. The priority factors have the relationship of $P_j >>> P_{j+1}$ which means that P_j always takes priority over P_{j+1}.

The basic assumption in formulating the initial tableau of GP is identical to that of LP. One can assume the initial solution to be at the origin where values of all decision variables are zero. Considering linear problems, one can visualize situations of cost functions having different cost slopes for positive and negative deviations.

Priority Ranking

After assigning the deviational variables, the next step is to assign the ordinal priority factors. The negative and/or positive deviations about the goal are ranked according to an ordinal priority ranking scale in order of preference of each goal level.

If deviations from the specified target have different cost slopes then the objective function can be written as

$$\textit{Minimize} \quad z = \sum_{k=1}^{K} (w_k^+ d_k^+ + w_k^- d_k^-) \tag{5.109}$$

subject to a set of constraints. Here w^+ and w^- represent the cost slopes for positive and negative deviations.

Based on the priority levels, GP can be classified as:

(i) preemptive goal programming, and
(ii) non-preemptive goal programming.

When there is a hierarchy of priority levels for goals in which one or more of the goals are far more important than others, it falls under preemptive GP. When dealing with goals on the same priority level, the approach is non-preemptive GP. It is possible to specify the order in which objectives are to be satisfied in a lexicographic context in which case the weights are referred to as preemptive weights. In this case the objective function is generally written as

$$Z = [p_1 h_1 (d^+, d^-), p_2 h_2 (d^+, d^-), \ldots, p_k h_k (d^+, d^-)] \tag{5.110}$$

One sided goals: In this case g_k for the k^{th} goal represents the bound on that goal rather than a specific amount that should be attained if possible. If g_k is a lower bound goal, then

$$\sum_{j=1}^{n} c_{jk} x_j \geq g_k \tag{5.111}$$

In this case any attainment over g_k is fine but any deviation below is to be avoided if feasible. The change that this causes in the formulation of the objective function is that

only negative deviations are incorporated into the objective function. Both type of deviations are still there in the constraints as before as both can still occur. Similarly, only positive deviations are incorporated in the objective functions in the case of upper bounds.

Non-commensurable objectives

Suppose there are k objectives of the following type which cannot be combined into a single objective function:

$$z_1 = \sum_{j=1}^{n} c_{j1} x_j \tag{5.112}$$

The overall objective function for the model becomes

$$\text{Maximize } z = \text{Minimize } [z_1, z_2, \ldots, z_k] \tag{5.113}$$

An optimal solution to this problem is the one that makes the smallest z_k as large as possible. This model is not in the LP format. However, it is equivalent to the following LP model:

$$\text{Maximize } Z = z$$

subject to

$$\sum_{j=1}^{n} c_{j1} x_j - z \geq 0 \qquad \text{for } k = 1,2,\ldots,k \tag{5.114}$$

$$x_j \geq 0 \quad \text{for } j = 1,2,\ldots,n$$

and any other constraints in the original model. The maximum feasible value of the new variable z in this model must equal the smallest $z_k = \Sigma\, c_{jk} x_j$, so an optimal solution for $(x_1, x_2, \ldots, x_n)$ will make this smallest z_k as large as possible.

When the objectives are to be minimized rather than maximized, the overall objective function for the original model would change to

$$\text{Min } Z = \max \{z_1, z_2, \ldots, z_k\} \tag{5.115}$$

and the corresponding LP model is

$$\text{Min } Z = z \tag{5.116}$$

$$\sum_{j=1}^{n} c_{jk} x_j - z \geq 0 \qquad \text{for } k = 1,2,\ldots,k \tag{5.117}$$

$$x_j \geq 0 \qquad \text{for } j = 1,2,\ldots,n \tag{5.118}$$

and any other constraints in the original model.

Example 5.5: Consider two objectives of maximizing food production in two different regions (x_1, x_2 are the numbers of projects to be undertaken):

$$\text{Objective 1:} \quad \text{Max } z_1 = 2000\, x_1 \tag{5.119a}$$
$$\text{Objective 2:} \quad \text{Max } z_2 = 3000\, x_2 \tag{5.119b}$$

subject to

$$x_2 \le 4 \quad \text{Equipment constraint} \tag{5.120a}$$
$$x_1 + 2x_2 \le 10 \quad \text{Experts constraint} \tag{5.120b}$$
$$60x_1 + 20x_2 \le 300 \quad \text{Money constraint} \tag{5.120c}$$
$$x_1, x_2 \ge 0$$

Solution: The equivalent LP problem is

$$\text{Max } Z = z \tag{5.121}$$

subject to

$$2000x_1 - z \ge 0 \tag{5.122a}$$
$$3000x_2 - z \ge 0 \tag{5.122b}$$
$$x_2 \le 4 \tag{5.122c}$$
$$x_1 + 2x_2 \le 10 \tag{5.122d}$$
$$60x_1 + 20x_2 \le 300 \tag{5.122e}$$
$$x_1 \ge 0,\ x_2 \ge 0,\ z \ge 0$$

The solution to this problem is

$x_1 = 45/11$, so $z_1 = 8182$.
$x_2 = 30/11$, so $z_2 = 8182$.
Hence, $z = 8182$.

Example 5.6: A farmer produces two crops: rice and wheat. The farmer has a production capacity of 40 ton of crops. Because of limited sale opportunity, he can sell a maximum of 24 tons of rice and 30 tons of wheat. The gross margin from the sale of 1 ton rice is 80 units and for wheat, it is 40 units. Find the optimal production.

