APPLIED SYSTEMS ANALYSIS

Engineering Planning and Technology Management

Richard de Neufville
Massachusetts Institute of Technology

McGraw-Hill Publishing Company
New York St. Louis San Francisco Auckland Bogotá Caracas Hamburg Lisbon London Madrid Mexico Milan Montreal New Delhi Oklahoma City Paris San Juan São Paulo Singapore Sydney Tokyo Toronto

CHAPTER

5

LINEAR PROGRAMMING

5.1 CONCEPT

Linear programming (LP) is the most powerful method of constrained optimization available. Its power is staggering when compared with other approaches. Linear programs routinely deal with problems involving tens of thousands of variables and thousands of constraints. These kinds of problems are virtually intractable by other methods of analysis. The traditional methods of constrained optimization, for example, require the solution of systems of partial differential equations, notoriously time-consuming and difficult even for small problems.

Linear programming is also particularly attractive for design because it automatically provides extensive information on the sensitivity of the optimal design to different formulations of the problem. This feature is most important, because of our inherent uncertainty about the precise parameters of any situation, as discussed in Chapter 15. This sensitivity analysis is presented in Chapter 6.

Linear programming is so powerful because it exploits the computer's ability to execute simple calculations, such as additions and multiplications, very quickly. By representing a design problem by a system of linear equations it implies an optimization procedure that consists of a long series of solutions to these linear equations. This task is both simple-minded and tedious—an ideal combination for a digital computer.

Semantic caution: Linear programming is called "programming" for historical reasons. It is one of a series of techniques referred to collectively as *mathematical programming*, which also includes integer programming (Section 5.9) and dynamic programming (Chapter 7). None of these techniques usually requires the user to write programs for the computer in the sense now commonly used. The mathematical programming techniques are typically available in convenient packages.

As with all mathematical programming techniques, linear programming works because it assumes that certain assumptions can be made about a problem. These assumptions establish a mathematical structure for the problem which can be solved by a particular process. Linear programming assumes that the problem can legitimately be described or approximated by linear, additive and continuous functions (see Section 5.2). Linear programming, as any other mathematical programming technique, is thus limited to specific classes of problems, those that meet its assumptions.

The standard form for a linear program consists of two parts, an objective function and constraints. The *objective function* is an equation that defines the quantity to be optimized. For linear programming this quantity must be a one-dimensional scalar quantity such as:

$$Y = \sum c_i X_i$$

The variables in the objective function, the X_i, are known as *decision variables* because we seek to make decisions about them so as to optimize the objective. This name is useful because it focuses attention on the idea that the arguments of the objective functions are the decisions we have—this is most helpful when we try to formulate a real problem.

The constraints are of the form:

$$\sum_i a_{ij} X_i \gtreqless b_j$$

where the b_j are the upper or lower bounds on a particular feature of a problem and the a_{ij} parameters define the contribution of each decision variable to that feature.

Semantic caution: Note that the coefficients of the decision variables in the objective function are labeled c_i, instead of a_i as in previous chapters and in the more general economic literature. This is the traditional format for linear programming, which was initially mostly applied to problems of minimizing costs. In linear programming the "a" coefficients are traditionally reserved for the matrix of constraints, and the "b" parameters for the constraints. In discussing linear programming it is useful to maintain this tradition to facilitate discussions with professionals in operations research.

Both maximization and minimization problems can be handled by linear programming. A typical maximization problem is to

- maximize output or profit
- subject to a budget and other constraints

The question here is "how much can we obtain (from this factory, mine, system) with the amount of money and resources available, given that we must operate within technical, legal, economic, and political realities?" Conversely, a typical minimization problem is to

- minimize costs
- subject to requirements on the output and other constraints

Here the question is "how cheaply can we produce our output while meeting all the technical and other standards?"

Because linear programming problems in practice typically deal with a very large number of variables, X_i, and constraints, b_j, they are almost invariably presented in vector and matrix notation. Thus:

Max or Min: $Y = \mathbf{CX}$

Subject to: $\mathbf{AX} \gtreqless \mathbf{B}$

where **C**, **X**, and **B** are row and column vectors, and **A** is the matrix of a_{ij} parameters of the variables in the system of constraints. Note that in accordance with traditional practice, the symbols for the vectors are capitalized although the b_j and c_i are normally not.

5.2 ASSUMPTIONS

The power of linear programming is a consequence of the assumptions it makes about a problem. If these assumptions are not met, even approximately, by a problem, then linear programming will not provide a useful analysis. Linear programming is definitely not a method that is universally applicable. It does not apply in many situations, which is why other, less powerful methods such as dynamic programming are useful.

The central assumption of linear programming is that the objective function and all constraints are linear. Additionally, the decision variables are assumed to be continuous and nonnegative.

Linearity. The concept of linearity has a precise meaning that needs to be carefully understood. This idea is much more specific, and limited, than the general idea that most people have that a linear equation is some kind of summation of variables all to the power of one.

Formally, a function $f(\mathbf{X}) = f(X_1, \ldots, X_n)$ is *linear* if, for all variables X_i and constants S_i:

$$f(S_1X_1, \ldots, S_nX_n) = S_1f(X_1) + \cdots + S_nf(X_n)$$

This definition can be divided into two equivalent statements.

Constant returns. First of all, a linear function must have constant returns or economies of scale. The definition indeed implies that

$$f(S\mathbf{X}) = Sf(\mathbf{X})$$

that is, that multiplying every decision variable by a common factor S leads to

Nonlinearity of Fixed Charge Equations

The expression

$$f(\mathbf{X}) = a_0 + \sum a_iX_i$$

with the fixed charge a_0 can be seen to be nonlinear by testing the condition that

$$f(S\mathbf{X}) = Sf(\mathbf{X})$$

Multiplying each X_i by S thus leads to

$$f(S\mathbf{X}) = a_0 + S\sum a_iX_i$$

This is evidently not equal to

$$Sf(\mathbf{X}) = Sa_0 + S\sum a_iX_i$$

Thus, the expression with the fixed charge is not linear.

an S-fold change in the objective function or constraint. In this connection, you should carefully note that the function:

$$f(\mathbf{X}) = a_0 + \sum a_iX_i$$

is not linear (see box above). Problems involving such expressions are known as *fixed charge* problems, due to the presence of the fixed amount a_0, and are discussed in Section 5.8.

The restriction on constant returns or economies of scale can be significant in practice. Many important industries do in fact exhibit increasing returns, as Section 4.5 indicates. Linear programming must be used carefully, if at all, for those kinds of problems. For situations involving decreasing returns, however, the situation is not so drastic; it is possible to represent the situation to the accuracy desired by using piecewise linear approximations (see Section 5.7).

Additivity. Secondly, a linear function must be *additive*. This means that the value of the function with all X_i simultaneously is equal to the sum of the values of that function with each X_i by itself:

$$f(X_1, \ldots, X_n) = f(X_1) + \cdots + f(X_n)$$

The implication is that the contribution of each X_i to $f(\mathbf{X})$ should not depend on the presence or absence of any others. Such situations in fact exist quite commonly in practice (see following box). However, the difficulty is avoided quite easily in practice by formulating the linear program using what is known as "activities" as decision variables, a device explained in Section 5.6.

Examples of Non-Additive Situations

The transport of air cargo requires both pilots and fuel. Both must be available simultaneously; each alone achieves nothing. Thus:

$$\text{Air Cargo Carried} = f(\text{Pilots, Fuel})$$

but:

$$f(\text{Pilots, Fuel}) \neq f(\text{Pilots}) + f(\text{Fuel})$$

since:

$$f(\text{Pilots}) = f(\text{Fuel}) = 0$$

Similarly, the production of calves as a function of the number of bulls and cows is also clearly not additive. Bulls or cows by themselves will not ordinarily produce calves.

Continuity and nonnegativity. Finally, as previously indicated, linear programming assumes that the decision variables X_i are both continuous and nonnegative. Neither assumption is particularly limiting in practice, however.

The assumption that the decision variables are continuous means, as a practical matter, that the physical realities do not restrict them to integer or discrete values. This restriction could be seen as a considerable difficulty because many systems do indeed consist of design elements that can only be integer or discrete. For example, the number of ships in a fleet must be integer, as must be the number of warehouses in a distribution system. Similarly various components of a system, say the computers or memories in a communication network, may only come in standard sizes.

In practice the assumption of continuity is generally not a difficulty. This is for two reasons. On the one hand we can simply assume that integer variables are continuous and round off our results. In many instances this will be quite satisfactory since the possible error of a few percent is well within the accuracy with which we can formulate any real problem, as Chapter 6 indicates. On the other hand there are ways to formulate linear programs that can cope with integer values. These techniques, known collectively as integer programming, are reasonably expensive and should be avoided unless necessary. Yet, as described in Section 5.9, they do provide a means to use most of linear programming for those situations.

The assumption that all of the decision variables should be nonnegative:

$$X_i \geq 0 \qquad \text{all } i$$

is a technicality imposed by the way a linear program finds the optimum. This requirement does not restrict us in practice; we can always define variables so that they are not negative (see box above). This assumption thus merely requires that we be careful in the way we formulate a linear program.

Making Decision Variables Nonnegative

This can be done in two ways. The simpler case is when a variable appears to be consistently negative. In this situation we simply define a new variable, X_i', which is the negative of the other:

$$X_i' = -X_i$$

We then use X_i' in the linear program. For example, suppose that in a safety program one of the arguments is the number of accidents, measured from a previous high level, and that this variable is consequently negative. By simply focusing on the alternative variable:

$$\text{Accidents Prevented} = -(\text{Accidents from High Level})$$

we could define a nonnegative variable.

