CHAPTER 18

Simulation Modeling

Simulation is the next best thing to observing a real system. It allows us to collect pertinent information about the behavior of the system with the passage of time. Simulation is not an optimization technique. Rather, it is used to estimate the measures of performance of a modeled system.

Modern simulation typically deals with situations that can be described in the context of a waiting line. This is not a limitation on the use of simulation because practically any operational situation can be viewed in some form as a waiting line. This is the reason simulation has enjoyed tremendous applications in communication networks, manufacturing, inventory control, consumer behavior, economic forecasting, biomedical systems, and war strategies and tactics.

A forerunner to present-day simulation is the Monte Carlo technique, a scheme that is aimed at estimating stochastic or deterministic parameters based on random sampling. The main difference between the two techniques is that in Monte Carlo the time element is not a pertinent factor. Examples of Monte Carlo applications include estimation of the area under a curve or, more generally, evaluation of multiple integrals, estimation of the constant π ($\cong$ 3.14159), and matrix inversion.

Simulation is a statistical experiment, and hence its output must be interpreted by appropriate statistical tests. This important point is emphasized throughout the chapter.

18.1 MONTE CARLO SIMULATION

This section uses an example to demonstrate the Monte Carlo technique. The objective of the example is to emphasize the statistical nature of the simulation experiment.

Example 18.1-1

We will use Monte Carlo sampling to estimate the area of a circle whose equation is

$$(x - 1)^2 + (y - 2)^2 = 25$$

The radius of the circle is $r = 5$ cm, and its center is $(x, y) = (1, 2)$.

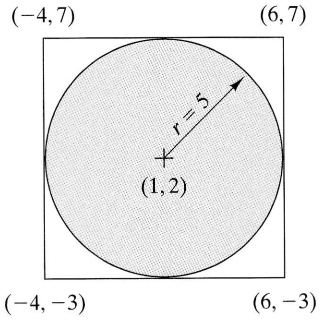


FIGURE 18.1
Monte Carlo estimation of the area of a circle

The procedure for estimating the area requires enclosing the circle "snugly" in a square whose side equals the diameter of the circle as shown in Figure 18.1. The corner points are determined from the geometry of the square.

The estimation of the area of the circle is based on the assumption that all the points in the square are equally likely to occur. Thus, if out of a random sample of n points in the square, m points are within the circle, then

$$\begin{pmatrix}\text{Estimate of the}\\ \text{area of the circle}\end{pmatrix} \cong \frac{m}{n}\begin{pmatrix}\text{Area of}\\ \text{the square}\end{pmatrix} = \frac{m}{n}(10 \times 10)$$

To ensure that all the points in the square are equally likely to occur, we represent the coordinates x and y of a point in the square by the following *uniform* distributions:

$$f_1(x) = \frac{1}{10}, \; -4 \le x \le 6$$

$$f_2(y) = \frac{1}{10}, \; -3 \le y \le 7$$

A sampled point (x, y) based on the distribution $f_1(x)$ and $f_2(y)$ guarantees that all points in the square are equally likely to be selected.

The procedure for determining a sample (x, y) starts with generating independent 0-1 random numbers and then mapping them on the (x, y)-axis. The 0-1 random numbers are determined using the following uniform distribution:

$$f(z) = \begin{cases} 1, & 0 \le z \le 1 \\ 0, & \text{otherwise} \end{cases}$$

Table 18.1 gives a small list of (0, 1) random numbers. These numbers are determined using special arithmetic operations that generate statistically independent values based on the uniform distribution $f(z)$, as will be explained in Section 18.4.

Given a pair of 0-1 random numbers, R_1 and R_2, a random point (x, y) in the square is determined as

$$x = -4 + [6 - (-4)]R_1 = -4 + 10R_1$$
$$y = -3 + [7 - (-3)]R_2 = -3 + 10R_2$$

To demonstrate the application of the procedure, consider $R_1 = .0589$ and $R_2 = .6733$. Then

$$x = -4 + 10R_1 = -4 + 10 \times .0589 = -3.411$$
$$y = -3 + 10R_2 = -3 + 10 \times .6733 = 3.733$$

TABLE 18.1

.0589	.3529	.5869	.3455	.7900	.6307
.6733	.3646	.1281	.4871	.7698	.2346
.4799	.7676	.2867	.8111	.2871	.4220
.9486	.8931	.8216	.8912	.9534	.6991
.6139	.3919	.8261	.4291	.1394	.9745
.5933	.7876	.3866	.2302	.9025	.3428
.9341	.5199	.7125	.5954	.1605	.6037
.1782	.6358	.2108	.5423	.3567	.2569
.3473	.7472	.3575	.4208	.3070	.0546
.5644	.8954	.2926	.6975	.5513	.0305

This point falls inside the circle because

$$(-3.411 - 1)^2 + (3.733 - 2)^2 = 22.46 < 25$$

Next, we investigate the effect of random sampling on the accuracy of estimating the area of the circle. We can increase the reliability of the estimate by increasing the sample size and/or using replications, the same procedures employed in ordinary statistical experiments.

Because the computations associated with each sample, though simple, are voluminous and tedious, Excel template ch18Circle.xls (with VBA macros) is developed to carry out these computations. The input data include the circle radius, r, and its center, (cx, cy), together with sample size, n, and the number of replications, N. The entry *Steps* in cell E4 allow executing several sample sizes in the same run. Thus, if $n = 30{,}000$ and Steps $= 3$, the template will automatically produce output for $n = 30{,}000$, $60{,}000$, $90{,}000$.

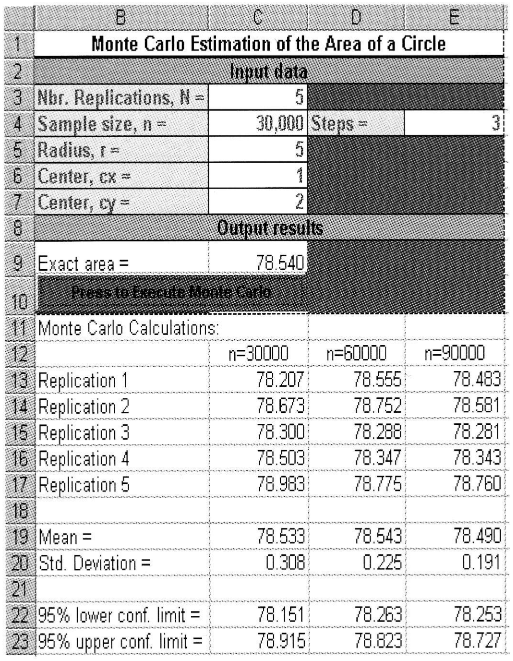
Figure 18.2 summarizes the results for Steps $= 3$ and $N = 5$ replications. The exact area is 78.54 cm^2 and the Monte Carlo results show that the mean estimated area for the three sample sizes varies from $\overline{A} = 78.533$ to $\overline{A} = 78.490$ cm^2. We note also that the standard deviation decreases from $s = .308$ for $n = 30{,}000$ to $s = .191$ for $n = 90{,}000$, an indication that accuracy of the results, generally, increases with the increase in the sample size.

Note that each time you press the command button `Press to Execute Monte Carlo`, new estimates are realized because Excel refreshes the random number generator to a different sequence.

Because of the random variation in the output of the experiment, it is necessary to express the results as a confidence interval. Letting $\overline{A}$ and s be the mean and variance of N replications, then, for a confidence level α, the confidence interval for the true area A is

$$\overline{A} - \frac{s}{\sqrt{N}} t_{\frac{\alpha}{2},N-1} \leq A \leq \overline{A} + \frac{s}{\sqrt{N}} t_{\frac{\alpha}{2},N-1}$$

The parameter $t_{\frac{\alpha}{2},N-1}$ is determined from the t-distribution tables given a confidence level α and $N - 1$ degrees of freedom (see the t-table in Appendix C). (Note that N equals the number of replications, which is distinct from n, the sample size.) In terms of the present experiment, we are interested in establishing the confidence interval based on the largest sample size (i.e., $n = 90{,}000$). Given $N = 5$, $\overline{A} = 78.490$ cm^2, and $s = .191$ cm^2, $t_{.025,4} = 2.776$, and the resulting 95% confidence interval is $78.25 \leq$



	B	C	D	E
1	Monte Carlo Estimation of the Area of a Circle			
2	Input data			
3	Nbr. Replications, N =	5		
4	Sample size, n =	30,000	Steps =	3
5	Radius, r =	5		
6	Center, cx =	1		
7	Center, cy =	2		
8	Output results			
9	Exact area =	78.540		
10	Press to Execute Monte Carlo			
11	Monte Carlo Calculations:			
12		n=30000	n=60000	n=90000
13	Replication 1	78.207	78.555	78.483
14	Replication 2	78.673	78.752	78.581
15	Replication 3	78.300	78.288	78.281
16	Replication 4	78.503	78.347	78.343
17	Replication 5	78.983	78.775	78.760
18				
19	Mean =	78.533	78.543	78.490
20	Std. Deviation =	0.308	0.225	0.191
21				
22	95% lower conf. limit =	78.151	78.263	78.253
23	95% upper conf. limit =	78.915	78.823	78.727

FIGURE 18.2

Excel output of Monte Carlo estimation of the area of a circle

$A \leq 78.73$. (The Excel template automatically computes the 95% confidence interval.) Note that, in general, the value of N should be at least 5 to realize reasonable accuracy in the estimation of the confidence interval.