Solution: Let the optimal production of rice and wheat be x_1 and x_2 tons. If the farmer had only the single goal of profit maximization, the decision problem could be easily formulated as an LP problem, as illustrated below:

$$\text{Maximize} \quad Z = 80x_1 + 40x_2 \tag{5.123}$$

subject to

$$x_1 + x_2 \le 40 \tag{5.124a}$$
$$x_1 \le 24 \tag{5.124b}$$
$$x_2 \le 30 \tag{5.124c}$$
$$x_1, x_2 \ge 0 \tag{5.124d}$$

The optimum solution is $x_1 = 24$, $x_2 = 16$, and the total profit $Z = 2560$ units.

This profit maximization problem can also be solved by the GP approach:

$$\text{Minimize } Z = p_1(d^+_1 + d_2^+ + d^+_3) + p_2 d^-_4 \tag{5.125}$$

subject to

$$x_1 + x_2 + d^-_1 - d^+_1 = 40 \tag{5.126a}$$
$$x_1 + d^-_2 - d^+_2 = 24 \tag{5.126b}$$
$$x_2 + d^-_3 - d^+_3 = 30 \tag{5.126c}$$
$$80x_1 + 40x_2 + d^-_4 - d^+_4 = 10000 \tag{5.126d}$$

The objective function of the GP formulation indicates that the highest priority (p_1) is assigned to the minimization of d^+_i. Since in the LP model the first three constraints are "less than or equal to" inequalities, the solution must be within the region that satisfies these three constraints. By assigning the highest preemptive priority factor to the minimization of positive deviations (d^+_i) in the first three constraints, the same restrictions are met. The second priority factor of GP is assigned to the minimization of d^-_4, i.e., minimization of the underachievement of some high profit goal. Arbitrarily, a limit of 10000 units is set, knowing that such a high profit will never be achieved. The minimization of underachievement of the profit goal will drive the values of x_1 and x_2, within the area of feasible solution, as close to 10000 as possible. The solution of the GP problem is the same as the LP solution: $x_1 = 24$, $x_2 = 16$, and $Z = 2560$ units.

Example 5.7: Consider that the farmer in Example 5.6 has set the following goals as arranged in order of their importance:

1. He wants to avoid any underutilization of the production capacity.
2. He wants to sell his crop as much as possible. Since the gross margin from the sale of rice is twice the amount from wheat, he has twice as much desire to sell more rice as for wheat.
3. The farmer wants to minimize the extra production from the farm.

How each of these goals can be incorporated in the model?

Solution: The consideration of each of these goals is discussed below.

Production Capacity: Since the goal regarding extra production was given the lowest priority, it is quite possible that the production of both rice and wheat may be more than 40 tons. The production capacity restriction can be expressed as:

$$x_1 + x_2 + d^-_1 - d^+_1 = 40 \tag{5.127}$$

where d^-_1 and d^+_1 represent the under- and over-utilization of the production capacity. Depending on the actual utilization, at least one of these variables will be zero.

Sales Capacity: Due to the limited sale opportunity, the farmer can sell a maximum of 24

tons of rice and 30 tons of wheat. One can omit positive deviations from the constraints which can be written as:

$$x_1 + d_2^- = 24 \quad (5.128)$$
$$x_2 + d_3^- = 30 \quad (5.129)$$

where d^-_2 and d^-_3 represent the underachievement of sale goals for rice and wheat, respectively.

In addition to the variables and constraints stated above, the following pre-emptive priority factors are defined in order to pursue the stated goals:

p_1 : The highest priority is assigned to minimize the underutilization of production capacity (i.e., d^-_1).

p_2 : The second priority factor is assigned to minimizing the underachievement of sale goals (i.e., d^-_2 and d^-_3).

However, the farmer also wants to assign differential weights (not pre-emptive but ordinary numerical weights) to the achievement of the sale goal according to the gross margin ratio between rice and wheat. Hence, the farmer assigns twice the weights to d^-_2 as assigned to d^-_3.

p_3 : The lowest priority factor is assigned to minimizing the extra production (i.e., d^+_1).

The objective function can now be stated as:

$$\text{Minimize } Z = p_1d^-_1 + 2p_2d^-_2 + p_2d^-_3 + p_3d^+_1 \quad (5.130)$$

The objective is to minimize deviations from the goals. The value of the deviational variable associated with the highest preemptive priority must be minimized first to the fullest possible extent. When no further improvement is possible or desired at the highest goal, the next attempt is to minimize the value of the deviational variables associated with the next highest priority factor, and so forth. Therefore, the complete model is:

$$\text{Min } Z = p_1d^-_1 + 2p_2d^-_2 + p_2d^-_3 + p_3d^+_1 \quad (5.131)$$

subject to
$$x_1 + x_2 + d^-_1 - d^+_1 = 40$$
$$x_1 + d^-_2 = 24$$
$$x_2 + d^-_3 = 30$$
$$x_1, x_2, d^-_1, d^-_2, d^-_3, d^+_1, d^+_2, d^+_3 \geq 0$$

Application Areas of GP

An important property of GP is its ability to handle multiple incompatible goals according to their importance. A GP model performs three types of analysis:

(1) It determines the degree of attainment of defined goals with given resources;
(2) It determines the input requirements to achieve a set of goals;
(3) It provides the optimum solution under the varying input and goal structures.