The second case occurs when a variable is expected to be both positive and negative. In this situation we could achieve nonnegativity by redefining a new base of reference for measuring the quantity sufficiently low so that all readings for our problem would be positive. Thus:

$$X_i' = X_i + (\text{Large Amount})$$

For example, if the variable were temperature, which might fluctuate above and below freezing and thus be both positive and negative on the Centigrade scale, we could redefine the temperature measurements in terms of degrees Kelvin and be sure that all temperature readings were positive on this absolute scale.

5.3 SOLUTION CONCEPT

The way a linear program finds the optimum is fundamentally very simple, despite the potentially enormous size and complexity of the actual problem. The solution is based on two consequences of the fundamental assumption of linearity of the objective function and the constraints. As with all mathematical programming techniques, linear programming achieves its power by exploiting the consequences of the assumptions it makes about the situation.

To describe the solution concept it is first necessary to explain the consequences of the linearity assumptions, which provide the basis for all linear programming procedures. The nature of these methods is described afterwards.

Effect of linearity assumption. The two consequences of the linearity assumption are that

- the feasible region is convex, if it exists at all.
- the optimum must be located at an edge of this region, specifically at one of its corner points.

As will be explained soon, these facts reduce the optimization procedure to a search through a limited set of well-defined combinations, which is very small compared to the infinite number of design possibilities.

To understand these results, it is first of all useful to imagine the feasible region to be composed of two parts: the space defined by the constraints and the volume defined by the relationship of the decision variables to the objective function.

The region defined by linear constraints must be convex if there is any feasible region at all. This is because any two points that satisfy any linear constraint must be on one side of it and any line between them must also be entirely in the feasible region. The operational definition of convexity of the feasible region (see Section 2.4) is therefore met. Figure 5.1 illustrates the situation in two dimensions.

It is possible, both mathematically and in practice, for there to be no feasible region at all. This occurs if the constraints are mutually contradictory and there is no possible set of the decision variables X that will satisfy all constraints. For example, it might simply not be possible to meet some environmental standards with available resources or technology. In such cases the solution is *infeasible*.

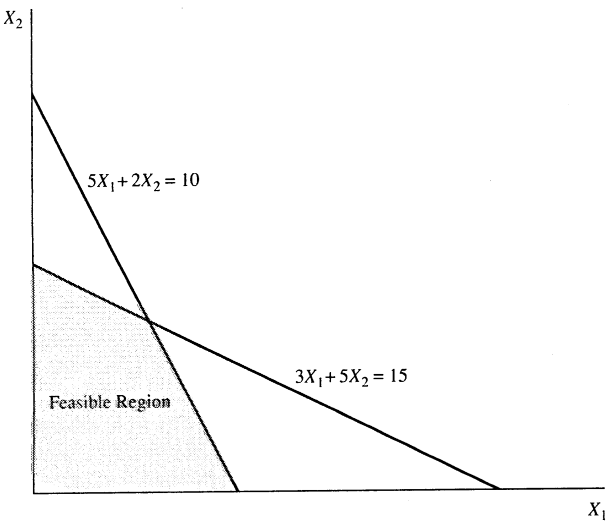

FIGURE 5.1
The feasible region is convex when defined by linear constants. (Here $5\ X_1\ + 2\ X_2 \leq 10$; $3\ X_1\ + 5\ X_2 \leq 15$; $X_1, X_2, \geq 0$).

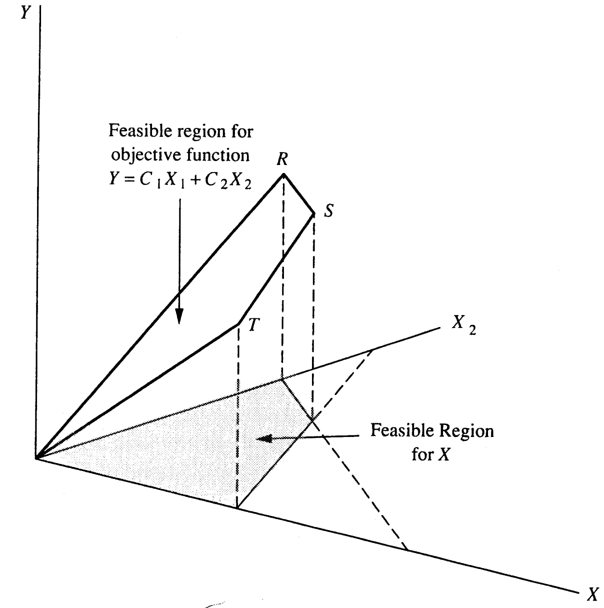

FIGURE 5.2
The feasible region for the linear objective function is also convex.

The volume defined by the linear objective function and the constraints is then also convex, by the same argument as before. Figure 5.2 illustrates the situation.

It follows from the above that the optimum solution to a linear program must be located at an edge to the feasible region, specifically at one of its corner points. Since the objective function is linear, the rate of change from any point interior to the region in any direction is constant. Thus, there is an edge that has a value of the objective function equal to or greater than any interior point. (The exception, in theory, occurs if the problem is *unbounded* so that it is possible to increase the objective function infinitely; the real world, however, does not provide this kind of luxury.) Similarly, since the rate of change of the objective function along every edge is also constant, the value of the objective function must be equal to or greater at one of the corner points to that edge than anywhere in the middle of the edge. An optimum solution must therefore lie at a corner point. For example, in trying to maximize the objective function shown in Figure 5.2, the highest point must be at R, S, or T.

Multiple optimal solutions to a linear program are possible. This occurs if the rate of change of the objective function along an edge is zero. Then the entire edge, including its corner points, constitutes a set of optimal solutions.

Every optimal solution to a linear program is a true, *global optimum*. It is the best maximum or minimum possible throughout the set, it is not a local optimum better than points in its vicinity but less desirable than some other point further away. This fact again results from the linearity of the objective function and the convexity of the feasible region, for the same reason that the optimum is at a corner point. The net effect of the linearity assumption is that a global optimum exists at a corner point of the feasible region. This provides the basis for all solution procedures.

Solution concept. The essence of all algorithms for solving linear programming problems is an organized search through the corner points. As it turns out, this is relatively easy to do, because of the nature of these points.

The corner points are defined by the intersections of the constraints. Each corner point is thus a solution to a set of linear equations. Solving such problems is easy in principle, being a system of elimination of unknowns and of substitutions, but tedious. It is an ideal task for high-speed computers.

The search through the corner points is facilitated by the fact that every corner point is linked to another by a constraint. Thus if you have solved for one corner point, for example point S in Figure 5.2, you can find others, such as R and T, by following the equations for the constraints to adjacent corner points.

The general procedure for finding the optimum solution to a linear program is thus:

- Find one corner point as a feasible solution to the constraints.
- Proceed sequentially to other corner points that provide better solutions.
- Stop when you arrive at a corner from which no adjacent corner leads to a better value of the objective function—this is the global optimum.

In practice, there are many different ways to execute this general procedure, as indicated next.

5.4 SOLUTION IN PRACTICE

A wide variety of computer programs are available to solve linear programming problems. They range from smaller routines that work on personal computers to very sophisticated programs that deal with thousands of variables to optimize the operation of huge enterprises. The suppliers of linear programming packages are in fact quite competitive so that the systems analyst can really pick and choose.

The systems analyst interested in solving problems does not have to be concerned with the exact details of the algorithms that actually do the linear program. The user of linear programs can now simply focus on the speed and capabilities of the alternatives, and only needs to be aware of the general possibilities. In this spirit this chapter does not spend any time on the actual details of any of the many algorithms for solving linear programs.

The most basic linear programs are based on the *simplex method*, which is the standard procedure described in textbooks. In its pure form the simplex method proceeds from corner point to corner point by the path that gives the

greatest improvement in the value of the objective function. From any corner it thus examines the rate of change of the objective function toward all adjacent corners. This approach is in fact quite inefficient for the kind of large problems found in practice, and is thus rarely used in sophisticated programs.

The simplex method is inefficient for two principal reasons. More obviously, it involves examination of all adjacent corners which, when dealing with thousands of variables and constraints, may be quite time-consuming. It can be much faster to consider only one or a few corner points that lead to improvements in the objective function. Depending on the exact nature of the problem you are dealing with, there are other algorithms that may be faster.

The simplex method is also inefficient because it does not deal explicitly with the nature of real problems in systems design. In practice, any system typically consists of many components, such as factories in an industry. There will be constraints on each of these components by themselves, constraints that do not affect other components of the systems. The net effect is that the matrix **A**, that describes the relationship of the decision variables **X** to the constraints **B**, will have a lot of empty spaces in it as suggested by Figure 5.3. This structure of the problem provides an opportunity for substantial savings in computer costs.

The computer effort required to solve a linear program is a function of the number of constraints involved. This is because the major task involved in any algorithm involves the solution of corner points, which requires the solution of a system of equations of the constraints. As a rough rule of thumb,

$$\text{LP Computer Effort} = f(\text{Number of Constraints})^2$$

The technical basis for this empirical function is that the solution of the equations requires the inversion of matrices of constraints.