The discussion in Example 18.1-1 poses two questions regarding the simulation experiment:

1. How large should the sample size, n, be?
2. How many replications, N, are needed?

There are some formulas in statistical theory for determining n and N, and both depend on the nature of the simulation experiment as well as the desired confidence level. However, as in any statistical experiment, the golden rule is that higher values of n and N mean more reliable simulation results. In the end, the sample size will depend on the cost associated with conducting the simulation experiment. Generally speaking, however, a selected sample size is considered "adequate" if it produces a relatively "small" standard deviation.

PROBLEM SET 18.1A

1. In Example 18.2-1, estimate the area of the circle using the first two columns of the (0, 1) random numbers in Table 18.1. (For convenience, go down each column, selecting R_1 first and then R_2.) How does this estimate compare with the ones given in Figure 18.2?
2. Suppose that the equation of a circle is

$$(x - 3)^2 + (y + 2)^2 = 16$$

   (a) Define the corresponding distributions $f(x)$ and $f(y)$, and then show how a sample point (x, y) is determined using the (0, 1) random pair (R_1, R_2).

   (b) Use the Excel template to estimate the area given $n = 100{,}000$ and $N = 10$. Then compute the associated 95% confidence interval.

3. Use Monte Carlo sampling to estimate the area of the lake shown in Figure 18.3. Base the estimate on the first two columns of (0, 1) random numbers in Table 18.1.

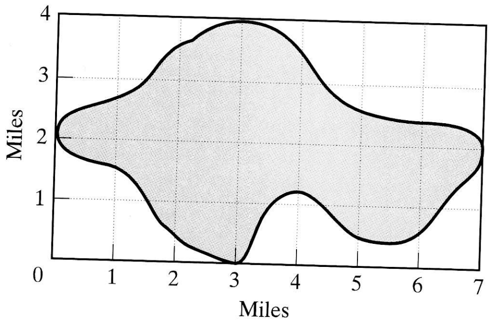


FIGURE 18.3

Lake map for Problem 3, Set 18.1a

4. Consider the game in which two players, Jan and Jim, take turns in tossing a fair coin. If the outcome is heads, Jim gets \$10 from Jan. Otherwise, Jan gets \$10 from Jim.

   (a) How is the game simulated as a Monte Carlo experiment?

   (b) Run the experiment for 5 replications of 10 tosses each to determine Jan's pay. Use the first five columns of the (0, 1) random numbers in Table 18.1, with each column corresponding to one replication.

   (c) Establish a 95% confidence interval on Jan's winnings.

   (d) Compare the confidence interval in (c) with Jan's expected theoretical winnings.

5. Consider the following definite integral:

$$\int_0^1 x^2 dx$$

   (a) Develop the Monte Carlo experiment to estimate the integral.

   (b) Use the first four columns in Table 18.1 to evaluate the integral based on 4 replications of size 5 each. Compute a 95% confidence interval, and compare it with the exact value of the integral.

6. Simulate five wins or losses of the following game of craps: The player rolls two fair dice. If the outcome sum is 7 or 11, the player wins \$10. Otherwise, the player records the resulting sum (called *point*) and keeps on rolling the dice until the outcome sum matches

the recorded *point*, in which case the player wins \$10. If a 7 is obtained, the player loses \$10.

7. The lead time for receiving an order can be 1 or 2 days, with equal probabilities. The demand *per day* assumes the values 0, 1, and 2 with the respective probabilities of .2, .7, and .1. Use the random numbers in Table 18.1 (starting with column 1) to estimate the joint distribution of the demand and lead time. From the joint distribution, estimate the pdf of demand during lead time. (*Hint*: The demand during lead time assumes discrete values from 0 to 4.)

8. Consider the Buffon needle experiment. A horizontal plane is ruled with parallel lines spaced D cm apart. A needle of length d cm ($d < D$) is dropped randomly on the plane. The objective of the experiment is to determine the probability that either end of the needle touches or crosses one of the lines. Define

$$h = \text{Perpendicular distance from the needle center to a (parallel) line}$$

$$\theta = \text{Inclination angle of the needle with a line}$$

(a) Show that the needle will touch or cross a line only if

$$h \le \frac{d}{2}\sin\theta,\ 0 \le h \le \frac{D}{2},\ 0 \le \theta \le \pi$$

(b) Design the Monte Carlo experiment, and provide an estimate of the desired probability.

(c) Use Excel to obtain 4 replications of size 10 each of the desired probability. Determine a 95% confidence interval for the estimate. Assume $D = 20$ cm and $d = 10$ cm.

(d) Prove that the theoretical probability is given by the formula

$$p = \frac{2d}{\pi D}$$

(e) Use the result in (c) together with this formula in (d) to estimate π.

18.

18.

18.2 TYPES OF SIMULATION

The execution of present-day simulation is based generally on the idea of sampling used with the Monte Carlo method. It differs in that it is concerned with the study of the behavior of real systems *as a function of time.* Two distinct types of simulation models exist.

1. **Continuous models** deal with systems whose behavior changes *continuously* with time. These models usually use difference-differential equations to describe the interactions among the different elements of the system. A typical example deals with the study of world population dynamics.

2. **Discrete models** deal primarily with the study of waiting lines, with the objective of determining such measures as the average waiting time and the length of the queue. These measures change only when a customer enters or leaves the system. At all other instants, nothing from the standpoint of collecting statistics occurs in the system. The instants at which changes take place occur at discrete points in time, giving rise to the name **discrete event simulation**.

This chapter presents the basics of discrete event simulation, including a description of the components of a simulation model, collection of simulation statistics, and the statistical aspect of the simulation experiment. The chapter also emphasizes the role of the computer and simulation languages in the execution of simulation models.

PROBLEM SET 18.2A

1. Categorize the following situations as either discrete or continuous (or a combination of both). In each case, specify the objective of developing the simulation model.
   - **(a)** Orders for an item arrive randomly at a warehouse. An order that cannot be filled immediately from available stock must await the arrival of new shipments.
   - **(b)** World population is affected by the availability of natural resources, food production, environmental conditions, educational level, health care, and capital investments.
   - **(c)** Goods arrive on pallets at a receiving bay of an automated warehouse. The pallets are loaded on a lower conveyor belt and lifted through an up-elevator to an upper conveyor that moves the pallets to corridors. The corridors are served by cranes that pick up the pallets from the conveyor and place them in storage bins.
2. Explain why you would agree or disagree with the following statement: "Most discrete event simulation models can be viewed in some form or another as queuing systems consisting of *sources* from which customers are generated, *queues* where customers may wait, and *facilities* where customers are served."

18.3 ELEMENTS OF DISCRETE EVENT SIMULATION

This section introduces the concept of events in simulation and shows how the statistics of the simulated system are collected.

18.3.1 Generic Definition of Events

All discrete-event simulations describe, directly or indirectly, queuing situations in which customers arrive, wait in a queue if necessary, and then receive service before they depart the system. In general, any discrete-event model is composed of a network of interrelated queues.

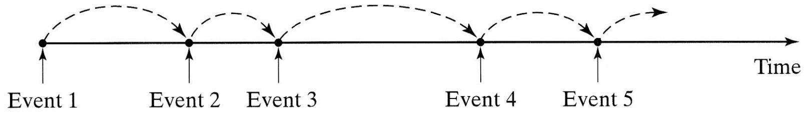
Given that a discrete-event model is in reality a composite of queues, collection of simulation statistics (e.g., queue length and status of the service facility) need only take place when a customer arrives or leaves the facility. This means that two principal events control the simulation model: arrivals and departures. These are the only two instants in time at which we need to examine the system. At all other instants, no changes affecting the statistics of the system take place.

Example 18.3-1

Metalco Jobshop receives two types of jobs: regular and rush. All jobs are processed on two consecutive machines with ample buffer areas. Rush jobs always assume nonpreemptive priority over regular jobs. Identify the events of the situation.

FIGURE 18.4

Example of the occurrence of simulation events on the time scale



This situation consists of two tandem queues corresponding to the two machines. At first thought, one may be inclined to identify the events of the situation as follows:

*A*11: A regular job arrives at machine 1.

*A*21: A rush job arrives at machine 1.

*D*11: A regular job departs machine 1.

*D*21: A rush job departs machine 1.

*A*12: A regular job arrives at machine 2.

*A*22: A rush job arrives at machine 2.

*D*12: A regular job departs machine 2.

*D*22: A rush job departs machine 2.

In reality, we only have exactly two events: an arrival of a (new) job at the shop and a departure of a (completed) job from a machine. First notice that events *D*11 and *A*12 are actually one and the same. The same applies to *D*21 and *A*22. Next, in discrete simulation we can use one event (arrival or departure) for both types of jobs and simply "tag" the event with an **attribute** that identifies the job type as either regular or rush. (We can think of the attribute in this case as a *personal identification number* and, indeed, it is.) Given this reasoning, the events of the model reduce to (1) an arrival *A* (at the shop) and (2) a departure *D* (from a machine). The actions associated with the departure event will depend on the machine at which they occur.