The biggest advantage of GP is its flexibility, which allows model simulation with numerous variations of constraints and goal priorities. GP can be applied to almost unlimited managerial and administrative decision areas. Allocation planning and scheduling, and policy analysis are the most readily applicable areas of GP. Some references for further study are Ignizio (1976), Lee (1972), and Loganathan and Bhattacharya (1990).

5.9 SIMULATION

Simulation is the process of duplicating the behavior of an existing or proposed system. It consists of designing a model of the system and conducting experiments with this model either for better understanding of the functioning of the system or for evaluating various strategies for its management. The essence of simulation is to reproduce the behavior of the system in every important aspect to learn how the system will respond to conditions that may be imposed on it or that may occur in the future. The main advantage of simulation models lies in their ability to accurately describe the reality. If a simulation model can be developed and is shown to represent a prototype system, it can provide insight about how the real system might perform over time under varying conditions. Thus, proposed configurations of projects can be evaluated to judge whether their performance would be adequate or not before investments are made. In a like manner, operating policies can be tested before they are implemented in actual control situations. Hufmschmidt and Fiering (1966) describe the simulation technique for design of water resources systems. James and Lee (1971) have noted that simulation is the most powerful tool to study complex systems.

Usually, the structure or behavior of the system being simulated is so complex that its analytical expression is not possible. A simulation model of a water resource system duplicates its operation with a defined operational policy, using the parameters of physical and control structures, time series of flows, demands, and the variables describing water quality, etc. The evaluation of the design parameters or operation policy is through the objective function (flow or demand related measures or economic indices) or some measure of reliability. Since simulation models do not use an explicit analytical procedure for determination of the best combination of the controlling variables, it is necessary to proceed by trial and error or follow a strategy of parameter sampling.

Since models are abstractions of reality, they usually do not describe all the features that are encompassed by a real-world situation. Only those aspects of the system that are relevant to the objective of the study are modeled so that solution is obtained at a reasonable cost and within a prescribed time frame. If the simulation model has to reproduce all the complexities of the prototype, it will be as complex as the prototype. Therefore, the model builder should attempt to model the detailed functioning of individual components to the necessary extent so as to meet the overall accuracy requirements while not making it unnecessarily complicated. To illustrate, if the objective is the design of a

large storage reservoir for irrigation and municipal water supply, it is quite unnecessary to model the complete runoff process. On the other hand, a monthly flow-generation model is entirely unsuited for modeling the peak discharges. An important aspect of model building in the context of simulation is to find the best permissible simplifications. When, for example, should the engineer responsible to issue flood forecasts use a simple routing model and when he should employ a dynamic wave model, using the complete St. Venant equations? The difference in efforts and computer time for the two methods is very large. The main reasons for searching for a simple model may be a lack or low quality of data. For example, consider that there are only a few rainfall and discharge data stations in a large catchment. In this scenario, there is no justification to set-up a detailed model which requires huge data, long time to calibrate and run, and skilled manpower.

The main advantage of simulation models lies in their ability to closely describe the reality. If a simulation model can be developed and is shown to represent the prototype system realistically, it can provide insight about how the real system might perform over time under varying conditions. Thus, proposed configurations of projects can be evaluated to judge whether their performance would be adequate or not before investments are made. In a like manner, operating policies can be tested before they are implemented in actual control situations. Simulation is widely believed to be the most powerful tool to study complex systems.

5.9.1 Classification of Simulation Models

Simulation models may be physical (a scale model of a spillway operated in a hydraulics laboratory), analog (a system of electrical components, resistors and capacitors, arranged to act as analog of pipe resistances and storage elements), or mathematical (a compilation of equations and logical statements that represent the actions of a system's elements). Mathematical simulation models are very useful and popular in the field of water resources. The platform that is used to operate models of this type is the digital computer. Only simulation models of this type are discussed here.

Simulation models can also be classified as static or dynamic. Dynamic models take into account the changing parameters of the system (structures and facilities) and the variations in their operation. The simulation model of a water resource system is considered a dynamic model if the operational policy can be dynamically changed with time and if such changes correspond to the system demands and related changes in system parameters. These are assumed as fixed in static models. The development and application of dynamic models is a more involved exercise and often static models give acceptable results.

Many hydrological variables are stochastic in character. Deterministic and stochastic simulation models are distinguished by the way this stochasticity is accounted for. A time-series of gauged flows represents a sample of the stochastic process. Under certain conditions, deterministic simulation models can be used with confidence. For example, if measured monthly flows for a period of 40 years are used as input in a study, a deterministic model may be adequate. If the process is stationary, the sample can be considered a reasonably good characterization of the stochastic process. Note that as the

inflow series will not repeat exactly, the future performance of the system will differ from that obtained by the model.

Two principal methods are used to account for stochastic properties in the simulation model:

- The synthetic flows generated by methods of stochastic hydrology are used as inputs, or
- The simulation model is combined with other models that permit a stochastic solution (e.g., the chance-constrained model, etc).

Besides the above, the degree of aggregation can be used to classify simulation models. A model with a high level of details is suitable to investigate the operation of an existing water resource system. The typical objective of applying such a model is the improvement of the system operation. On the contrary, simulation models with much more aggregated data are appropriate for design of water resource systems. Jacoby and Loucks (1972) were among the first to use this strategy to investigate a system in combination with analytical optimization models. Aggregation simplifies modeling but should not introduce significant deviations from reality. Simplification can also be achieved by neglecting variables that do not impart a decisive effect on the system behavior. If the output is not sensitive to the variation of certain variables, these can be considered as constants.