The costs of solving real problems can thus be drastically reduced if we can break these big problems into a series of small problems. This is precisely

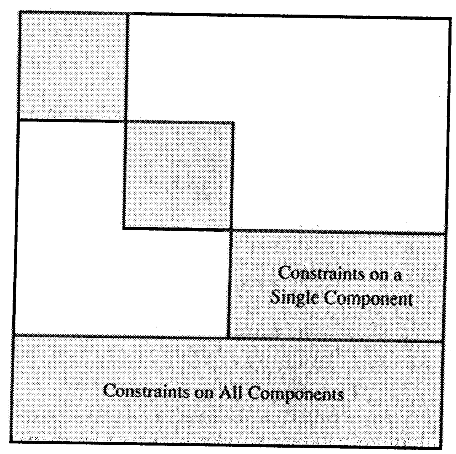

FIGURE 5.3
The A matrix of constraints on a real system involving components, typically having many zero entries.

what can be done when the structure of the problem leads to a matrix **A** similar to that shown in Figure 5.3. This *decomposition* can lead to an order of magnitude increase in speed of the linear programming procedure. Advanced linear programming algorithms can decompose large problems automatically. Most interestingly, decomposition appears to work particularly well on the new parallel processor computers.

The simplex algorithm is thus rarely used in practice for significant problems. Most of the methods actually used are beyond description because the developers of particularly efficient procedures like to keep them secret so that they maintain their advantage over the competition.

What the user must realize in this situation is that

- many different algorithms are available for solving linear programs.
- the efficiency of these algorithms depends both on the precise mathematical structure of the problem and the special tricks used by the algorithm.

Users must therefore explore the possibilities available to them and find which one best suits their needs.

5.5 BASIC FORMULATIONS

Linear programming can be used to optimize the design of many different kinds of systems. In practice, however, a large fraction of its applications has been directed toward two classes of problems, those known as "blending" and "transportation" problems. Each is described in turn to illustrate these basic formulations.

Blending problems. These optimize the mix of ingredients in a product, that is its blend, that will minimize the cost of production while meeting all constraints. The question here is essentially how to select from a very large menu. The optimum design will specify the amount of each ingredient to be selected or, alternatively, the percent of each ingredient in the product.

The blending problem is in the standard form:

Minimize: $\text{Cost} = \mathbf{CX}$

Subject to: $\mathbf{AX} \gtrless \mathbf{B}$

The decision variables, **X**, are the quantities of each ingredient selected. The constants, **C**, are the unit costs or prices of each ingredient. The constraints, **B**, are limits on various aspects of the product contributed by each **X** as defined by the matrix **A**. See box for an example of a blending problem.

The design of many products can be viewed as blending problems. These include the making of steel and other alloys from a wide variety of ores and recycled scrap; the production of fuels and petrochemicals from crude and feedstocks

A Typical Blending Problem

Suppose a food company wants to make sausages as cheaply as possible. Their ingredients are various meats, grains, spices, et cetera. Their constraints are industry and legal restrictions on the amount of fat, water, meat, and so on that can be in the sausages; and also the technical realities of the production process.

This problem can be formulated in several equivalent ways. We will define the decision variables as the weight of each ingredient in a standard batch of sausages.

- The objective is

  Minimize: $\text{Cost of Batch} = \mathbf{CX}$

  where the **C** are the unit costs by weight of each ingredient.
- Each constraint is of the form

$$\sum_i a_{ij} X_i \gtrless b_j$$

  where the a_{ij} represent the contribution per unit weight of each X_i to the factor constrained by b_j. Thus, if b_j is the maximum weight of fat in the batch of sausages, the a_{ij} are the percent of fat in each unit of weight of the X_i.
- There is also at least one technical constraint, that the weight of the ingredients including changes in processing has to add up to the total weight of the batch of sausages.
- Finally, the ingredients cannot be negative, so that

$$X_i \geq 0 \qquad \text{all } i$$

If the weight of the batch is defined as 100 units (kilograms or pounds, say) then the decision variables and the constraints are conveniently expressed in terms of percent.

---

of quite different constituents; the production of foods, such as bread, cereals, or hot dogs from grains and meats of quite different protein, moisture, and fat content; et cetera.

Industries faced with blending problems in fact use large linear programs routinely, sometimes daily. It must be realized that these industries are constantly redesigning their blends, to account for weekly or seasonal changes in the quality of their ingredients (scrap iron varies in chrome content, say, according to whether it comes from cars or ships; wheat has more water in the summer than in the winter), and to take advantage of even daily fluctuations in the prices of the ingredients. The optimal design of their products can typically be quite sensitive to small changes in the possible ingredients, and sensitivity analysis as described in Chapter 6 is most important in these industries.

Transportation problems. These optimize the distribution of a single product or commodity over some network. The question is where to send things, from what origin to what destination. The optimum design defines the quantities moving between any two points on the network.

These problems are called "transportation" problems because of their historical association with shipping. In fact, they relate to a wide class of situations where decisions have to be made about distribution over a network. The optimization of the development of a river basin or of an irrigation scheme are thus classically done as "transportation" problems—in these cases it is the water that is to be distributed or shipped over a series of paths.

The transportation problems use the standard form:

Minimize: $\text{Cost} = \mathbf{CX}$

Subject to: $\mathbf{AX} \gtrless \mathbf{B}$

The decision variables, X_{ij}, are the quantities of a product shipped from origin i to destination j, and the c_{ij} are the associated costs of making this shipment. The constraints may be both legal and technical, but the technical ones are most interesting in the transportation problem. These technical constraints are of two types:

- The total supplied from an origin i cannot exceed its inventory or capacity:

$$\sum_j X_{ij} \leq \text{Supply at Origin } i$$

- The total amount received at destination j should just equal the required demand since unnecessary shipments waste money:

$$\sum_i X_{ij} = \text{Need at Destination } j$$

- Finally, shipments cannot realistically be negative, so

$$X_{ij} \geq 0 \qquad \text{all } i \text{ and } j$$

The particularly interesting aspect of the transportation problems is that the matrix **A** is especially full of zeros. This is because each X_{ij} appears only twice in the matrix, once for the constraint on the origin and once for constraint at the destination. It then appears rather as in Figure 5.3. Moreover, all the nonzero entries in **A** are 1 as the preceding constraint equations show.

Special algorithms are available to solve the transportation problem. They exploit the special characteristics of the problem as contained in the matrix **A**, along the lines described in Section 5.4. These algorithms can be particularly fast and efficient compared to the standard simplex methods.

5.6 ACTIVITIES

This and the following two sections present several ways to formulate real problems into forms suitable for linear programming. The difficulty arises from the fact that whereas linear programming requires that a problem be described by a system of linear equations, the reality is that actual problems are generally nonlinear. The essential issue is: how can we adequately represent nonlinear situations with linear approximations?

The concept of an "activity" provides the basic method for representing a nonlinear model of a system through a set of linear equations. An *activity* is a specific way of combining basic materials or resources to achieve some objective or output. For example, the activity "flying Boeing 747 aircraft" is a specific way of using jet fuel and aircraft to accomplish the output of transporting passengers. An activity thus represents something intermediate between resources and output.

The concept of activities is subtle. Most people find it difficult to understand at first. Experience suggests that this is because we are used to thinking in terms of direct linkages between cause and effect, between resource and output, and thus have to make an extra effort to appreciate the idea and usefulness of activities.

To motivate the understanding of the usefulness of activities in linear programming, it is helpful to see what happens when we try to formulate a realistic problem without them. Consider the maximization of the output using several resources:

Maximize: $Y = g\text{ (Resources)}$

This problem involves a production function that generally, in a real situation, has isoquants asymptotic to the axes of the inputs (refer back to Figure 2.2 and the discussion in Section 2.4). If we try to represent this problem with a linear combination of resources:

$$g(\text{Resources}) = \sum (\text{Constant } i)(\text{Resource } i)$$

we obtain isoquants that are straight lines intercepting the axes, as shown in Figure 5.4. This representation is unrealistic. The concept of activities allows us to represent the production function with the curving, asymptotic isoquants it should realistically have. The demonstration is by example.

Consider a production function describing the creation of some automobile part, fenders for example, using labor and energy. Consider that there are two kinds of machines that can make these parts. Normally, these machines have a specific output, and require fixed amounts of labor and energy in any period. Table 5.1 shows the combinations assumed.

Use of either of the machines constitutes an activity. Each unit of an activity, for example, machine-hours of use, produces the same output and uses the same resources. The total result of any number of units of any activity is thus a linear function of the resources used. Other machines, using resources in a different ratio, would provide the basis for different activities.

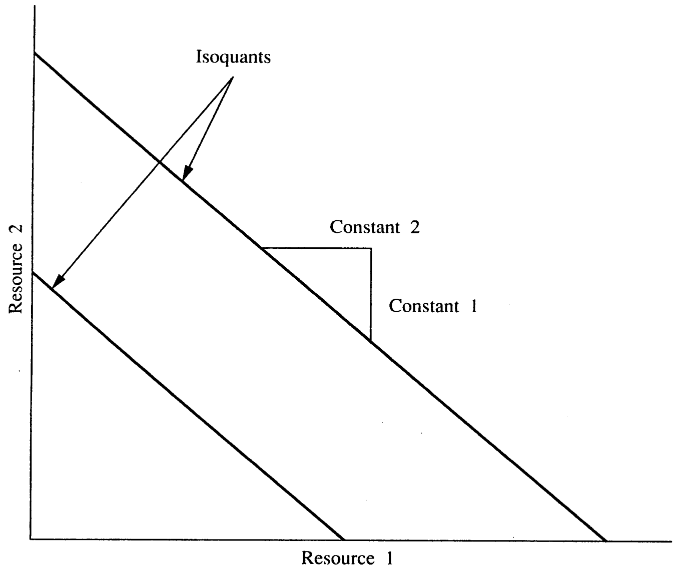

FIGURE 5.4
Isoquants for a production function represented as a linear combination of resources.