Having defined the basic events of a simulation model, we show how the model is executed. Figure 18.4 gives a schematic representation of typical occurrences of events on the simulation time scale. After all the actions associated with a current event have been performed, the simulation advances by "jumping" to the next chronological event. In essence, the execution of the simulation occurs at the instants at which the events occur.

How does the simulation determine the occurrence time of the events? The arrival events are separated by the interarrival time (the interval between successive arrivals), and the departure events are a function of the service time in the facility. These times may be deterministic (e.g., a train arriving at a station every 5 minutes) or probabilistic (e.g., the random arrival of customers at a bank). If the time between events is deterministic, the determination of their occurrence times is straightforward. If it is probabilistic, we use a special procedure to sample from the corresponding probability distribution. This point is discussed in the next section.

PROBLEM SET 18.3A

1. Identify the discrete events needed to simulate the following situation: Two types of jobs arrive from two different sources. Both types are processed on a single machine, with priority given to jobs from the first source.
2. Jobs arrive at a constant rate at a carousel conveyor system. Three service stations are spaced equally around the carousel. If the server is idle when a job arrives at the sta-

tion, the job is removed from the conveyor for processing. Otherwise, the job continues to rotate about the carousel until a server becomes available. A processed job is stored in an adjacent shipping area. Identify the discrete events needed to simulate this situation.

3. Cars arrive at a two-lane, drive-in bank, where each lane can house a maximum of four cars. If the two lanes are full, arriving cars seek service elsewhere. If at any time one lane is at least two cars longer than the other, the last car in the longer lane will jockey to the last position in the shorter lane. The bank operates the drive-in facility from 8:00 A.M. to 3:00 P.M. each workday. Define the discrete events for the situation.
4. The cafeteria at Elmdale Elementary provides a single-tray, fixed-menu lunch to all its pupils. Students arrive at the dispensing window every 30 seconds. It takes 18 seconds to receive the lunch tray. Map the arrival-departure events on the time scale for the first five pupils.

18.3.2 Sampling from Probability Distributions

Randomness in simulation arises when the interval, t, between successive events is probabilistic. This section presents three methods for generating successive random samples ($t = t_1, t_2, \ldots$) from a probability distribution $f(t)$.

1. Inverse method.
2. Convolution method.
3. Acceptance-rejection method.

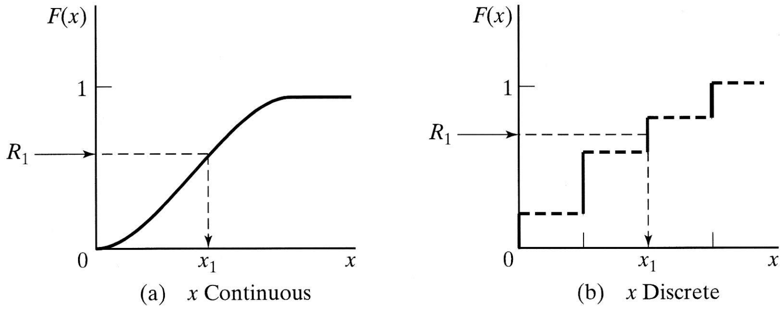
The inverse method is particularly suited for analytically tractable probability density functions, such as the exponential and the uniform. The remaining two methods deal with more complex cases, such as the normal and the Poisson. All three methods are rooted in the use of independent and identically distributed uniform (0, 1) random numbers.

Inverse Method. Suppose that it is desired to obtain a random sample x from the (continuous or discrete) probability density function $f(x)$. The inverse method first determines a closed form expression of the cumulative density function $F(x) = P\{y \leq x\}$, where $0 \leq F(x) \leq 1$, for all defined values of y. Given R is a random value obtained from a uniform (0, 1) distribution, and assuming that F^{-1} is the inverse of F, the steps of the method are as follows:

Step 1. Generate the (0, 1) random number, R.
Step 2. Compute the desired sample, $x = F^{-1}(R)$.

Figure 18.5 illustrates the procedures for both a continuous and a discrete random distribution. The uniform (0, 1) random value R_1 is projected from the vertical $F(x)$-scale to yield the desired sample value x_1 on the horizontal scale.

The validity of the proposed procedure rests on showing that the random variable $z = F(x)$ is uniformly distributed in the interval $0 \leq z \leq 1$, as the following theorem proves.



(a) *x* Continuous (b) *x* Discrete

FIGURE 18.5

Sampling from a probability distribution by the inverse method

Theorem 18.3-1. *Given the cumulative density function* $F(x)$ *of the random variable* x, $-\infty \le x \le \infty$, *the random variable* $z = F(x)$, $0 \le z \le 1$, *has the following uniform 0-1 density function*:

$$g(z) = 1, 0 \le z \le 1$$

Proof. The random variable is uniformly distributed if, and only if,

$$P\{z \le Z\} = Z, 0 \le Z \le 1$$

This result applies to $F(x)$ because

$$P\{z \le Z\} = P\{F(x) \le Z\} = P\{x \le F^{-1}(Z)\} = F[F^{-1}(Z)] = Z$$

Additionally, $0 \le Z \le 1$ because $0 \le P\{z \le Z\} \le 1$.

Example 18.3-2 (Exponential Distribution)

The time, t, between customers arrivals at a facility is represented by an exponential distribution with mean $E\{t\} = \frac{1}{\lambda}$ time units—that is,

$$f(t) = \lambda e^{-\lambda t}, t > 0$$

Determine a random sample t from $f(t)$.

The cumulative density function is determined as

$$F(t) = \int_0^t \lambda e^{-\lambda x} dx = 1 - e^{-\lambda t}, t > 0$$

Setting $R = F(t)$, we can solve for t, which yields

$$t = -\left(\frac{1}{\lambda}\right)\ln(1 - R)$$

Because $1 - R$ is the complement of R, we may replace $\ln(1 - R)$ with $\ln(R)$.

In terms of simulation, the result means that arrivals are spaced t time units apart. For example, given $\lambda = 4$ customers per hour and $R = .9$, the time period until the next arrival occurs is computed as

$$t_1 = -\left(\frac{1}{4}\right)\ln(1 - .9) = .577 \text{ hour} = 34.5 \text{ minutes}$$

The values of R used to obtain successive samples must be selected *randomly* from a uniform (0, 1) distribution. We will show in Section 18.4 how these (0, 1) random values are generated during the course of the simulation.

PROBLEM SET 18.3B

1. In Example 18.3-2, suppose that the first customer arrives at time 0. Use the first three random numbers in column 1 of Table 18.1 to generate the arrival times of the next three customers and graph the resulting events on the time scale.

2. *Uniform Distribution.* Suppose that the time needed to manufacture a part on a machine is given by the following uniform distribution:

$$f(t) = \frac{1}{b - a}, \ a \leq t \leq b$$

Determine an expression for the sample t given the random number R.

3. Jobs are received randomly at a one-machine shop. The time between arrivals is exponential with a mean of 2 hours. The time needed to manufacture a job is uniform between 1.1 and 2 hours. Assuming that the first job arrives at time 0, determine the arrival and departure time for the first five jobs using the (0, 1) random numbers in column 1 of Table 18.1.

4. The demand for an expensive spare part of a passenger jet is 0, 1, 2, or 3 units per month with probabilities .2, .3, .4, and .1, respectively. The airline maintenance shop starts operation with a stock of 5 units and will bring the stock level back to 5 units immediately after it drops below 3 units.

   (a) Devise the procedure for determining demand samples.

   (b) How many months will elapse until the first replenishment occurs? Use successive values of R from the first column in Table 18.1.

5. In a simulation situation, TV units are inspected for possible defects. There is an 80% chance that a unit will pass inspection, in which case it is sent to packaging. Otherwise, the unit is repaired. We can represent the situation symbolically in one of two ways.

```
goto REPAIR/.2, PACKAGE/.8
goto PACKAGE/.8, REPAIR/.2
```

These two representations appear equivalent. Yet, when a given sequence of (0, 1) random numbers is applied to the two representations, different decisions (REPAIR or PACKAGE) may result. Explain why.

6. A player tosses a fair coin repeatedly until a head occurs. The associated payoff is 2^n, where n is the number of tosses until a head comes up.

   (a) Devise the sampling procedure of the game.

   (b) Use the random numbers in column 1 of Table 18.1 to determine the cumulative payoff after two heads occur.

7. *Triangular Distribution.* In simulation, the lack of data may make it impossible to determine the probability distribution associated with a simulation activity. In most of these situations, it may be easy to describe the desired variable by estimating its smallest, most likely, and largest values. These three values are sufficient to define a triangular distribution, which can then be used as a "rough cut" estimation of the real distribution.

   (a) Develop the formula for sampling from the following triangular distribution whose parameters are a, b, and c, where $a < b < c$:

$$f(x) = \begin{cases} \dfrac{2(x-a)}{(b-a)(c-a)}, & a \leq x \leq b \\ \dfrac{2(c-x)}{(c-b)(c-a)}, & b \leq x \leq c \end{cases}$$

(b) Generate three samples from a triangular distribution with parameters (1, 3, 7) using the first three random numbers in column 1 of Table 18.1.

8. Consider a probability distribution that consists of a rectangle flanked on the left and right sides by two symmetrical right triangles. The respective ranges for the triangle on the left, the rectangle, and the triangle on the right are $[a, b]$, $[b, c]$, and $[c, d]$, $a < b < c < d$. Each triangle has the same height as the rectangle.