5.9.2 Monte Carlo Simulation

Design of real world systems is generally based on observed historical data. For example, the observed streamflow data are used in sizing a reservoir, historical traffic data is used in design of highways, observed data are used in design of customer services, etc. However, frequently the historical records are not long enough and the observed pattern of data is not likely to repeat exactly. The performance of a system critically depends on the extreme values of input variables and the historical data may not contain the entire range of input variables. There are many instances when a flood with peak value exceeding the historical records entered a reservoir.

An important conclusion from the above is that one does not get a complete picture of the system performance and risks involved when historical data are used in evaluation. Thus, for instance, the planner cannot determine the risks of a water supply system failing to meet the demands during its economic life because this requires a very large sample of data which are not commonly available.

For many systems, some or all inputs are random, system parameters are random, initial conditions may be random, and boundary condition(s) may also be random in nature. The probabilistic properties of these are known. For such systems, simulation experiments are conducted using a set of inputs which are synthetically (artificially) generated. While generating the inputs, it is ensured that the statistical properties of the random variables are preserved. Each simulation experiment with a particular set of inputs gives an answer. When many such experiments are conducted with different sets of inputs, a set of answers is

obtained. These are statistically analyzed to understand or forecast the behavior of the system. This approach is known as Monte Carlo simulation, and using it, planners get better insight of the working of the system and can determine the risk of failures, e.g., chances of a reservoir running dry or a customer service center failing to provide services within promised time.

The main advantages of Monte Carlo simulation are that the system, its inputs, outputs, and parameters can be easily described. All the critical parameters of the system can be included in its description. The other advantages include saving in time and expenses. It is important to remember that the synthetically generated data are no substitute of the actual observed data but this is a useful pragmatic tool which allows the analyst to extract detailed information from the available data.

Random number generation is an important part of Monte Carlo analysis. In the early days of mathematical simulation, mechanical means were employed to generate random numbers. The techniques that were used to generate random numbers were drawing cards from a pack, drawing numbered balls from a vessel, reading numbers from a telephone directory, etc. Printed tables of random numbers were also in use for quite some time. The Monte Carlo techniques have got this name because roulette wheels similar to those in use at Monte Carlo were used to generate random numbers. The current approach is to use a computer to generate random numbers and this is discussed in Appendix 5A.

5.9.3 Time Management in Simulation

The modeling of a continuous process by a discrete model requires the assumption that the continuous changes during a defined period take place instantaneously at the end or at the beginning of the period. The decision-making process in water resource systems is discrete; simulation models are also discrete models. The real-life process, however, is continuous. Therefore, the time step size is an important aspect of the model and should be chosen carefully. This choice depends on the degree of aggregation and the time variability of the inputs.

Event scanning and periodic scanning are two common ways of time management in simulation models. In the event scanning approach, the clock is advanced by the amount of time which is required for the occurrence of the next event. In many natural phenomena, the periods of high activity are separated by long periods in which the system lies inactive. This approach is suitable for these types of situations and can save computational time. Note that it requires some scheme to determine the time when the events take place.

In periodic or fixed time scanning, the whole simulation horizon is divided into smaller time periods. The system time is incremented by the predetermined step and simulation is performed. This procedure is repeated till the end of the period of analysis. A judicious choice of time increment is necessary in the periodic scanning approach. This increment should be small enough so that no significant information is lost. The fixed-time scanning approach is commonly used in water resources problems.

A simulation model of a water resource system commonly mimics the behavior of the system in discrete time steps using arithmetical and logical procedures (algorithms) by a series of "snapshots." These algorithms try to reproduce the way inputs to the system are acted upon and transformed into outputs. The size of this time step depends on the purpose of the study. In planning studies dealing with irrigation, hydroelectric power generation, water supply purposes, minimum flow maintenance, etc., monthly or 10-daily periods are generally used. These periods make it possible to reflect the seasonal variability of demand and hydrological inputs. During floods, however, the condition of a surface water system changes rapidly and, therefore, weekly or 10-daily time steps are too long to give any meaningful result. Hence, for the purpose of flood related simulation studies, multi-hourly or shorter time steps are used.

5.9.4 Design of Sampling Strategy

Sampling strategy is a combination of methods whose basic object is to find the nature of the response surface in a problem and the location of extreme points. A good strategy should minimize the computational efforts and save time. The first few samples are exploratory with the aim to know the nature of the response surface and to find out the kind of variation it has -- whether it is smooth or there are jagged edges. This data helps in identification of the region of response surface which requires further detailed examination. The range of decision variables over which further experiments need to be conducted is also identified. Next, extensive sampling is carried out in the identified region to get the desired information. The sampling methods are diagrammatically shown in Fig. 5.14.

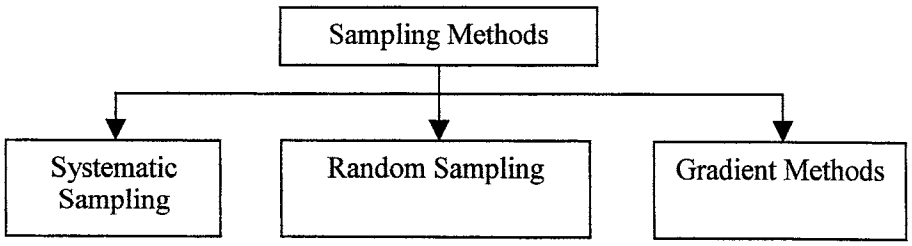

Fig. 5.14 Sampling methods used in simulation.