Consider now what the use of activities does to the representation of the isoquants of the production function. As Figure 5.5 indicates, each activity can be shown in the space of resources as a ray directed out from the origin at the constant angle, defined by the ratio in which they use resources. The use of a single machine of each type is defined by a point on these rays, in the figure by points *A* and *B*, and normally leads to different amounts of production. In this case point *A* is associated with an output of 50 fenders and point *B* with an output of 25. To obtain an isoquant we will thus have to use different quantities

TABLE 5.1
Production and Resources associated with each machine in a period.

Name of Machine	Resource 1 Energy Used (kwh)	Resource 2 Labor (person-hours)	Production Fenders
Activity 1	40	20	50
Activity 2	60	6	25

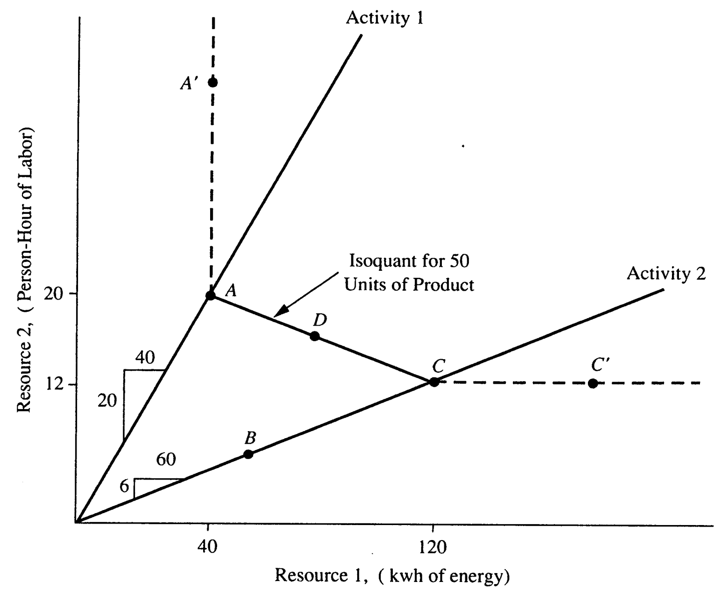

FIGURE 5.5
Isoquants for a production function represented as a linear combination of activities.

of each activity. For example, we can use either 1 unit of Activity 1 or 2 units of Activity 2, the latter being represented by point *C*. The isoquant passes through these two points.

What does the isoquant look like beyond the activity rays? The answer depends on whether one is between the activity rays, or between a ray and an axis. Between any two activity rays the output is simply a linear combination of the production of each type of activity. This means that the isoquant is simply a straight line between appropriate points on two rays. Caution: when several activities are used, each of the several different activities can combine, and the isoquant is represented by the line segments that imply the use of the least resources (those that lie closest to the origin). In our case, it is represented by the segment *AC*. For example, use of 1 unit of Activity 2, involving 60 kwh and 6 person-hours and giving a product of 25; plus half a unit of Activity 1, involving 20 kwh and 10 person-hours and also giving a product of 25; results in a total product of 50 using 80 kwh and 16 person-hours—as represented by point *D*. More generally, any point on the segment *AC* is a vector sum of some fraction of Activities 1 and 2.

The isoquant in the region between an axis and a ray is simply parallel to the axis. Consider point *C'* representing resources of 150 kwh and 12 person-

Formulation of LP Using Activities

Consider the problem of minimizing the cost of making a mixture, specifically an alloy, using different ores. Each ore is composed of various elements. The final alloy must have a definite chemical composition with upper and lower limits on the percentages of the various elements.

Each ore is an "activity" in that it is composed of a definite ratio of elements. The objective function is thus:

Minimize: $\text{Cost} = \mathbf{CX}$

where X_i is the quantity of ore i and c_i is its cost per unit weight. The set of constraints is

$$\mathbf{AX} \geq \mathbf{B}$$

where the b_j represent, first of all, a limit on the percent of some element that can be in the alloy. The a_{ij} then represent the percent of this element in ore i.

In this type of problem there is an additional constraint that must be introduced to reflect conservation of mass. The contribution by weight of each ore must sum to the total weight of alloy. If there are no losses in the smelting the conservation of mass is simply expressed as:

$$\sum X_i = \text{weight of alloy}$$

If, however, there are losses or impurities that burn off, this final constraint is:

$$\sum w_i X_i = \text{weight of alloy}$$

where each w_i represents the fraction that ore i contributes to the alloy.

---

hours: using the most energy-intensive activity available, we can only use up to 120 kwh for the labor available, additional power provides no extra product. A similar argument applies to A'.

The complete isoquant for 50 units of output is the broken line segment $A'ADCC'$. As more activities are used, the isoquant they portray will approximate a smooth curve more closely. We thus see that a linear combination of activities can correctly portray a realistic production function.

In practical problems, activities can represent a wide variety of technical possibilities. Here are some examples:

- Different types of machines, as previously discussed
- Different ways of organizing a production, for example the use of express trains or local trains

- Different materials used in a mixture or alloy, such as ores with various percentages of metals or grains with different protein and water contents

A linear program formulated using activities has the same form as any other linear program. The difference lies in the interpretation of the variables. Given a problem stated as

Optimize: $Y = \mathbf{CX}$

Subject to: $\mathbf{AX} \leq \mathbf{B}$

the decision variables X represent quantities of activities, and the constraints typically include limitations on the quantities of resources available or that must be used. See previous box for an example.

Notice that the use of activities also provides a way to deal with resources that are nonadditive, as defined in Section 5.2. We could not, for example, express the production of a transportation system for linear programming directly in terms of pilots and fuel—each alone achieves nothing. We could express this transport in terms of the use of different types of aircraft, each using standard ratios of pilots and fuel. This formulation would be additive, as the amount carried by one type of aircraft is independent of the amounts carried by other types.

5.7 NONLINEARITIES OF SCALE

The objective function for real problems is often nonlinear with respect to the scale or quantity of one or more of the decision variables X_i. Such situations commonly arise because of increasing or decreasing returns to scale, and economies or diseconomies of scale. Figure 5.6 illustrates the proposition. In such situations a simple linear representation of the form $Y = \sum c_i X_i$ may not be adequate.

Nonlinear objective functions $f(X_i)$ can be represented in linear programming by *piecewise linear approximations* consisting of a sequence of line segments. Linear programming modified to incorporate piecewise linear functions is called *separable programming* by some practitioners.

There are two difficulties in using piecewise linear approximations. The first is that the mathematical expression for the approximation is not obvious; indeed, it is arbitrary. Secondly, the approximation is only guaranteed to lead to a correct optimum, and should only be used when the feasible region is convex.

The piecewise linear approximation to a nonlinear function $f(X_i)$ consists of line segments that abut at *breakpoints*, as Figure 5.7 illustrates. The analyst must define these breakpoints, both in terms of X_i and $f(X_i)$. Unfortunately there are many equally acceptable ways this can be done, and each may lead to a different optimum solution. Greater accuracy can be obtained by using more line segments, but the best practical result is still an approximation.

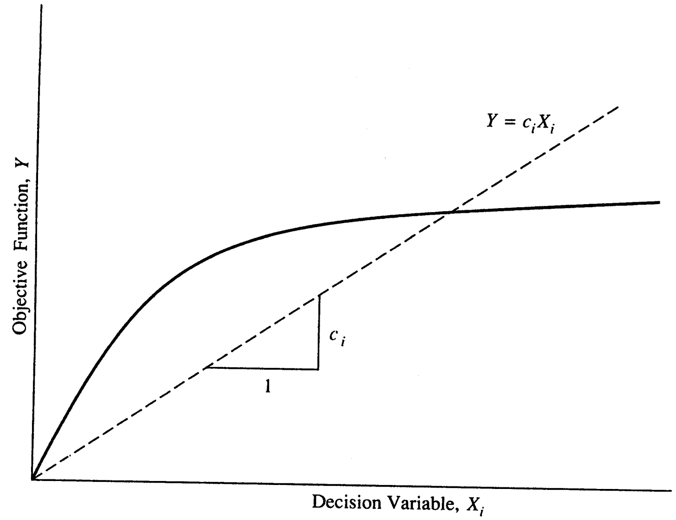

FIGURE 5.6
A simple linear representation may be an inadequate approximation to a nonlinear objective function.