(a) Develop a sampling procedure

(b) Determine five samples with $(a, b, c, d) = (1, 2, 4, 6)$ using the first five random numbers in column 1 of Table 18.1.

9. *Geometric Distribution.* Show how a random sample can be obtained from the following geometric distribution:

$$f(x) = p(1 - p)^x,\ x = 0, 1, 2, \ldots$$

The parameter x is the number of (Bernoulli) failures until a success occurs, and p is the probability of a success, $0 < p < 1$. Generate five samples for $p = .6$ using the first 5 random numbers in column 1 of Table 18.1.

10. *Weibull Distribution.* Show how a random sample can be obtained from the Weibull distribution whose pdf is defined as

$$f(x) = \alpha\beta^{-\alpha}x^{\alpha-1}e^{-(x/\beta)^{\alpha}},\ x > 0$$

where $\alpha > 0$ is the shape parameter, and $\beta > 0$ is the scale parameter.

Convolution Method. The idea of the convolution method is to express the desired sample as the statistical sum of other easy-to-sample random variables. Typical among these distributions are the Erlang and the Poisson whose sample can be obtained from the exponential distribution samples.

Example 18.3-3 (Erlang Distribution)

The m-Erlang random variable is defined as the statistical sum (convolutions) of m independent and identically distributed exponential random variables. Let y represent the m-Erlang random variable; then

$$y = y_1 + y_2 + \cdots + y_m$$

where y_i, $i = 1, 2, \ldots, m$, are independent and identically distributed exponential random variables whose probability density function is defined as

$$f(y_i) = \lambda e^{-\lambda y_i},\ y_i > 0,\ i = 1, 2, \ldots, m$$

From Example 18.3-2, the ith exponential sample is

$$y_i = -\left(\frac{1}{\lambda}\right)\ln(R_i),\ i = 1, 2, \ldots, m$$

Thus, the m-Erlang sample is computed as

$$\begin{aligned} y &= -\left(\frac{1}{\lambda}\right)\{\ln(R_1) + \ln(R_2) + \ldots + \ln(R_m)\} \\ &= -\left(\frac{1}{\lambda}\right)\ln(R_1 R_2 \ldots R_m) \end{aligned}$$

To illustrate the use of the formula, suppose that $m = 3$, and $\lambda = 4$ events per hour. The first 3 random numbers in column 1 of Table 18.1 yield $R_1R_2R_3 = (.0589)(.6733)(.4799) = .0190$, which yields

$$y = -(\tfrac{1}{4})\ln(.019) = .991 \text{ hour}$$

Example 18.3-4 (Poisson Distribution)

Section 17.3.1 shows that if the distribution of the time between the occurrence of successive events is exponential, then the distribution of the number of events per unit time must be Poisson, and vice versa. We use this relationship to sample the Poisson distribution.

Assume that the Poisson distribution has a mean value of λ events per unit time. Then the time between events is exponential with mean $\frac{1}{\lambda}$ time units. This means that a Poisson sample, n, will occur during t time units if, and only if,

$$\text{Period until event } n \text{ occurs } \leq t < \text{ Period until event } n + 1 \text{ occurs}$$

This condition translates to

$$t_1 + t_2 + \dots t_n \leq t < t_1 + t_2 + \dots t_{n+1},\ n > 0$$
$$0 \leq t < t_1,\ n = 0$$

where t_i, $i = 1,2, \dots, n$, is a sample from the exponential distribution with mean $\frac{1}{\lambda}$. From the result in Example 18.3-3, we have

$$-\left(\frac{1}{\lambda}\right)\ln(R_1R_2\dots R_n) \leq t < -\left(\frac{1}{\lambda}\right)\ln(R_1R_2\dots R_{n+1}),\ n > 0$$
$$0 \leq t < -\left(\frac{1}{\lambda}\right)\ln(R_1),\ n = 0$$

which reduces to

$$R_1R_2\dots R_n \geq e^{-\lambda t} > R_1R_2\dots R_{n+1},\ n > 0$$
$$1 \geq e^{-\lambda t} > R_1,\ n = 0$$

To illustrate the implementation of the sampling process, suppose that $\lambda = 4$ events per hour and that we wish to obtain a sample for a period $t = .5$ hour. This gives $e^{-\lambda t} = .1353$. Using the random numbers in column 1 of Table 18.1, we note that $R_1 = .0589$ is less than $e^{-\lambda t} = .1353$. Hence, the corresponding sample is $n = 0$.

Example 18.3-5 (Normal Distribution)

The Central Limit Theorem (see Section 12.4.4) states that the sum (convolution) of n independent and identically distributed random variables becomes asymptotically normal as n becomes sufficiently large. We use this result to generate samples from normal distribution with mean μ and standard deviation σ.

Define

$$x = R_1 + R_2 + \dots + R_n$$

The random variable is asymptotically normal by the Central Limit Theorem. Given that the uniform (0, 1) random number R has a mean of $\frac{1}{2}$ and a variance of $\frac{1}{12}$, it follows that x has a mean of $\frac{n}{2}$ and a variance of $\frac{n}{12}$. Thus, a random sample, y, from a normal distribution with mean μ and standard deviation σ, $N(\mu, \sigma)$, can be computed from x as

$$y = \mu + \sigma\left(\frac{x - \frac{n}{2}}{\sqrt{\frac{n}{12}}}\right)$$

In practice, we take $n = 12$ for convenience, which reduces the formula to

$$y = \mu + \sigma(x - 6)$$

To illustrate the use of this method, suppose that we wish to generate a sample from $N(10, 2)$ (mean $\mu = 10$ and standard deviation $\sigma = 2$). Taking the sum of the first 12 random numbers in columns 1 and 2 of Table 18.1, we get $x = 6.1094$. Thus, $y = 10 + 2(6.1094 - 6) = 10.2188$.

The disadvantage of this procedure is that it requires generating 12 random numbers for each normal sample, which is computationally inefficient. A more efficient procedure calls for using the transformation

$$x = \sqrt{-2\ln(R_1)}\cos(2\pi R_2)$$

Box and Muller (1958) prove that x is a standard $N(0, 1)$. Thus, $y = \mu + \sigma x$ will produce a sample from $N(\mu, \sigma)$. The new procedure is efficient because Box and Muller additionally prove that the preceding formula will produce another $N(0, 1)$ sample simply by replacing $\cos(2\pi R_2)$ with $\sin(2\pi R_2)$. This means that two random numbers, R_1 and R_2, will generate *two* $N(0, 1)$ samples.

To illustrate the implementation of the Box-Muller procedure to the normal distribution $N(10, 2)$, the first two random numbers in column 1 of Table 18.1 yield the following $N(0, 1)$ samples:

$$x_1 = \sqrt{-2\ln(.0589)}\cos(2\pi \times .6733) \approx -1.103$$
$$x_2 = \sqrt{-2\ln(.0589)}\sin(2\pi \times .6733) \approx -2.109$$

Thus, the corresponding $N(10, 2)$ samples are

$$y_1 = 10 + 2(-1.103) = 7.794$$
$$y_2 = 10 + 2(-2.109) = 5.782$$

PROBLEM SET 18.3C[1]

1. In Example 18.3-3, compute an Erlang sample, given $m = 4$ and $\lambda = 5$ events per hour.
2. In Example 18.3-4, generate a Poisson samples during a 1-hour period given that the mean of the Poisson is 5 events per hour.
3. In Example 18.4-5, generate two samples from $N(8, 1)$ by using both the convolution method and the Box-Muller method.
4. Jobs arrive at Metalco jobshop according to a Poisson distribution, with a mean of six jobs per day. Received jobs are assigned to the five machining centers of the shop on a strict

[1] For all the problems of this set, use the random numbers in Table 18.1 starting with column 1.

rotational basis. Determine one sample of the interval between the arrival of jobs at the first machine center.

5. The ACT scores for the 1994 senior class at Springdale High are normal, with a mean of 27 points and a standard deviation of 3 points. Suppose that we draw a random sample of six seniors from that class. Use the Box-Muller method to determine the mean and standard deviation of the sample.
6. Psychology professor Yataha is conducting a learning experiment in which mice are trained to find their way around a restricted maze. The base of the maze is square. A mouse enters the maze at one of the four corners and must find its way through the maze to exit at the same point where it entered. The design of the maze is such that the mouse must pass by each of the remaining three corner points exactly once before it exits. The multipaths of the maze connect the four corners in a strict clockwise order. Professor Yataha estimates that the time the mouse takes to reach one corner point from another is uniformly distributed between 10 and 20 seconds, depending on the path it takes. Develop a sampling procedure for the time a mouse spends in the maze.
7. In Problem 6, suppose that once a mouse makes an exit from the maze, another mouse instantly enters. Develop a sampling procedure for the number of mice that exit the maze in 5 minutes.
8. *Negative Binomial.* Show how a random sample can be determined from the negative binomial whose distribution is given as

$$f(x) = C_x^{r+x-1} p^r (1 - p)^x, \ x = 0, 1, 2, \ldots$$

where x is the number of failures until the rth success occurs in a sequence of independent Bernoulli trials and p is the probability of success, $0 < p < 1$. (*Hint*: The negative binomial is the convolution of independent geometric samples. See Problem 9, Set 18.3b.)