The basic sampling strategies when using simulation models are: systematic sampling, random sampling and the gradient methods (e.g., the steepest ascent method). In systematic sampling, a survey of the response surface is carried out at points systematically distributed over the range of input variables. The method of uniform grid is suitable when a large number of alternatives are to be examined. In this method, a grid of uniform spacing is drawn in the feasible region and sampling is carried out at various nodes of the grid. The size of the grid is decided keeping in view the number and range of variables. The objective function will have to be evaluated at many points if the grid size is small. In case a large grid size is chosen, the chances of missing the extreme points is higher particularly if the response surface is jagged.

An alternative sampling scheme which is known as single factor method involves experiments in which one variable is changed at a time. The process is repeated and each variable is evaluated over its range. This strategy is suitable when the variables involved are independent but it requires that objective function evaluation at a large number of points to gather desired information about the response surface. An improvement of this method consists of changing two or three variables at a time to find out the behavior of the response surface and further experiments are directed in the direction of the desired change. If the number of variables is small and the variables that have a major influence on the objective function can be easily identified, this method is useful. As the number of combinations in systematic sampling could be large, a combination of this sampling with heuristic sampling is often used. Obviously inefficient combinations of parameters are excluded in advance. This method is effective in situations where the system behavior can be guessed beforehand.

The random sampling method is suitable at the beginning of computation when the system behavior is completely unknown. This strategy is helpful in preliminary identification of the zone of response surface which requires further refinement. Some points are chosen at random from a systematic grid and the objective function is computed at these points. Note that as the number of sampling points increases, better information of response surface is obtained and the chances of finding the extreme points also increases.

In a detailed investigation of the behavior of a system and its response to input parameters, the method of steepest ascent is often used. By small variations of parameters, the direction of steepest ascent on the response surface (i.e., the surface formed by the values of the objective function in n-dimensional space) is identified. The parameters are changed in the direction of the steepest ascent and a new value of the objective function is computed. The algorithm is repeated until the region of the optimum is reached.

5.9.5 Steps in Simulation Modeling

The following are the steps in development and application of a simulation model:

a) define the problem;
b) describe the water resource system and its hydrological relationships;
c) decide the model structure, input, and output;
d) design the model;
e) test the model, if it is not suitable, go to step c); and
f) apply the model to the problem.

After the model of a system is developed, experiments are conducted with it to verify the analytical results or to answer the question "*WHAT IF*?" The simulation models are much helpful in understanding the consequences or implications of changing one or more of the decision variables.

In application of simulation models, it is often advantageous to subdivide a large problem into subsystems by any one of the following methods:

1. The first approach is based on the flow of water particles. Subsystems are defined on

the basis of the origin of flow or its modification. If some groups of elements show little relationships to other elements but the interactions inside these groups are relatively strong and numerous, a subsystem consisting of this group of elements can be formulated. For example, the delineation of a basin in various sub-basins is based on the flow paths of water.

2. The second method of identifying subsystems is the functional approach. This method is adopted if the flow approach is not suitable due to typical needs and requirements of the system functions. For example, consider the simulation of operation of a multi-purpose reservoir for conservation and flood control purposes. The length of the time period for conservation simulation is usually a month or a week. No flood routing calculations are performed when such a long time step is used. However, to simulate flood control regulation, a multi-hourly time step is used. A short-term operation has some unique characteristics. Flood routing is very important and the model may also have a component for forecasting.

5.9.6 Inputs to Simulation Models

The requirement of input data depends on the objective of using a simulation model. Broadly, there are two main types of input data for the simulation model: (1) the variables describing the system, its configuration and parameters of structures, e.g., reservoir capacities, characteristics of outlets, and aquifer properties; and (2) time series of flows, demands of the system (either existing or projected), evaporation depths, operation policy, etc. The observed hydrological long time-series data will require pre-processing before it can be used in the model. For instance, the model may require the average precipitation in the catchment and this will require determination of station weights if the Thiessen Polygon method is used. The periods of observation often do not coincide for all the stations relevant to the system and the stations may not be located at points where the data are needed by the model. If the records are interrupted and the observation periods are shorter in some stations, the records should be completed by filling-in the missing values. The procedures for processing of hydrometeorological data are described in Chapter 2 as well as in many books such as Singh (1992).

The parameters of the system include the storage capacities of reservoirs, carrying capacities of canals, river, the acceptable minimum releases from reservoirs, the diversion of flows within the basin or inter-basin diversion, the characteristics of aquifer, the requirements for water quality, etc. The input data also include the requirements at the demand centres and diversion points.

The economic inputs include costs of storage, irrigation diversion, power plants, and recreation facilities, etc. The economic output values include costs of operation, maintenance and replacement of facilities, benefits associated with water supply for municipal, industrial, agricultural use, hydropower, recreation, reduction of flood damage and low-flow augmentation. The broad data requirements for analysis of water resources systems have been discussed in Chapters 2 and 9.

5.9.7 Outputs of Simulation Models

The output of a simulation model can be in terms of hydrological and/or economic variables. The hydrologic results comprise values of various design variables, operation policies, working tables, hydrographs at important locations, etc. The design variables are, for example, the optimum storage capacity of the reservoir. Typical hydrological variables for a surface water project comprise, for various time periods, the reservoir storage, inflows, demands, release, spill, power generated, deficit in release for various purposes. Reliability indices are evaluated in terms of volume or time deficits.