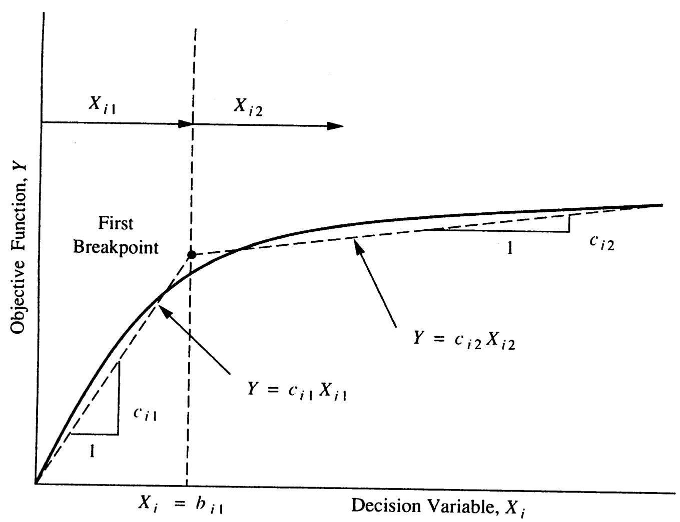

FIGURE 5.7
A piecewise linear approximation to a nonlinear objective function, illustrating the breakpoint; the new parameters c_{i1}, c_{i2}, and b_{i1}, and new variables X_{i1}, X_{i2} that must be introduced.

The use of piecewise linear approximations increases the number of variables and constraints. Specifically, each breakpoint used in the approximation requires one new variable and one new constraint. This fact limits the accuracy one may want to achieve by having the approximation consist of more line segments and breakpoints; this is because the cost of getting the solution is proportional to the square of the number of constraints (See Section 5.4).

The way the piecewise linear approximation increases the variables and constraints is a bit complicated. The central idea is that the expression for Y is no longer in terms of a single representation of X_i, as in Figure 5.6, but in terms of as many X_{ij} as there are line segments. The contributions to Y will be linear, of the form

$$Y = \sum_j c_{ij}X_{ij}$$

so long as the X_{ij} are defined to start from zero at the breakpoint beginning the line segment. Thus X_{i2} starts at b_{i1} as indicated in Figure 5.7. Concurrently, it is necessary to ensure that as each X_{ij} starts the preceding one stops—otherwise one would be double-counting the quantity of X_i. This is accomplished by placing a constraint on the X_{ij} up to the breakpoint defining the next line segment. Thus, $X_{i1} \leq b_{i1}$.

A linear program can be formulated for nonlinear returns to scale for the decision variables X_i provided that we:

1. Define each X_i as a $\sum_j X_{ij}$, $X_{ij} \geq 0$.
2. Define $f(X_i)$ as $\sum_j c_{ij}X_{ij}$ with the c_{ij} determined by the breakpoints selected.
3. Constrain the X_{ij} to the length of the associated line segment, $X_{ij} \leq b_{ij}$.
4. Incorporate the X_i in the matrix of constraints by multiplying each a_{ij} by X_{ij}.

See following box for example.

This formulation leads to a successful optimization only if the feasible region is convex. If it is not, the optimization cannot be depended on to succeed, and should not be tried. This follows generally from the theoretical basis of linear programming, which is that the global optimum can be obtained because the feasible region is convex. It is also interesting to see in detail why the piecewise linear approximation fails when the feasible region is not convex.

Consider first the convex feasible region below the curve in Figure 5.7, which might be associated with the maximization of the production $Y = f(X)$. As the computer seeks to maximize, it will choose initial amounts of X_{ij} whenever possible, for example X_{i1} over X_{i2}. This is because $c_{i1} > c_{i2}$ and thus that X_{i1} contributes more to Y than X_{i2}. Meanwhile X_{i1} meets the constraints as well as and no more than X_{i2}. Eventually, if the computer arrives at a solution that requires an $X_i > b_{i1}$, the quantity of X_{i1} in the solution will be limited and X_{i2} will be selected. The final solution will make sense in terms of X_i because a continuous amount of it will be specified: from 0 to b_{i1} by X_{i1} and from b_{i1} on by X_{i2}, and so on.

Formulation of LP Using Piecewise Approximation

Consider the problem of minimizing the cost of production involving labor, X_1, and materials X_2. Labor costs $10 an hour up to 6000 hours a month for the 40 person work force; beyond that salaries are $15 an hour (that is, time and a half for overtime). Suppose also that the production, defined by $a_1X_1 + a_2X_2$, must meet its quota of 400.

The feasible region is convex; we have diseconomies of scale and are minimizing costs. We can thus proceed with the approximation, substituting (X_{11}, X_{12}) for X_1.

The objective function is

$$\text{Minimize:} \quad \text{Cost} = 10X_{11} + 15X_{12} + c_2X_2$$

The constraint on production is

$$a_1X_{11} + a_1X_{12} + a_2X_2 \geq 400$$

There is the constraint associated with the breakpoint,

$$X_{11} \leq 6000$$

as well as the usual:

$$X_{11}, X_{12}, X_2 \geq 0$$

Consider now the nonconvex feasible region above the curve in Figure 5.7, which might be associated with the minimization of costs. In trying to minimize Y, the computer will select X_{i2} before X_{i1}, because $c_{i2} < c_{i1}$. It is then entirely possible to have a result come out of the computer which makes no sense at all in that the quantity of X_i starts at b_{i1} with nothing before it! In short, if you use a piecewise linear approximation on a nonconvex feasible region, you may get an answer from the computer which superficially looks reasonable but is actually garbage.

5.8 INTEGER PROGRAMMING

Real problems are often discontinuous. Most typically this occurs when the inputs occur in integer amounts. For example, aircraft in an airline fleet are clearly integer. These kinds of discontinuities are frequently ignored in practice, especially when numbers are large. Thus, if the solution specifies 56.2 aircraft it is usual to round off this result. This is all the more appropriate when we realize that the formulation of the linear program is an approximation to reality, and that its results must be further tested, as Chapter 6 describes.

Optimization over convex feasible regions with discontinuous variables can also be solved formally by *integer programming*. This is a variation of linear programming which forces the solution to be integer by defining additional variables and constraints. There are several versions of integer programming. They are all relatively expensive in terms of computer time and are thus avoided if possible. If they must be used, they can be found in software libraries, combined with regular linear programs, in what are then typically called *mixed integer programs*. As far as the user is concerned, they operate pretty much like standard linear programs, except for the cost of computation.

5.9 FIXED CHARGES

A particular, common form of discontinuity requiring special attention is the fixed charge problem. A *fixed charge* is a specific amount, typically a cost, associated with any level of a decision variable. For example, the cost of X_i might be of the form

$$\text{Cost of } X_i = c_0 + c_iX_i$$

Fixed charges occur in practice when there is a basic cost of operating a facility, regardless of its level of activity. The cost of a warehouse, for instance, may involve fixed charges for rent and taxes, independent of the number of tons stored.

The difficulty with a fixed charge arises because it is often possible to avoid the charge entirely. We might, for example, have the option of establishing a warehouse or not having one at all. The cost associated with X_i is then more properly stated as

$$\text{Cost of } X_i \begin{cases} = c_0 + c_iX_i & X_i > 0 \\ = 0 & X_i = 0 \end{cases}$$

This situation, illustrated in Figure 5.8, causes difficulty because the optimization

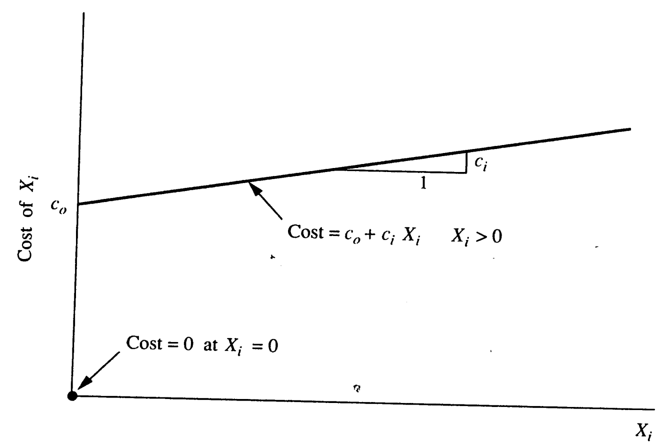

FIGURE 5.8
Graphical representation of a cost function with an avoidable fixed charge.

procedure in the linear programming chooses to include decision variables in the solution based on the coefficients c_i and does not have a way of appreciating the significant increase in cost c_0 that arises from the introduction of X_i into the solution. Linear programming simply does not handle fixed charges in the form given previously.

In the form in which fixed charges usually arise, the problem cannot be solved directly. This is because it involves minimizing cost, and the feasible region—above the line in Figure 5.8 and at the origin—is not convex. The only situation in which the fixed charge problem can be solved directly is when the charge cannot be avoided:

$$\text{Minimize:} \quad \text{Cost} = g(c_0 + c_iX_i) \qquad X_i \geq 0$$

One can then subtract the fixed charges from both sides of the equation and optimize the remainder:

$$\text{Minimize:} \quad (\text{Cost} - c_0) = g(c_iX_i)$$

The solution to this problem is the solution to the simpler problem when the fixed charge cannot be avoided.

If the fixed charges are in fact unavoidable, there are three possible ways to look for a solution:

- Neglect the fixed charge, assuming that it is small compared to the total costs.
- Use a simple, straight-line approximation, which may be rather crude (Note that a piecewise linear approximation could not be used to represent the cost function with the fixed charge because the feasible region would not be convex).
- Run the problem with and without specific facilities in several passes.

The last approach works in principle but is only practical if the number of activities with fixed charges is very small, so that a manageable number of runs of the linear program are involved.

5.10 DUALITY

Duality is a mathematical concept of great importance to anyone who wants or needs to become involved in the details of any of the algorithms for linear programming. A person who wishes simply to apply the techniques to solve real problems could get by without knowing anything about duality. However, since much of the literature on linear programming refers to duality, and because understanding duality can help one develop a more intuitive feel for linear programming and sensitivity analysis, it is appropriate to explore this concept.