Acceptance-Rejection Method. The acceptance-rejection method is designed for complex pdfs that cannot be handled by the preceding methods. The general idea of the method is to replace the complex pdf $f(x)$ with a more analytically manageable "proxy" pdf $h(x)$. Samples from $h(x)$ can then be used to sample the original pdf $f(x)$.

Define the **majorizing function** $g(x)$ such that it dominates $f(x)$ in its entire range—that is,

$$g(x) \geq f(x), \ -\infty < x < \infty$$

Next, define the proxy pdf, $h(x)$, by normalizing $g(x)$ as

$$h(x) = \frac{g(x)}{\int_{-\infty}^{\infty} g(y) d(y)}, \ -\infty < x < \infty$$

The steps of the acceptance-rejection method are thus given as

Step 1. Obtain a sample $x = x_1$ from $h(x)$ using the inverse or the convolution method.

Step 2. Obtain a (0, 1) random number R.

Step 3. If $R \leq \frac{f(x_1)}{g(x_1)}$, accept x_1 as a sample from $f(x)$. Otherwise, discard x_1 and return to step 1.

The validity of the method is based on the following equality:

$$P\{x \leq a | x = x_1 \text{ is accepted}, -\infty < x_1 < \infty\} = \int_{-\infty}^{a} f(y)dy, \ -\infty < a < \infty$$

This probability statement states that the sample $x = x_1$ that satisfies the condition of step 3 in reality is a sample from the original pdf $f(x)$, as desired.

The efficiency of the proposed method is enhanced by the decrease in the rejection probability of step 3. This probability depends on the specific choice of the majorizing function $g(x)$ and should decrease with the selection of a $g(x)$ that "majorizes" $f(x)$ more "snugly."

Example 18.3-6 (Beta Distribution)

Apply the acceptance-rejection to the following beta distribution:

$$f(x) = 6x(1 - x), \ 0 \leq x \leq 1$$

Figure 18.6 depicts $f(x)$ and a majorizing function $g(x)$.

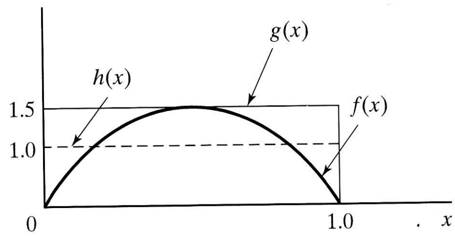


FIGURE 18.6

Majorizing function, $g(x)$, for the beta distribution, $f(x)$

The height of the majorizing function $g(x)$ equals the maximum of $f(x)$, which occurs at $x = .5$. Thus, the height of the rectangle is $f(.5) = 1.5$. This means that

$$g(x) = 1.5, \ 0 \leq x \leq 1$$

The proxy pdf $h(x)$, also shown in Figure 18.6, is computed as

$$h(x) = \frac{g(x)}{\text{Area under } g(x)} = \frac{1.5}{1 \times 1.5} = 1, \ 0 \leq x \leq 1$$

The following steps demonstrate the procedure using the (0, 1) random sequence in Table 18.1.

Step 1. $R = .0589$ gives the sample $x = .0589$ from $h(x)$.
Step 2. $R = .6733$.
Step 3. Because $\frac{f(.0589)}{g(.0589)} = \frac{.3326}{1.5} = .2217$ is less than $R = .6733$, we accept the sample $x_1 = .0589$.

To obtain a second sample, we continue as follows:

Step 1. Using $R = .4799$, we get $x = .4799$ from $h(x)$.
Step 2. $R = .9486$.

Step 3. Because $\frac{f(.4799)}{g(.4799)} = .9984$ is larger than $R = .9486$, we reject $x = .4799$ as a valid beta sample. This means that the steps must be repeated again with "fresh" random numbers until the condition of step 3 is satisfied.

The efficiency of the acceptance-rejection method is enhanced by selecting a majorizing function $g(x)$ that "jackets" $f(x)$ as tightly as possible while yielding an analytically tractable proxy $h(x)$. For example, the method will be more efficient if the rectangular majorizing function $g(x)$ in Figure 18.5 is replaced with a step-pyramid function (see Problem 2, Set 18.3d for an illustration). The larger the number of steps, the more tightly $g(x)$ will majorize $f(x)$, and hence the higher the probability of accepting a sample. However, a "tight" majorizing function generally entails additional computations which, if excessive, may offset the savings resulting from increasing the probability of acceptance.

PROBLEM SET 18.3D

1. In Example 18.3-6, continue the steps of the procedure until a valid sample is obtained. Use the (0, 1) random numbers in Table 18.1 in the same order in which they are used in the example.
2. Consider the beta pdf of Example 18.3-6. Determine a two-step pyramid majorizing function $g(x)$ with two equal jumps each of height $\frac{1.5}{2} = .75$. Obtain one beta sample based on the new majorizing function using the same (0, 1) random sequence in Table 18.1 that was employed in Example 18.3-6. The conclusion, in general, is that a tighter majorizing function will increase the probability of acceptance. Observe, however, that the amount of the computations associated with the new function is larger.
3. Determine the functions $g(x)$ and $h(x)$ for applying the acceptance-rejection method to the following function:

$$f(x) = \frac{\sin(x) + \cos(x)}{2}, \; 0 \le x \le \frac{\pi}{2}.$$

   Use the (0, 1) random numbers from column 1 in Table 18.1 to generate two samples from $f(x)$. [*Hint*: For convenience, use a rectangular $g(x)$ over the defined range of $f(x)$.]
4. The interarrival time of customers at HairKare is described by the following distribution:

$$f_1(t) = \frac{k_1}{t}, \; 12 \le t \le 20$$

   The time to get a haircut is represented by the following distribution:

$$f_2(t) = \frac{k_2}{t^2}, \; 18 \le t \le 22$$

   The constant k_1 and k_2 are determined such that $f_1(t)$ and $f_2(t)$ are probability density functions. Use the acceptance-rejection method (and the random numbers in Table 18.1) to determine when the first customer will leave HairKare and when the next customer will arrive. Assume that the first customer arrives at $T = 0$.

18.4 GENERATION OF RANDOM NUMBERS

Uniform (0, 1) random numbers play a key role in sampling from distributions. True (0, 1) random numbers can only be generated by electronic devices. However, because simulation models are executed on the computer, the use of electronic devices to generate random numbers is much too slow for that purpose. Additionally, electronic devices are activated by laws of chance, and hence it will be impossible to duplicate the same sequence of random numbers at will. This point is important because debugging, verification, and validation of the simulation model often require duplicating the same sequence of random numbers.

The only practical way for generating (0, 1) random numbers for use in simulation is based on arithmetic operations. Such numbers are not truly random because they can be generated in advance. It is thus more appropriate to refer to them as **pseudorandom numbers**.

The most common arithmetic operation for generating (0, 1) random numbers is the **multiplicative congruential method**. Given the parameters u_0, b, c, and m, a pseudorandom number R_n can be generated from the formulas:

$$u_n = (bu_{n-1} + c) \bmod (m),\ n = 1, 2, \ldots$$

$$R_n = \frac{u_n}{m},\ n = 1, 2, \ldots$$

The initial value u_0 is usually referred to as the **seed** of the generator.

Variations of the multiplicative congruential method that improve the quality of the generator can be found in Law and Kelton (2000).

Example 18.4-1

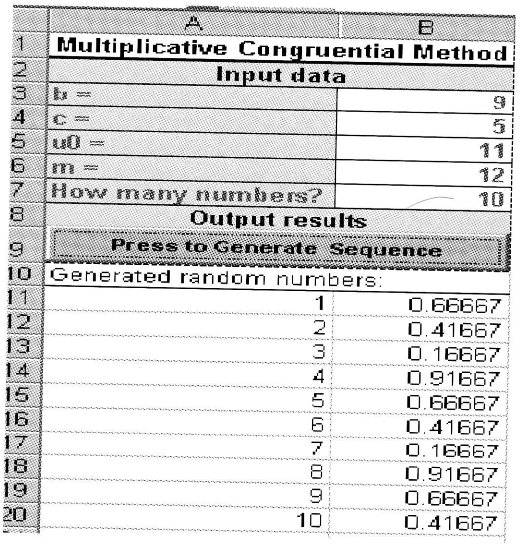
Generate three random numbers based on the multiplicative congruential method using the following initial values: $b = 9$, $c = 5$, $u_0 = 11$, and $m = 12$.

$$u_1 = (9 \times 11 + 5) \bmod 12 = 8,\ R_1 = \frac{8}{12} = .6667$$

$$u_2 = (9 \times 8 + 5) \bmod 12 = 5,\ R_2 = \frac{5}{12} = .4167$$

$$u_3 = (9 \times 5 + 5) \bmod 12 = 2,\ R_3 = \frac{2}{12} = .1667$$

For convenience, the Excel template ch18RandomNumberGenerator.xls is designed to carry out the multiplicative congruential calculations. Figure 18.7 provides the sequence associated with the parameters of this example. Observe carefully that the cycle length is exactly 4, after which the sequence repeats itself. The conclusion here is that the choice of u_0, b, c, and m is critical in determining the (statistical) quality of the generator and its cycle length. Thus, "casual" implementation of the congruential formula is not advisable. Instead, one must use a reliable and tested generator. Practically all commercial software are equipped with dependable random number generators.