Display or print of input data in the initial model runs helps in detecting the input errors. A good simulation model should allow the user to control how much detailed output he wants. The user should carefully decide what variables and how much detailed results are needed; unnecessary and long tables are difficult to study and the useful information may remain unnoticed. The interpretation of results is simpler if these are also available graphically.

Water resource systems at certain periods fail to meet target outputs. Therefore, the simulation model outputs should include the deficits for various targets, energy deficits; reservoir level fluctuations; and the duration, magnitude, and total volume of the deficits. Reliability indices are helpful in assessing the system performance. The economic consequences of these deficits are evaluated by an economic loss function. The economic output data typically include the time-series of benefits and costs, cash flows and benefit/cost ratio.

The interpretation of output is an important step in the analysis. A careful browse is necessary for proper choice of variables to be changed for the next step of computation. The sensitivity of system performance to various decision variables can be ascertained through a few model runs. To verify the results of simulation models, "common sense" based on prior experience is often used. If the results of simulation model fall outside the expected limits, detailed print out containing, for example, the reservoir contents at the end of each month, releases from reservoirs, flow at important points of the system, can be taken. This will help to locate possible errors, if any.

5.9.8 Pros and Cons of Simulation Models

Simulation and other types of models can be valuable aids in decision making but their advantages and disadvantages must be weighed for the circumstances of concern. Some advantages of the use of simulation models include:

(a) they impose a logic and structure to analyses;
(b) they provide insights into system behavior;
(c) their structure is ideally suited to experimental work;
(d) they may be designed to accommodate many options;
(e) projections into the future are facilitated with their use; and
(f) they can aid in communications between analysts and policy makers.

Some common problems associated with simulation models are:

(a) some simulation models are data hungry and their data requirements cannot be easily met;
(b) the handling of intangibles is difficult in simulation models;
(c) there are high costs of development and/or use of simulation models, particularly the complex ones;
(d) trained and experienced man-power is needed to interpret their output; and
(e) the users may not always be ready to accept the models and their results.

5.10 CLOSURE

Since optimization and simulation models are used for a common end, many times the analyst is in a dilemma as to which type of model to use. Simulation models are effective tools for problems, such as evaluating the performance of various configurations of reservoir and hydroelectric power plant capacities, water use allocation targets, operating policies, and the like. But they are not a very effective means for choosing or defining the best configurations or combinations of capacities, targets, and policies. For these problems, optimization models have proven to be effective, at least for eliminating the worst solutions from further consideration.

Due to several reasons, optimization models cannot determine the exact optimal solution to water resource management problems. The first reason is that their solution algorithms often require some simplifying assumptions which may not be realistic and may be limiting. Nonlinear cost, benefit and loss functions, and nonlinear expressions required for defining evaporation losses, hydroelectric energy production, and some combinations of flood control alternatives are often approximated by piecewise linear functions. The analyst should be aware of the implications of these limitations.

The modelling of even relatively small surface water systems may require a large number of mathematical statements, constraints, and variables. Even though very high capability desk-top computers are easily available nowadays, software, trained man-power, and finances may sometimes be a limiting factor. It is often necessary to make simplifying assumptions to reduce the size of the problem to manageable limits.

A third and perhaps the most limiting aspect of optimization models is the conceptual difficulty associated with the quantification and specification of a criterion for evaluating each possible management alternative. Public policy objectives are a mixture of monetary and nonmonetary goals which make quantification difficult. Furthermore, during the preliminary planning phase of water resources projects, the goals and objectives are not always clear and there may not be an agreement among decision makers on what these goals should be and the extent to which they are to be satisfied.

There are other important limitations related to the quantification of hydrologic, technologic and economic uncertainties, and inaccurate/incomplete data. All of these

mathematical, computational, conceptual, and data limitations restrict the use of optimization models to preliminary screening. Those alternative investment and operating policies that survive the preliminary screening process should be further analyzed, evaluated, and improved using simulation or other techniques where appropriate. While simulation models share many of the same conceptual and data limitations, they are far less restrictive mathematically and computationally. Hence, they are usually better suited for evaluating more precisely the alternatives defined by the optimization or preliminary screening models.

To take the full advantage of the strengths of systems techniques, it is necessary that due care and attention is paid to problem formulation. The objective(s) and constraints should be carefully designed. Usually, great effort of the analyst goes in to reduce the system to a manageable representation without destroying its essential features and relationships. After listing all the alternatives in the beginning, those that are clearly inferior may be discarded so that the potential and competing solutions can be examined in detail. Sensitivity analysis gives further insight into the solution; it also highlights the variables which should be carefully monitored.

Systems analysis may be applied for structuring a water resources project. To that end, a block diagram of the system is drawn and the elements are connected by logical statements. In this form of representation, it is easier to see how different components interact within the system and with its environment. By isolating the sub-systems, their performance can be tested and analyzed separately.