Duality in mathematics refers to cases in which there are two quite different ways of representing a problem or situation, both of which provide all the same kind of information and detail, but in a different form. These two representations are known as *duals* of each other.

Duality is usually a difficult concept to explain, although it is in fact quite simple. It is perhaps best introduced by specific examples:

Use of Duality for Editing Cartographic Data Bases

The nature of maps is changing fundamentally. The Gutenberg era is fading and electronic or computer-based maps are coming in. These cartographic data bases have enormous potential because they can be combined with all kinds of other files that can be flagged by address or location (such as the entire U.S. Census) to provide cartographic spread sheets in which data is displayed geographically rather than in a tabular form with no particular organization.

The computer maps are created by encoding the latitude and longitude of every intersection on the map, together with indications of their connections to other points. This process of digitizing is subject to a lot of error that is very difficult to correct by itself.

The standard way of editing these cartographic data bases is by exploiting the duality between boundaries and surfaces. This is known as the DIME (Dual Integrated Map Encoding) procedure. It uses the fact that areas within a boundary should only touch certain other areas; if they do not, this signals an error. The process can locate errors efficiently.

- A photograph and its mirror image are duals of each other, the difference between them is that all features are transposed left to right.
- A negative of a photograph and its positive are likewise duals, here the transposition is between light and dark tones.
- The boundaries to areas or spaces and these spaces themselves are duals, here the transposition is between the shell and its content.

These examples illustrate two aspects of duality. First, a dual is not simply a change in scale, it is a structurally different way of representing the same idea. Thus an enlargement of a print is not its dual, but its mirror image is. Secondly, a dual can be constructed from the original one once one knows the rules of transposition that are to be used.

Duals have a number of practical applications in many different fields besides linear and mathematical programming. An important use is in automatic editing and correction of complicated systems. Because duals are completely redundant representations of an original problem in every respect, they provide an ideal basis for verifying an original (see above box for an example).

All linear programs have duals. If the original problem is a maximization the dual is a minimization, and vice-versa. The first formulation, whether a maximization or minimization, is called the *primal* and its mirror image the dual.

The rule for defining a dual linear program from its primal consists of two parts: the transposition of the structure of the primal to a different form, and the introduction of a new objective function and set of decision variables. Thus, if we start with the problem

Maximize: $Y = \mathbf{CX}$

Subject to: $\mathbf{AX} \gtrless \mathbf{B} \qquad \mathbf{X} \geq 0$

its dual is

Minimize: $Z = \mathbf{B}^T\mathbf{W}$

Subject to: $\mathbf{A}^T\mathbf{W} \gtrless \mathbf{C}^T \qquad \mathbf{W} \geq 0$

where **W** denotes the new decision variables and Z is the new objective function. [Note: superscript T indicates that the vector or matrix has been transposed.] Particularly observe that the inequality signs in the dual are reversed. See box for an example.

Writing the Dual from the Primal

Suppose the primal problem is

Minimize: $4X_1 + 12X_2 + 10X_3$

Subject to:

$$\begin{aligned} 2X_1 + X_2 \quad &\geq 2 \\ X_1 + 2X_2 + X_3 &\geq 5 \\ X_3 &\geq 1 \\ X_1, \quad X_2, \quad X_3 &\geq 0 \end{aligned}$$

In vector notation we have $\mathbf{C} = [4,12,10]$

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 2 \\ 5 \\ 1 \end{bmatrix}$$

To write the dual we have to transpose the matrices. This operation rewrites the row vectors as columns and vice-versa, and rotates a matrix. This leads to

$$\mathbf{C}^T = \begin{bmatrix} 4 \\ 12 \\ 10 \end{bmatrix} \qquad \mathbf{A}^T = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\mathbf{B}^T = [2, 5, 1]$$

The dual problem is then

Minimize: $2W_1 + 5W_2 + W_3$

Subject to:

$$\begin{aligned} 2W_1 + W_2 \quad &\leq 4 \\ W_1 + 2W_2 \quad &\leq 12 \\ W_2 + W_3 &\leq 10 \\ W_1, \quad W_2, \quad W_3 &\geq 0 \end{aligned}$$

It should particularly be noted that the inequality signs are reversed in the dual.

A most important feature of dual problems is that the optimal answer to the primal and the dual is identical:

$$Y^* = Z^*$$

This follows from the fact that the dual problems represent the same problem, even though in a different view.

The fact that you get the same result whether you solve the primal or the dual problem makes it possible to construct linear programming algorithms that may be particularly efficient. Indeed, it is often the case that solving the dual is an order of magnitude easier than solving the original, primal problem.

The potential advantage of solving dual problems derives from the possibility that the dual may have far fewer constraints than the primal. Since the cost of solving linear programs is approximately proportional to the number of constraints squared (see Section 5.4), this difference may lead to significant savings.

The difference in the number of constraints in the primal and dual follows from the way the dual is constructed. Going from the primal to the dual, the number of decision variables in the primal (the entries in **C**) becomes the number of constraints (the entries in $\mathbf{C}^T$). Conversely, the number of constraints in the primal (indicated by **B**) becomes the number of decision variables in the dual ($\mathbf{B}^T$). If our primal problem has few decision variables and many constraints, it would be far easier to solve by its dual, which would have few constraints.

In practice, we really only have to know about the possible advantage of solving the dual if we are using unsophisticated linear programming packages. The most advanced versions will automatically work on the dual problem if that is advantageous; moreover, some of them work back and forth between the primal and the dual because this tactic may provide some advantages.

Finally, the elements of dual problems have some useful physical interpretations. These are explained in detail in the discussion of sensitivity analysis in Chapter 6.

5.11 APPLICATIONS

This section focuses on the practical questions that must be confronted when trying to analyze a real system using linear programming. The particular application concerns the minimization of costs in a transportation problem, and thus represents one of the most common uses of linear programming. Specifically, the example deals with the development of the Jordan river basin around Salt Lake City. This case, being in the public sector, has the advantage of being well-documented, in contrast to industrial applications which are usually confidential.

As for all transportation problems, the design of the system requires the analyst to specify which flows will go from the sources of supply to the intermediate and final destinations. In general, we may think of shipments being made from factories to users through warehouses and stores. For a river basin, the many sources are rivers and wells, supplemented by water imported by aqueducts or obtained by desalting. The uses are mostly residential or municipal; industrial, as for cooling power systems; agricultural; and hydroelectric power. Additionally, of course, the upstream users normally must deliver a fair share of the water supply to users downstream. Finally, the intermediate destinations and sources of supply are the facilities for recycling municipal and industrial wastes and the runoff from irrigation systems.

The network of flows in a river basin is quite complex whenever water is scarce. The planner must then anticipate that with recycling and pumping, every user is also a potential source of water. Virtually all combinations of sources and users may be linked. Such is the situation for Salt Lake City, especially as population growth puts pressure on a normally arid region.

The basic linear program for this situation was set up as a transportation problem as described in Section 5.5. As normally true for any real situation, this formulation has to be augmented to incorporate the particular features of the problem. The ones of interest here are those that apply generally to many cases. These concern the nature of intermediate operations, infeasible paths, losses, and blending operations.

Intermediate operations are bound by continuity and capacity. Continuity means that, over time, what flows in must flow out. This leads to a constraint for each intermediate facility of the form

$$\sum \text{(flows in)} - \sum \text{(flows out)} = 0$$

Similarly, any treatment plant or reservoir (or warehouse) may have fixed capacity, for example

$$\sum \text{(flows in)} \leq \text{Capacity}$$

When the optimal capacity is to be determined, however, the flows through the intermediate point can be unconstrained. This allows the linear program to set the flows at the level required to minimize costs and thus provides insight into the optimum capacity of these intermediate facilities.

In some cases, limitations may make it impractical to have flows between a source and a use. The flow may be physically impractical (as over a mountain) or politically infeasible due to legal agreements or environmental regulations (untreated municipal wastes cannot be used for irrigation). Infeasible flows can be handled in two ways: they can either be excluded from the formulation or included in a way that precludes their use. Superficially it would seem simpler to exclude impossible flows. As a practical matter, this may not be the case for large systems: the task of describing the matrix of constraints (possibly 1000 $\times$5000, say) may best be done by an automated preprocessor. It then may be easier to leave all flows in and to exclude those that are infeasible by assigning them extraordinarily high costs that will prevent them from being incorporated into the optimal plan.

Losses in the system, as through evaporation or stealing, can be incorporated easily into the linear program. They are simply defined as a final use or outflow from the system.

Blending operations mix flows with different characteristics, such as salt content, to achieve an acceptable product. Thus rain water might be mixed with some salt water to obtain a larger amount of drinkable water. This situation is handled by a constraint involving the maximum blending ratio of the pure and salty water:

$$\text{(Blending Ratio)} \sum \text{(Pure Flows)} - \sum \text{(Salty Flows)} > 0$$

The coefficients defining the specific problem must be detailed once its overall structure is set. This requires considerable hard work, often very much more than needed to formulate the linear program. For the river basin, flows, uses, and costs needed to be estimated. Here, and in general, the parameters selected are approximate at best.