	A	B
1	Multiplicative Congruential Method	
2	Input data	
3	b =	9
4	c =	5
5	u0 =	11
6	m =	12
7	How many numbers?	10
8	Output results	
9	Press to Generate Sequence	
10	Generated random numbers:	
11	1	0.66667
12	2	0.41667
13	3	0.16667
14	4	0.91667
15	5	0.66667
16	6	0.41667
17	7	0.16667
18	8	0.91667
19	9	0.66667
20	10	0.41667

FIGURE 18.7

Excel random numbers output for the data of Example 18.4-1

PROBLEM SET 18.4A

1. Use Excel template ch18RandomNumberGenerator.xls to observe the change in the cycle length of the generator. Use the following sets of parameters and compare the results with those in Example 18.4-1:

$$b = 17,\ c = 111,\ u_0 = 7,\ m = 103$$

2. Find a random number generator for your computer, and use it to generate 1000 zero-one random numbers. Histogram the resulting values (using Microsoft histogram tool, see Section 12.5) and visually convince yourself that the obtained numbers reasonably follow the (0, 1) uniform distribution. Actually, to test the sequence properly, you would need to apply the following tests: chi-square goodness of fit (see Section 12.6), runs test for independence, and correlation test (see Law and Kelton [2000] for details).

18.5 MECHANICS OF DISCRETE SIMULATION

This section details how typical statistics are collected in a simulation model. The vehicle of explanation is a single queue model. Section 18.5.1 uses a numeric example to detail the actions and computations that take place in a single-server queuing simulation model. Because of the tedious computations that typify the execution of a simulation model, Section 18.5.2 shows how the single-server model is modeled and executed using Excel spreadsheet.

18.5.1 Manual Simulation of a Single-Server Model

The interarrival time of customers at HairKare Barbershop is exponential with a mean of 15 minutes. The shop is operated by one barber, and it takes between 10 and 15 minutes, uniformly distributed, to complete a haircut. Customers are served on a first-in, first-out (FIFO) basis. The objective of the simulation is to compute the following measures of performance:

1. The average utilization of the shop.
2. The average number of waiting customers.
3. The average time a customer waits in queue.

The logic of the simulation model can be described in terms of the actions associated with its arrival and departure events.

Arrival Event.

1. Generate and store chronologically the arrival time of the next customer (= current simulation time + interarrival time).
2. If the facility (barber) is idle
   - **(a)** Start service and declare the facility busy. Update the facility utilization statistics.
   - **(b)** Generate and store chronologically the departure time of the customer (= current simulation time + service time).
3. If the facility is busy, place the customer in the queue and update the queue statistics.

Departure Event.

1. If the queue is empty, declare the facility idle. Update the facility utilization statistics.
2. If the queue is not empty
   - **(a)** Select a customer from the queue, and place it in the facility. Update the queue and facility utilization statistics.
   - **(b)** Generate and store chronologically the departure time of the customer (= current simulation time + service time).

From the data of the problem, the interarrival time is exponential with mean 15 minutes, and the service time is uniform between 10 and 15 minutes. Letting p and q represent random samples of interarrival and service times, then, as explained in Section 18.3.2, we get

$$p = -15\ln(R) \text{ minutes}, \ 0 \leq R \leq 1$$

$$q = 10 + 5R \text{ minutes}, \ 0 \leq R \leq 1$$

For the purpose of this example, we use R from Table 18.1, starting with column 1. We also use the symbol T to represent the simulation clock time. We further assume that the first customer arrives at $T = 0$ and that the facility starts empty.

Because the simulation computations are typically voluminous and tedious, the simulation is limited to the first 5 arrivals only. The example is designed to cover all possible situations that could arise in the course of the simulation. Later in the section we introduce the Excel template ch18SingleServerSimulator.xls, which allows you to experiment with the model without the need to carry out the computations manually.

Arrival of Customer 1 at $T = 0$. Generate the arrival of customer 2 at

$$T = 0 + p_1 = 0 + [-15\ln(.0589)] = 42.48 \text{ minutes}$$

Because the facility is idle at $T = 0$, customer 1 starts service immediately. The departure time is thus computed as

$$T = 0 + q_1 = 0 + (10 + 5 \times .6733) = 13.37 \text{ minutes}$$

The *chronological* list of future events is thus given as:

Time, T	Event
13.37	Departure of customer 1
42.48	Arrival of customer 2

Departure of Customer 1 at $T = 13.37$. Because the queue is empty, the facility is declared idle. At the same time, we record that the facility has been busy between $T = 0$ and $T = 13.37$ minutes. The updated list of future events becomes

Time, T	Event
42.48	Arrival of customer 2

Arrival of Customer 2 at $T = 42.48$. Customer 3 will arrive at

$$T = 42.48 + [-15\ln(.4799)] = 53.49 \text{ minutes}$$

Because the facility is idle, customer 2 starts service and the facility is declared busy. The departure time is

$$T = 42.48 + (10 + 5 \times .9486) = 57.22 \text{ minutes}$$

The list of future events is updated as

Time, T	Event
53.49	Arrival of customer 3
57.22	Departure of customer 2

Arrival of Customer 3 at $T = 53.49$. Customer 4 will arrive at

$$T = 53.49 + [-15\ln(.6139)] = 60.81 \text{ minutes}$$

Because the facility is currently busy (until $T = 57.22$), customer 3 is placed in queue at $T = 53.49$. The updated list of future events is

Time, T	Event
57.22	Departure of customer 2
60.81	Arrival of customer 4

Departure of Customer 2 at T = 57.22. Customer 3 is taken out of the queue to start service. The waiting time is

$$W_3 = 57.22 - 53.49 = 3.73 \text{ minutes}$$

The departure time is

$$T = 57.22 + (10 + 5 \times .5933) = 70.19 \text{ minutes}$$

The updated list of future events is

Time, T	Event
60.81	Arrival of customer 4
70.19	Departure of customer 3

Arrival of Customer 4 at T = 60.81. Customer 5 will arrive at

$$T = 60.81 + [-15\ln(.9341)] = 61.83 \text{ minutes}$$

Because the facility is busy until T = 70.19, customer 4 is placed in the queue. The updated list of future events is

Time, T	Event
61.83	Arrival of customer 5
70.19	Departure of customer 3

Arrival of Customer 5 at T = 61.83. The simulation is limited to 5 arrivals only, hence customer 6 arrival is not generated. The facility is still busy, hence the customer is placed in the queue at T = 61.83. The updated list of events is

Time, T	Event
70.19	Departure of customer 3

Departure of Customer 3 at T = 70.19. Customer 4 is taken out of the queue to start service. The waiting time is

$$W_4 = 70.19 - 60.81 = 9.38 \text{ minutes}$$

The departure time is

$$T = 70.19 + [10 + 5 \times .1782] = 81.08 \text{ minutes}$$

The updated list of future events is

Time, T	Event
81.08	Departure of customer 4

Departure of Customer 4 at $T = 81.08$. Customer 5 is taken out of the queue to start service. The waiting time is

$$W_5 = 81.08 - 61.83 = 19.25 \text{ minutes}$$

The departure time is

$$T = 81.08 + (10 + 5 \times .3473) = 92.82 \text{ minutes}$$

The updated list of future events is

Time, T	Event
92.82	Departure of customer 5

Departure of Customer 5 at $T = 92.82$. There are no more customers in the system (queue and facility), and the simulation ends.

Figure 18.8 summarizes the changes in the length of the queue and the utilization of the facility as a function of the simulation time.

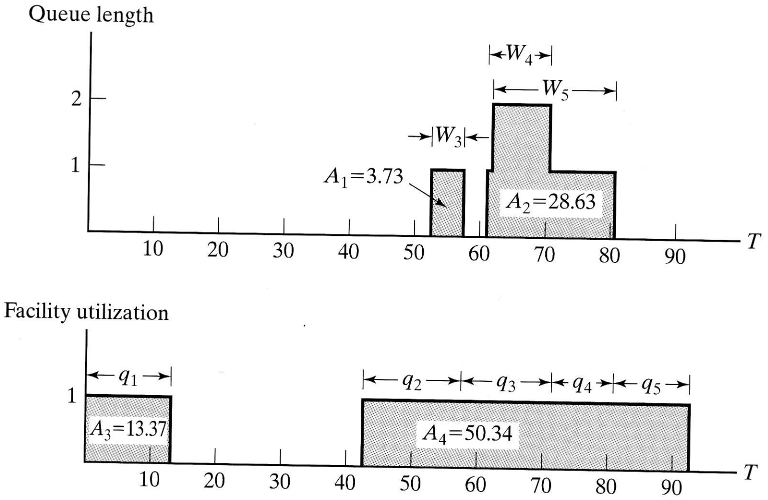


FIGURE 18.8

Changes in queue length and facility utilization as a function of simulation time, T

The queue length and the facility utilization are known as **time-based** variables because their variation is a function of time. As a result, their average values are computed as

$$\begin{pmatrix}\text{Average value of a}\\ \text{time-based variable}\end{pmatrix} = \frac{\text{Area under curve}}{\text{Simulated period}}$$