APPENDIX 5A

5A.1 Generation of Random Numbers

Most modern compilers have built-in routines to generate uniformly distributed random numbers between 0 and 1. A number of arithmetic techniques are used for this purpose, such as midsquare method, the congruence method, the composite generators, etc. The most popular of these is the congruence method. The random number generators have a mathematical expression that is used as a recursive equation to generate numbers. To start the process, a number known as 'seed' is input to the equation which gives a random number. This number is input to the equation to generate another number and so on. When this process is repeated n time, n random numbers are obtained. The recursive equation used in the *congruence method* is:

$$R_i = (aR_{i-1} + b) \text{ (modulo d)} \tag{5A.1}$$

where R_i are the random variables; and *a, b,* and *d* are positive integer constants which depend upon the properties of the computer. The word 'modulo' denotes that the variable to the left of this word is divided by the variable to the right (in this case *d*) and the remainder is assigned the value R_i. The initial value of the random variable (R_0) in eq. (4A.1) is called the seed. The properties of the generated numbers depends on the values of constants *a, b,*

and *d*, their relationships, and the computer used. The value of constant *a* needs to be sufficiently high; low values may not give good results. Constants *b* and *d* should not have any common factors. In computer generation, the sequence of random numbers is repeated after a certain lag and it is desirable that the length of this cycle should be more than the numbers that are needed for the study. This lag increases as *d* increases and therefore a large value of *d* should be chosen. Normally, *d* is set equal to the word length (the number of bits retained as a unit) of the computer; a typical value being 2^{31} - 1. This technique of random number generation is 'deterministic', i.e., the generated numbers can be duplicated again. Therefore, these numbers are called *pseudo random numbers*. The generated random numbers should be tested to ensure that these are not serially correlated and are uniformly distributed.

Example 5A.1: Generate uniformly distributed random numbers using eq. (5A.1) with a = 5, b = 3, and d = 7. The seed R_0 can be assigned a value of unity.

Solution: The eq. (5A.1) is re-written as $R_i = (5*R_{i-1} + 3)$ (modulo 7)

i	R_{i-1}	$(5*R_{i-1} + 3)$ (modulo 7)	R_{i-1}
1	1.00	(5*1.00 + 3) mod 7 = 8.00 mod 7	0.14
2	0.14	(5*0.14 + 3) mod 7 = 3.70 mod 7	0.53
3	0.53	(5*0.53 + 3) mod 7 = 5.65 mod 7	0.81
4	0.81	(5*0.81 + 3) mod 7 = 7.05 mod 7	0.01
5	0.01	(5*0.01 + 3) mod 7 = 3.05 mod 7	0.43
6	0.43	(5*0.43 + 3) mod 7 = 5.15 mod 7	0.73
7	0.73	(5*0.73 + 3) mod 7 = 6.65 mod 7	0.95
8	0.95	(5*0.95 + 3) mod 7 = 7.75 mod 7	0.11
9	0.11	(5*0.11 + 3) mod 7 = 3.55 mod 7	0.51

It may be noted that the numbers are (nearly) repeating after a cycle of 7.

5A.2 Transformation of Random Numbers

The input to the prototype system will have certain probability distribution. The input random variables in the Monte Carlo simulation should follow the same probability distribution. Therefore, the uniformly distributed random numbers are converted to follow the desired probability distribution. The variables involved may either be continuous or discrete random variables.

If the inverse form of a distribution can be expressed analytically, the inverse transformation is the simplest methods to generate random variables that follow a given distribution. In this method, first a uniformly distributed random number r_i in the range [0,1] is generated. If $F_Q(q)$ is the desired cumulative distribution function of random variable Q, then Q can be generated as

$$Q = F_Q^{-1}[r] \qquad (5A.2)$$

where F_Q^{-1} is the inverse of the cumulative distribution function of random variable Q. This is a simple and computationally efficient method, graphically illustrated in Fig. 5A.1.

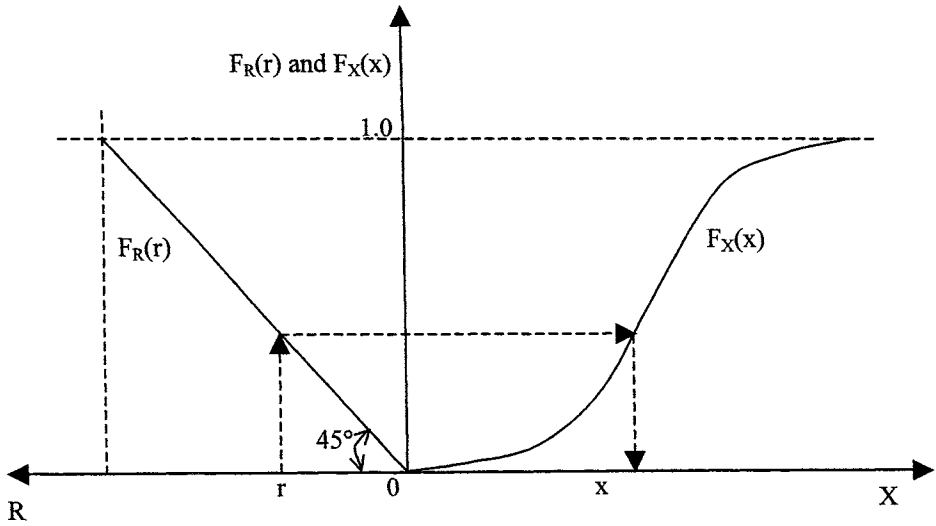

Fig. 5A.1 Determination of random number x with desired distribution from uniformly distributed random number r.

Example 5A.2: Generate exponentially distributed random numbers with parameter $\lambda = 2.3$.

Solution: The cumulative distribution function of an exponential distribution is

$$F_X(x) = 1 - e^{-\lambda x}$$

Its inverse can be written as

$$x = F_X^{-1}[r] = \ln(1 - r)/\lambda \tag{5A.3}$$

Since $(1 - r)$ is uniformly distributed, this can be replaced by r which is also uniformly distributed. Hence, exponentially distributed random numbers with the desired property can be generated by

$$x = -\ln (r)/2.3$$

If the first uniformly distributed random number $r_1 = 0.89$, the corresponding number x_1 will be

$$x_1 = -\ln(0.89)/2.3 = 0.05067.$$

The other common methods to generate random variables are the composition method and the functions based method.