Rain and other natural sources of water are inherently variable. Over time their distribution may be quite stable, as field measurements can document, but their level in any year may fluctuate widely. This difficulty is normally handled by assuming that the available flow is somewhere toward the conservative side of the distribution. In planning flood control reservoirs, it is usual to estimate the flow as the "100-year flood", thus including about 99% of the distribution. For drinking water, one looks at the "safe yield" which is normally greater than the lowest 1% of the possibilities. In short, normal practice is to define the flow using judgements about what appears to be a reasonable value from the distribution of possibilities.

Future uses must likewise be approximated. These estimates are necessarily worse than those of the flows. Uses, unlike flows, are not drawn from a stable distribution. They tend to grow, with unpredictable spurts and lags due to economic ups and downs, population shifts, and technical changes. Costs are also difficult to estimate satisfactorily, as Chapter 14 explains in detail.

In planning developments involving long-term investments, such as reservoirs and aqueducts (or mines and factories), it is necessary to make sure that the system performs well throughout its lifetime. For the Jordan River, the analysts projected the uses for the years 1980, 2000, and 2020, and defined optimal flows for each of these scenarios.

The solution for any year depends directly on the values assumed for the parameters of the problems. Since these are estimates, one can only obtain estimates of what would be the optimal design in fact. To give oneself confidence that the projected systems will perform satisfactorily for the various values of the parameters that may occur, it is necessary to explore the sensitivity of the solutions to these values. Chapter 6 indicates how to do this systematically.

The solutions for different periods typically vary significantly from each other. For the Jordan River, for example, no desalting is necessary immediately—but seems vital for the next century. Likewise the optimal pattern of flows may change: whereas now it is most economical to pump groundwater directly to households, it soon would appear necessary to blend it with rainwater. This difference in optimal designs presents planners with a problem: to what extent should investments be made in facilities needed now but not in the future? This kind of issue is best dealt with by dynamic programming, which Chapter 7 presents in detail. Ultimately, the design of a complete system may require the complementary use of both linear and dynamic programming, as Chapter 9 indicates.

REFERENCES

Bishop, A. B., and Hendricks, D. W., (1971, 1974). Chapter 12 in de Neufville and Marks (1974). Also "Water Resources Systems Analysis," *Journal of the Sanitary Engineering Div.*, ASCE, 99, February 1971, pp.41–57.

de Neufville, R., and Marks, D. W., (1974). *Systems Planning and Design: Case Studies in Modeling, Optimization and Design*, Prentice-Hall, Englewood Cliffs, NJ.

Wagner, H. M., (1975). "Sensitivity Testing and Duality," Chapter 5, *Principles of Operations Research*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ.

Williams, H. P., (1985). *Model Building in Mathematical Programming*, 2nd ed., Wiley, New York.

PROBLEMS

5.1. *Feasible Regions*

Which of the following problems can be solved by LP? Explain your answer.

(*a*) Maximize: $Z = X + Y$
Subject to: $4X + 3Y \leq 2 \quad 2XY \leq 3 \quad X, Y \geq 0$

(*b*) Maximize: $Z = 4X + Y$
Subject to: $X + Y = 2 \quad 0.5X + Y \geq 0 \quad X, Y \geq 0$

(*c*) Minimize: $Z = X + Y$
Subject to: $X \leq 4 \quad X + Y \leq 7$

(*d*) Maximize: $Z = X + 9Y$
Subject to: $2X - Y \leq 2 \quad -5X + 2.5Y \leq 1 \quad X, Y \geq 0$

(*e*) Minimize: $Z = X + 9Y$
Subject to: $2X - Y \geq 2 \quad -5X + 2.5Y \geq 1 \quad X, Y \geq 0$

(*f*) Maximize: $Z = X + Y$
Subject to: $X - Y \geq 0 \quad 3X - Y \leq -3 \quad X, Y \geq 0$

5.2. *Feasible Region I*

Consider the constraints:

$$3X_1 + 4X_2 \geq 12 \qquad 5X_1 + 4X_2 \leq 40$$
$$X_1 - 3X_2 \leq 3 \qquad 0 \leq X_1 \qquad X_1 \leq 6 \qquad X_2 \leq 6$$

Sketch the feasible region, label each extreme point, and find the extreme point(s) that

(*a*) Maximize(s): X_1; $X_1 + X_2$; $X_1 - 2X_2$; $10X_1 + 8X_2$
(*b*) Minimize(s): X_2; $X_1 + X_2$; $X_1 - 2X_2$

5.3. *Feasible Region II*

Show graphically the feasible region for the constraint set:

$$\begin{aligned} 2X_1 + 3X_2 &\leq 18 \\ 2X_1 + X_2 &\leq 12 \\ 3X_1 + 3X_2 &\leq 24 \\ X_1, \quad X_2 &\geq 0 \end{aligned}$$

Do these equations define a convex region? With what linear objective function, if any, will (2,2) be optimal?

5.4. *Graphic Interpretation*

Given the linear programming problem:

Maximize: $Z = 4X_1 + 3X_2$

Subject to:

$$\begin{aligned} 5X_1 + X_2 &\leq 40 \\ 2X_1 + X_2 &\leq 20 \\ X_1 + 2X_2 &\leq 30 \\ X_1, \quad X_2 &\geq 0 \end{aligned}$$

(*a*) Draw the feasible region in the resource space.
(*b*) Label the extreme points.
(*c*) Solve the problem for the maximum Z by inspection.
(*d*) Write the dual of the above problem.

5.5. *Meat Market*

Annette's Meat Market sells two kinds of hamburger, both of which are ground fresh daily. The regular grade contains 20% ground beef and 80% oatmeal, while the deluxe grade is 30% beef and 70% oatmeal. The regular grade sells for p_1 cents/lb; the deluxe grade sells for p_2 cents/lb. Annette can buy up to L pounds of beef per day at c_1 cents/lb, but she must pay c_2 cents/lb for each additional pound beyond L per day ($c_2 > c_1$). Oatmeal costs c_3 cents/lb, regardless of quantity purchased.

Annette can spend a total of D dollars per day on the ingredients. Her problem is how much of each ingredient she should purchase daily and how much regular and deluxe hamburger she should make. We are assuming that the customers will buy all of the hamburger she can grind. Carefully formulate as a linear programming problem. This involves: Clear definition of variables; Specification of an objective function, and of constraints.

Include in your answer a succinct explanation of the meaning of the objective function and constraints.

5.6. *STOMP Manufacturing*

STOMP, Inc., buys bulk metal (rods, strips, and so on.) and runs it through processing machines to produce parts, which are then sold to consumer products manufacturers. One stamping machine may be used to produce bird-cage hinges, nameplates for hi-fi speakers, or a mix of the two. A case of the hinges requires 15 min of time on the machine, 5 lbs of metal, and sells for $150. A case of the nameplates, requiring 20 machine min and 10 lbs of metal, sells for $300. Any number of hinges, but only 75 cases of nameplates, may be sold per week.

The metal used costs $20/lb, and is available in any desired quantity. The marginal costs of running the machine (power, use-related maintenance, and so on.) are about $25 hr. The machine operator is on a 40-hr/wk union contract which must be paid anyway, but is willing to work overtime at $12/hr.

Due to delays in receiving payments from customers, STOMP cannot afford to lay out more than $30,000/wk for metal, machine costs, and overtime. Formulate this as a linear programming problem.

(*a*) Define the variables.
(*b*) Specify the objective function and explain its meaning.
(*c*) Specify the constraints and explain their meaning.

5.7. *Mountain Movers*
The Mountain Movers truck company has $1,200,000 to replace its fleet. The company is considering three different truck types:

	Truck type		
	A	B	C
Carrying capacity (tons):	10	20	18
Average speed (mph):	40	35	35
Price ($1,000):	48	78	90
Drivers required:	1	2	2
Shifts per day:	3	3	3
Hours per day:	18	18	21

Type C is the improved model of type B. It has sleeping space for one driver. This reduces its carrying capacity to 18 tons and increases the price to $90,000. The company employs 150 drivers every day, has 50 overnight parking spaces, and wishes to maximize the delivery capacity of its fleet (in ton-mi/day).

(*a*) Define the selection of the trucks as an optimization problem. Explain the decision variables, objective function, and constraints.
(*b*) What assumptions, if any, did you make in this problem?

5.8. *Alloy Optimization*
Suppose a manufacturer must make the decision to use one or more of three different production processes, each yielding the same item. The inputs for each process include chrome and carbon (both measured in pounds). Since each item varies in its input requirements, the profitabilities of the processes vary. Process one requires 6 lbs of chrome, 4 lbs of carbon, and yields a $30 profit/unit. Process two uses 5 lbs of chrome, 2 lbs of carbon, and yields $23/unit. Process three uses 3 lbs of chrome, 5 lbs of carbon, and yields $29/unit. The manufacturer in deciding on the production schedule is limited by the available amounts of chrome and carbon. There are 26 tons of chrome available and 7 tons of carbon.

(*a*) Formulate a linear programming model to determine the manufacturer's optimal production plan; that is, write the objective function and the constraints.
(*b*) What are the activities associated with the formulation?