Implementing this formula for the data in Figure 18.8, we get

$$\begin{pmatrix}\text{Average queue}\\ \text{length}\end{pmatrix} = \frac{A_1 + A_2}{92.82} = \frac{32.36}{92.82} = .349 \text{ customer}$$

$$\begin{pmatrix}\text{Average facility}\\ \text{utilization}\end{pmatrix} = \frac{A_3 + A_4}{92.82} = \frac{63.71}{92.82} = .686 \text{ barber}$$

The average waiting time in the queue is an **observation-based** variable whose value is computed as

$$\begin{pmatrix}\text{Average value of an}\\ \text{observation-based variable}\end{pmatrix} = \frac{\text{Sum of observations}}{\text{Number of observations}}$$

Examination of Figure 18.8 reveals that the area under the queue-length curve actually equals the sum of the waiting time for the three customers who joined the queue; namely,

$$W_1 + W_2 + W_3 + W_4 + W_5 = 0 + 0 + 3.73 + 9.38 + 19.25 = 32.36 \text{ minutes}$$

The average waiting time in the queue for all customers is thus computed as

$$\overline{W}_q = \tfrac{32.36}{5} = 6.47 \text{ minutes}$$

PROBLEM SET 18.5A

1. Suppose that the barbershop of Section 18.5.1 is operated by two barbers, and customers are served on a FCFS basis. Suppose further that the time to get a haircut is uniformly distributed between 15 and 30 minutes. The interarrival time of customers is exponential, with a mean of 10 minutes. Simulate the system manually for 75 time units. From the results of the simulation, determine the average time a customer waits in queue, the average number of customers waiting, and the average utilization of the barbers. Use the random numbers in Table 18.1.
2. Classify the following variables as either *observation-based* or *time-based*:
   - **(a)** Time to failure of an electronic component.
   - **(b)** Inventory level of an item.
   - **(c)** Order quantity of an inventory item.
   - **(d)** Number of defective items in a lot.
   - **(e)** Time needed to grade test papers.
   - **(f)** Number of cars in the parking lot of a car-rental agency.
3. The following table represents the variation in the number of waiting customers in a queue as a function of the simulation time.

Simulation time, T (hr)	No. of waiting customers
$0 \le T \le 3$	0
$3 < T \le 4$	1
$4 < T \le 6$	2
$6 < T \le 7$	1
$7 < T \le 10$	0
$10 < T \le 12$	2
$12 < T \le 18$	3
$18 < T \le 20$	2
$20 < T \le 25$	1

Compute the following measures of performance:

(a) The average length of the queue.

(b) The average waiting time in the queue for those who must wait.

4. Suppose that the barbershop of Example 18.5-1 is operated by three barbers. Assume further that the utilization of the servers (barbers) is summarized as given in the following table:

Simulation time, T (hr)	No. of busy servers
$0 < T \le 10$	0
$10 < T \le 20$	1
$20 < T \le 30$	2
$30 < T \le 35$	1
$35 < T \le 40$	0
$40 < T \le 60$	1
$60 < T \le 70$	2
$70 < T \le 75$	3
$75 < T \le 80$	2
$80 < T \le 90$	1
$90 < T \le 100$	0

Determine the following measures of performance:

(a) The average utilization of the facility.

(b) The average idle time of the facility.

18.5.2 Spreadsheet-Based Simulation of the Single-Server Model

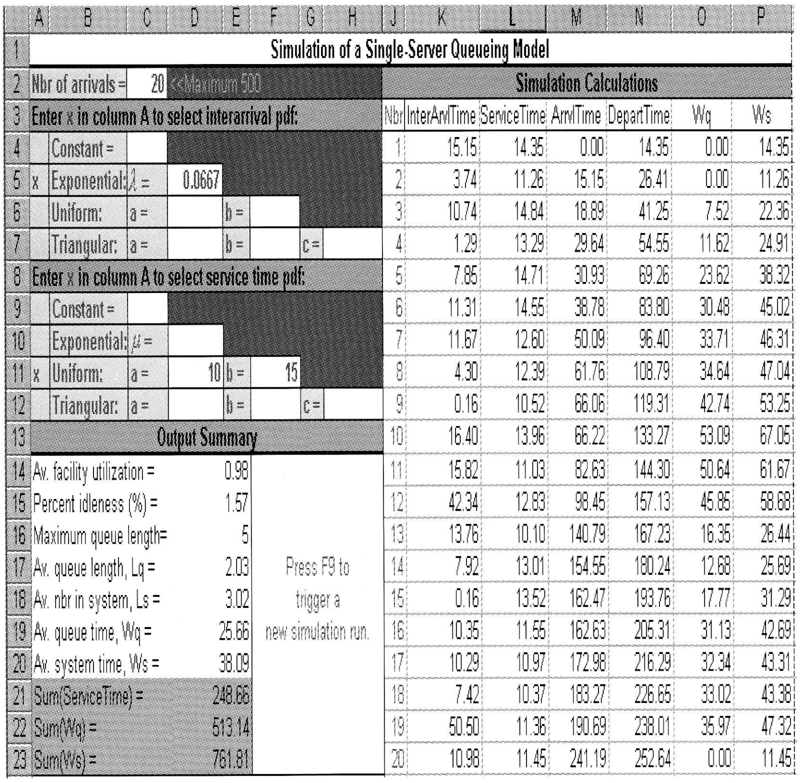
The presentation in Section 18.5.1 shows that simulation computations are typically tedious and voluminous. Thus, the use of the computer to execute simulation models is a must. This section develops a spreadsheet-based model for the single-server model. The objective of the development is to reinforce the ideas introduced in Section 18.5.1. Of course, a single-server model is a simple situation, and for this reason it can be modeled readily in a spreadsheet environment. Other situations require more intensive modeling effort, which is facilitated by available simulation packages (see Section 18.7).

The presentation in Section 18.5.1 shows that the simulation model of the single-server facility requires two basic elements:

1. A chronological list of the model's events.
2. A graph that keeps track of the changes in facility utilization and queue length.

These two elements remain essential in the development of the spreadsheet-based (indeed, any computer-based) simulation model. The difference is that the implementation is effected in a manner that is compatible with the use of the computer. As in Section 18.5.1, customers are served in order of arrival (FIFO).

Figure 18.9 provides the Excel template ch18SingleServerSimulator.xls. The input data allow representing the interarrival and service time in one of four ways: constant, exponential, uniform, and triangular. The triangular distribution is useful in that



Simulation of a Single-Server Queueing Model

	A	B	C	D	E	F	G–H
2	Nbr of arrivals =		20	<<Maximum 500			
3	Enter x in column A to select interarrival pdf:						
4		Constant =					
5	x	Exponential: λ =		0.0667			
6		Uniform:	a =		b =		
7		Triangular:	a =		b =		c =
8	Enter x in column A to select service time pdf:						
9		Constant =					
10		Exponential: μ =					
11	x	Uniform:	a =	10	b =	15	
12		Triangular:	a =		b =		c =
13	Output Summary						
14	Av. facility utilization =			0.98			
15	Percent idleness (%) =			1.57			
16	Maximum queue length=			5			
17	Av. queue length, Lq =			2.03	Press F9 to		
18	Av. nbr in system, Ls =			3.02	trigger a		
19	Av. queue time, Wq =			25.66	new simulation run.		
20	Av. system time, Ws =			38.09			
21	Sum(ServiceTime) =			248.66			
22	Sum(Wq) =			513.14			
23	Sum(Ws) =			761.81			

Simulation Calculations

Nbr	InterArvlTime	ServiceTime	ArvlTime	DepartTime	Wq	Ws
1	15.15	14.35	0.00	14.35	0.00	14.35
2	3.74	11.26	15.15	26.41	0.00	11.26
3	10.74	14.84	18.89	41.25	7.52	22.36
4	1.29	13.29	29.64	54.55	11.62	24.91
5	7.85	14.71	30.93	69.26	23.62	38.32
6	11.31	14.55	38.78	83.80	30.48	45.02
7	11.67	12.60	50.09	96.40	33.71	46.31
8	4.30	12.39	61.76	108.79	34.64	47.04
9	0.16	10.52	66.06	119.31	42.74	53.25
10	16.40	13.96	66.22	133.27	53.09	67.05
11	15.82	11.03	82.63	144.30	50.64	61.67
12	42.34	12.83	98.45	157.13	45.85	58.68
13	13.76	10.10	140.79	167.23	16.35	26.44
14	7.92	13.01	154.55	180.24	12.68	25.69
15	0.16	13.52	162.47	193.76	17.77	31.29
16	10.35	11.55	162.63	205.31	31.13	42.69
17	10.29	10.97	172.98	216.29	32.34	43.31
18	7.42	10.37	183.27	226.65	33.02	43.38
19	50.50	11.36	190.69	238.01	35.97	47.32
20	10.98	11.45	241.19	252.64	0.00	11.45

FIGURE 18.9

Excel output of a single server simulation model

it can be used as a rough initial estimate of any distribution, simply by providing three estimates, *a, b,* and *c*, that represent the smallest, most likely, and the largest values of the interarrival or service time. The only other information needed to drive the simulation is the length of the simulation run, which, in this model, is specified by the number of arrivals that can be generated in the model.