5.11 REFERENCES

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

Buras, N. (1966). Dynamic Programming in Water Resources Development, in Advances in Hydroscience. Edited by V.T. Chow, Vol. 3, Academy Press, N.Y.

Cohon, J.L., and D.H. Marks (1975). A review and evaluation of multi-objective programming techniques. Water Resources Research, 11(2), 208-220.

Neufville, R. de, and Stafford, J.H. (1971). Systems Analysis for Engineers and Managers. McGraw-Hill Book Company, New York.

Dandy, G.C., Simpson, A.R., and Murphy, L.J. (1996). An improved genetic algorithm for pipe network optimization. Water Resources Research, 32(2), 449-458.

Dantzig, G.B., and Thapa, M.N. (1997). Linear Programming 1: Introduction. Springer-Verlag, New York.

Fletcher, R. (1980), Practical Methods of Optimization, Vol. 1. John Wiley & Sons, New York.

Goldberg, D.E. (1989). Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading, Mass.

Goulter, I.C., and Tai, F-K. (1985). Practical implications in the use of stochastic dynamic programming for reservoir operation, Water Resources Bulletin, 21(1), 65-74.

Haimes, Y.Y., and Hall, W.A. (1974). Multiobjectives in water resources system analysis: The surrogate worth trade-off method. Water Resources Research, 10(4), 615-623.

Heidari, M., Chow, V.T., Koktovic, P.V., and Meredith, D.D. (1971). Discrete differential dynamic programming approach to water resources systems optimization. Water Resources Research, 7(2), 273-282.

Himmelblau, D.M. (1972). Applied Nonlinear Programming, McGraw Hill Book Company, New York.

Hufmschmidt, M.M., and Fiering, M.B. (1966). Simulation Techniques for Design of Water Resources Systems. Harvard University Press, Cambridge, Massachusetts.

Ignizio, J.P. (1976). Goal Programming and Extensions. D C Death and Company, Lexington Books.

Jacoby, H.D., and Loucks, D.P. (1972). Combined use of optimization and simulation models in river basin planning. Water Resources Research, 8(6), 1401-1414.

James, L. D., and Lee, R. R. (1971). Economics of Water Resources Planning, McGraw-Hill Book Co., New York.

Jensen, P.A., and Barnes, J.W. (1980). Network Flow Programming, McGraw-Hill Inc., New York.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. Combinatorica, 4, 373-395.

Larson, R.E. (1968). State Increment Dynamic programming, Elsevier, New York.

Lee, S.M. (1972). Goal Programming for Decision Analysis. Auerbach Publishers Inc., Philadelphia.

Loganathan, G.V., and Bhattacharya, D. (1990). Goal programming techniques for optimal reservoir operations. Journal of Water Resources Planning and Management, ASCE, 116(6), 820-838.

Loucks, D.P., and Dorfman, P. (1975). An evaluation of some linear decision rules in

chance constraint models for reservoir planning and operation. Water Resources Research, 11(6), 777-782.

Major, D.C. (1977). Multiobjective Water Resources Planning, Water Resources Monograph No. 4, American Geophysical Union, Washington, D.C.

Marquardt, D.M. (1963). An Algorithm for Least Squares Estimation of Non-Linear Parameters. J. of Soc. Indus. App. Math, 11, pp.431-441.

Martin, Q.W. (1981). Surface Water Resources Allocation Model, AL-V, Program Documentation and Users Manual, Texas Department of Water Resources, USA.

Pedrycz, W. (1993). Fuzzy Logic and Fuzzy Systems, John Wiley, New York.

Rao, S.S. (1979). Optimization, Theory and Practice, Wiley Eastern, New Delhi.

ReVelle, C., Whitlatch, E.E., and Wright, J.R. (1997). Civil and Environmental Systems Engineering. Prentice Hall, New Jersey.

ReVelle, C. (1999). Optimizing Reservoir Resources. John Wiley & Sons Inc., New York.

ReVelle, C., Jores, E., and Kirkby, W. (1969). Linear decision rule in reservoir management and design 1: Development of stochastic model. Water Resources Research, 5(4), 767-777.

Reklaitis, G.V., Ravindran, A., and Ragsdell, K.M. (1983). Engineering Optimization – Methods and Applications, John Wiley & Sons, New York.

Russell, S.O. and Campbell, P.F. (1996). Reservoir operating rules with fuzzy programming. Water Resources Research, 122(3), 165-170.

Singh, V.P. (1992). Elementary Hydrology. Prentice-Hall, New Jersey.

Taha, H.A. (1982). Operations Research An Introduction. McMillan, New York.

Votruba, L., Kos, Z., Nachazel, K., Patera, A., and Zeman, V. (1988). Analysis of Water Resources Systems. Elsevier, Amsterdam.

Wardlaw, R. and Sharif, M. (1999). Evaluation of genetic algorithms for optimal reservoir system operation. Water Resources Research, 125(1), 26-33.

Yeh, W.W-G., and Becker, L. (1982). Multi-objective analysis of multireservoir operations. Water Resources Research, 18(5), 1326-1336.

Young, G.K. (1967). Finding reservoir operation rules. Journal of Hydraulics Division, ASCE, 93(HY6), 297-321.