5.9. *Paint it Easy*
The president of the "Paint it Easy" paint company is in trouble: People complain that the paint does not last long. Experts now recommend a new formula such that

$$\begin{aligned} \text{Boiling point} &\geq 70°\text{C} \\ \text{Hardening point} &\leq 4 \text{ hr} \\ \text{Plastic} &\leq 70\% \text{ by volume} \\ \text{Acid} &\geq 15\% \text{ by volume} \end{aligned}$$

Four components, A–D, can be used to produce the paint. Their contribution to each of the preceding characteristics is in proportion to their part in the mixture. The coefficients are:

	A	B	C	D
Boiling point	60	99	15	30
Hardening time (h)	1	4	3	2
% plastic	80	50	2	14
% acid	20	0	0	10
Price per liter ($)	1.0	2.4	1.5	0.8

Write the paint production problem as a cost minimization and explain the decision variables, objective function, and contraints.

5.10. *Commodity Shipment*
Suppose a company has S_1 and S_2 tons of some commodity available at its two storage warehouses, and wishes to transport various amounts to its three retail stores where the demands are for D_1, D_2, and D_3 tons, respectively. Let C_{ij} be the nonnegative unit cost of shipping from the *i*th warehouse to the *j*th store.

(*a*) Write the linear programming formulation, the solution to which satisfies the demands at a minimum cost. What minimum condition is necessary for a feasible solution to exist?
(*b*) Suppose that $D = D_1 + D_2 + D_3$ tons are available at the factory for shipping to the two warehouses, where it will be stored for a while before being shipped to the retail outlets. Storage costs at the two warehouses are a_1 and a_2 dollars per ton, and their capacities are k_1 and k_2 tons. Ignoring the costs of sending the commodity from the factory to the warehouses, formulate the model that will give the overall shipping plan that will satisfy the retail store demands at minimum total cost.

5.11. *Wheat Shipment*
You own two warehouses, W1 and W2, in which you can store wheat. You can buy your wheat from two sources, A and B. A will sell you up to 400 tons of wheat at $100/ton, B up to 50 tons at $75/ton. You must also pay transportation costs to ship your wheat to your warehouses. The current rate is $0.15/ton-km. The distances from each source to each warehouse are:

A to W1, 350 km; to W2, 200 km
B to W1, 200 km; to W2, 100 km

(*a*) You are already under contract with A for at least 100 tons of wheat. Assuming that you must supply at least 200 tons to W1 and 150 tons to W2, formulate this problem as a linear programming problem.

(*b*) Source B has become the agent of a third party, who is willing to sell up to 350 tons of wheat for $175/ton. Incorporate this change into your linear program.

5.12. *Heavy Metals Inc.*
HMI sells two rhodium-iridium alloys. The high rhodium blend is 60% rhodium and 40% iridium and the high iridium blend is 20% rhodium, 80% iridium. The two alloys sell for p_r and p_i \$/kg.

Two grades of rhodium ore are available. The total cost including processing is $\$C_1$ /kg for the higher grade and $\$C_2$ for the lower grade, where $C_2 > C_1$. Unfortunately there is only enough of the higher grade to produce L kg of rhodium, whereas there is an abundant supply of the lower grade. Iridium on the other hand is available in any quantity at a uniform cost of $\$C_3$/kg including processing.

HMI can afford to spend D dollars in buying and processing the ores. The market will absorb as much of either alloy as HMI can produce. HMI wants to know how much of each alloy to produce and of each ore to purchase. Formulate this as an LP problem, carefully specifying and explaining the decision variables, the objective function, and constraints.

5.13. *Red Cross Relief*
The Red Cross wishes to maximize the amount of relief material it can move from a port to an inland disaster area using the available semi-trailers and vans. The constraints on the situation are

	Semis	Vans
Number available	100	200
Fuel use, gal/day	50	12
Capacity, tons/truck/day	40	15

Additionally, only 4500 gal/day of fuel are available to the relief operations, and the capacity of the ferry that replaces a bridge that has fallen down is 250 vehicles/day.

(*a*) Formulate as an LP problem, specifying and explaining the decision variables, the objective function, and constraints.

(*b*) Graph the activities in the fuel-ferry resource space, and sketch the isoquant for 120 tons of material transported.

(*c*) Can the problem still be formulated as an LP if, when more than 30 semis are used, deterioration of the road will require additional semis to be loaded with only 30 tons/truck/day?

(*d*) Repeat (c), but supposing instead that, when more than 60 vans are used, the road will pack down and additional vans can carry 20 tons/truck/day.

5.14. *SMC Factory*
As manager in charge of formulating batches of SMC (sheet molding compound, a plastic composite material used structurally in cars), you need to minimize costs subject to constraints on the properties of the material. SMC can be made by blending the following materials:

	Cost (\$/lb)	Tensile (ksi)	Modulus (msi)	Specific gravity
Polyester	0.55	8	0.4	1.10
Vinylester	1.05	10	0.5	1.10
E-glass fiber	1.00	500	10	2.54
Carbon fiber	25.00	300	30	1.70
Kevlar fiber	21.00	400	18	1.44

An order has come in for a batch of SMC meeting the following specification:

$$100 \text{ ksi} < \text{Tensile strength} < 200 \text{ ksi}$$
$$\text{Tensile modulus} > 10 \text{ msi}$$
$$\text{Specific gravity} < 1.9$$
$$\text{Total volume \% of fiber} < 78.5\%$$

Assuming that all material properties combine according to their volume percentage, formulate the design of this SMC as a linear programming problem.

5.15. *Transportation Funding*
A Department of Transportation has asked you to suggest the allocation of the DOT's budget of $500M between highway, rapid rail, bus, and innovative urban systems. Political considerations require that urban populations receive at least $300M in benefits, suburbs at least $200M and rural areas at least $100M.

As a rule of thumb, a $1M investment yields total user benefits of

$1.25M for $1M spent on highways
$1.0M for $1M spent on rapid rail
$1.5M for $1M spent on bus
$.8M for $1M spent on innovative systems

Your research also indicates that the distribution of benefits for each transportation mode is

	Urban	Suburban	Rural
Highway	20%	40%	40%
Rapid rail	30%	50%	10%
Bus	30%	40%	30%
Innovative urban systems	60%	40%	—

(*a*) Formulate an LP problem to maximize total user benefits subject to the Department's constraints.

(*b*) Reformulate the problem to reflect the limited number of "good" sites available for rapid rail development; if more than $75M is spent on rapid rail, additional rapid rail funds are spent at "bad" sites, producing only $.7M benefits for a $1M investment.

5.16. *Beans and Corn*
A farmer is planning to produce beans and corn. The unit profit on a pound of beans is $0.50 for the first 1000 lbs. After this the unit profit is $0.35. Corn profits are divided into three categories. The unit profit for the first 800 lbs is $0.75. The next 400 lbs yield $0.40/lb. Any additional lbs of corn will yield $0.25.

The farmer has a limited storage capacity and thus can produce no more than 5000 lbs of produce, in any combination of beans and corn.

The amount of land available is also a restriction on the output. The farmer has 10 acres of workable land for any combination of the two crops. He has determined that every 2000 lbs of beans requires 4 acres. Every 2000 lbs of corn requires 5 acres of land.

The farmer must use a cooperative machine to harvest these crops. He will have access to this machine for 20 hr. He can use it to harvest 200 lbs of beans/hr or 250 lbs of corn/hr.

Formulate an LP to determine the optimal quantity of each crop the farmer should plant, being sure to define your variables.

5.17. *Chemical Company*
A company makes two chemicals, 1 and 2. These chemicals can be manufactured by three different processes using two different raw materials and a fuel. Production data are given below. Formulate an LP model to estimate the time required to run each process in order to maximize the total amount of chemicals manufactured.

	Requirements per unit time			Output per unit time	
Process	**Raw material 1**	**Raw material 2**	**Fuel**	**Chemical 1**	**Chemical 2**
1	9	5	50	9	6
2	6	8	75	7	10
3	4	11	100	10	6
Amount Available	200	400	1850		

5.18. *Irrigation Problem*
A fertile but arid area in Southern California is to be cleared, leveled, fertilized, irrigated, and turned into prime farmland. Dealing specifically with the irrigation problem, a linear program is written to select, from a large number of possible canal and pipeline routes, the cheapest set of irrigation links that will provide adequate water to the entire area.

Given the characteristics of LP, explain how each of the following four considerations can be included in the LP, or why it cannot be included.

(*a*) It is futile to construct feeder line #2 with a greater capacity than link #1, which connects link #2 to the main supply line.
(*b*) The cost of a pipeline is a nonlinear function of its capacity.
(*c*) As bulldozer-hours/week increase, the cost of a bulldozer-hour increases.
(*d*) Water demand is a function of weather and therefore varies from year to year.

CHAPTER 6

SENSITIVITY ANALYSIS

6.1 CONCEPT

Sensitivity analysis is the process of investigating the dependence of an optimal solution to changes in the way a problem is formulated. Doing a sensitivity analysis is a key part of the design process, equal in importance to the optimization process itself.

The significance of sensitivity analysis stems from the fact that the mathematical problem we solve in any optimization is only an approximation of the real problem. The exact solution we obtain and use to represent reality is thus not an exact solution to the real problem of design. At best, the optimization process provides a good approximation to the best design of a real system.

None of our mathematical models will ever represent systems exactly. All these representations are approximations in some way. They each differ from reality in any or all of the following three ways:

- Structurally, because the overall nature of the equations does not correspond precisely to the actual situation.
- Parametrically, as we are not able to determine all coefficients precisely.
- Probabilistically, in that we typically assume that the situation is deterministic when it is generally variable.

Structural differences arise as a matter of course in the modeling process. The way we typically construct a mathematical model of a system is to imagine some form we believe is appropriate or useful, and then to match the real situation