The spreadsheet calculations reserve one row for each arrival. The interarrival and service times for each arrival are generated from the input data. The first arrival is assumed to occur at $T = 0$. Because the facility starts idle, customer starts service immediately. Thus,

$$\begin{pmatrix}\text{Departure time}\\ \text{of customer 1}\end{pmatrix} = \begin{pmatrix}\text{Arrival time}\\ \text{of customer 1}\end{pmatrix} + \begin{pmatrix}\text{Service time}\\ \text{of customer 1}\end{pmatrix}$$

$$= 0 + 14.35 = 14.35$$

$$\begin{pmatrix}\text{Arrival time}\\ \text{of customer 2}\end{pmatrix} = \begin{pmatrix}\text{Arrival time}\\ \text{of customer 1}\end{pmatrix} + \begin{pmatrix}\text{Interarrival time}\\ \text{of customer 1}\end{pmatrix}$$

$$= 0 + 15.15 = 15.15$$

To determine the departure time of any customer *i*, we use the following formula

$$\begin{pmatrix}\text{Departure time}\\ \text{of customer } i\end{pmatrix} = \max\left\{\begin{pmatrix}\text{Arrival time}\\ \text{of customer } i\end{pmatrix}, \begin{pmatrix}\text{Departure time}\\ \text{of customer } i-1\end{pmatrix}\right\} + \begin{pmatrix}\text{Service time}\\ \text{of customer } i\end{pmatrix}$$

The formula says that a customer cannot start service until the facility becomes available. To illustrate the use of this formula in Figure 18.9, we have

$$\text{Departure time of customer 3} = \max\{18.89, 26.41\} + 14.86 = 41.25$$

We now turn our attention to collecting the statistics of the model. First, note that for customer *i*, the waiting time in queue, $W_q(i)$, and in the entire system, $W_s(i)$, are computed as

$$W_q(i) = \begin{pmatrix}\text{Departure time}\\ \text{of customer } i\end{pmatrix} - \begin{pmatrix}\text{Arrival time}\\ \text{of customer } i\end{pmatrix} - \begin{pmatrix}\text{Service time}\\ \text{of customer } i\end{pmatrix}$$

$$W_s(i) = \begin{pmatrix}\text{Departure time}\\ \text{of customer } i\end{pmatrix} - \begin{pmatrix}\text{Arrival time}\\ \text{of customer } i\end{pmatrix}$$

Next, it may appear that computing the remaining statistics of the model necessitate tracking the changes in facility utilization and in queue length (as we did in Section 18.5.1). Fortunately, the calculations are simplified by two observations we made in Section 18.5.1 and explained in Figure 18.8:

1. Area under facility utilization curve = Sum of service times of all arrivals
2. Area under queue length curve = Sum of waiting times of all arrivals

To explain this point, Excel output in Figure 18.9 computes three sums:

$$\text{Sum of service times} = 248.66$$

$$\text{Sum of } W_q = 513.14$$

$$\begin{aligned}\text{Sum of } W_s &= \text{Sum of } W_q + \text{Sum of service times}\\ &= 761.81\ (\simeq 248.66 + 513.14)\end{aligned}$$

Given that the last arrival (customer 20) departs at $T = 252.64$, it follows that

$$\begin{pmatrix}\text{Average facility}\\ \text{utilization}\end{pmatrix} = \frac{248.66}{252.64} = .9842$$

$$\begin{pmatrix}\text{Average queue}\\ \text{length}\end{pmatrix} = \frac{513.14}{252.64} = 2.03$$

Percent idleness of the facility is computed as $(1 - .9842) \times 100 = 1.575\%$.

The remainder of the statistics are calculated in a straightforward manner; namely,

$$\begin{pmatrix}\text{Average waiting}\\ \text{time in queue}\end{pmatrix} = \frac{\text{Sum of } W_q}{\text{Number of arrivals}} = \frac{513.14}{20} = 25.66$$

$$\begin{pmatrix}\text{Average system}\\ \text{time}\end{pmatrix} = \frac{\text{Sum of } W_s}{\text{Number of arrivals}} = \frac{761.81}{20} = 38.09$$

The Excel template is designed for a maximum of 500 arrivals. Also, you can obtain different simulation samples by pressing F9 or by changing any of the input data cells.

Another spreadsheet was developed for simulating multiserver models (ch18 MultiServerSimulator.xls). The design of the template is based on the same ideas used in the single-server case. However, the determination of the departure time is not as straightforward and, hence, requires the use of VBA macros.

PROBLEM SET 18.5B

1. Using the input data in Section 18.5.1, run the Excel simulator for 10 arrivals and graph the changes in facility utilization and queue length as a function of the simulation time. Verify that the areas under the respective curves equal the sum of the service times and the sum of the waiting times, respectively.
2. Simulate the $M/M/1$ model for 500 arrivals given the arrival rate $\lambda = 4$ customers per hour and the service rate $\mu = 6$ departures per hour. Run 5 replications (by refreshing the spreadsheet—pressing F9) and determine a 95% confidence interval for all the measures of performance of the model. Compare the results with the steady-state theoretical values of the $M/M/1$ model.
3. Television units arrive on a conveyor belt every 11.5 minutes for inspection at a single-operator station. Detailed data for the inspection station are not available. However, the operator estimates that it takes 9.5 minutes "on the average" to inspect a unit. Under worst conditions, the inspection time does not exceed 15 minutes, and in certain units inspection time may be as low as 9 minutes.
   (a) Use the Excel simulator to simulate the inspection of 200 TV units.
   (b) Based on 5 replications, estimate the average number of units awaiting inspection and the average utilization of the inspection station.

18.6 METHODS FOR GATHERING STATISTICAL OBSERVATIONS

Simulation is a statistical experiment, and its output must be interpreted using proper statistical inference tools (e.g., confidence intervals and hypothesis testing). To accomplish this task, the observations of the simulation experiment must satisfy three conditions:

1. Observations are drawn from stationary (identical) distributions.
2. Observations are sampled from a normal population.
3. Observations are independent.

It so happens that, in the strict sense, the simulation experiment does not satisfy any of these conditions. Nevertheless, we can render these conditions statistically viable by restricting the manner in which the simulation observations are gathered.

First, we consider the question of stationarity. Simulation output is a function of the length of the simulated period. The initial period produces erratic behavior and is usually referred to as the **transient** or **warm-up period**. When the output stabilizes, the system operates under **steady state**. Unfortunately, there is no way to predict the start point of steady state in advance. In general, a longer simulation run has better chance of reaching steady state. This point is demonstrated in Example 18.1-1 where the accuracy of estimating the area of a circle by Monte Carlo increases with the sample size. Thus, nonstationarity can be accounted for by using a sufficiently large sample size.

Next, we consider the requirement that simulation observations must be drawn from a normal population. This requirement is realized by using the *Central Limit Theorem* (see Section 12.4.4), which states that the distribution of the average of a sample is asymptotically normal regardless of the parent population from which the sample is drawn. The Central Limit Theorem is thus the main tool we use for satisfying the normal distribution assumption.

The third condition deals with the independence of the observations. The nature of the simulation experiment does not guarantee independence among successive simulation observations. However, by using sample average to represent a simulation observation, we can alleviate the problem of lack of independence. This is particularly true when we increase the time base used to compute the sample average.

Having discussed the peculiarities of the simulation experiment and ways to circumvent them, we present the three most common methods for collecting observations in simulation:

1. Subinterval method
2. Replication method
3. Regenerative (or cycles) method

18.6.1 Subinterval Method

Figure 18.10 illustrates the idea of the subinterval method. Suppose that the simulation is executed for T time units (i.e., run length $= T$) and that it is desired to collect n observations. The subinterval method first truncates the initial transient period and then subdivides the remainder of the simulation run into n equal subintervals (or batches). The average of the desired measure of performance (e.g., queue length or waiting time in queue) within each subinterval is then used to represent a single observation. The truncation of the initial transient period implies that no statistical data are collected during the period.

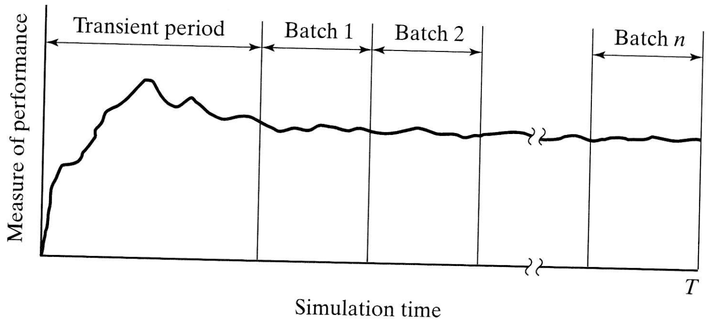


FIGURE 18.10
Collecting simulation data using the subinterval method

The advantage of the subinterval method is that the effect of the transient (nonstationary) conditions is mitigated, particularly for those observations that are collected toward the end of the simulation run. The disadvantage of the method is that successive batches with common boundary conditions are necessarily correlated. The effect of correlation can be alleviated by increasing the time base for each batch.

Example 18.6-1

Figure 18.11 shows the change in queue length in a single queue model as a function of the simulation time. The simulation run length is $T = 35$ hours, and the length of the transient period is estimated to equal 5 hours. It is desired to collect 5 observations—that is, $n = 5$. The corresponding time base for each batch thus equals $\frac{(35-5)}{5} = 6$ hours.

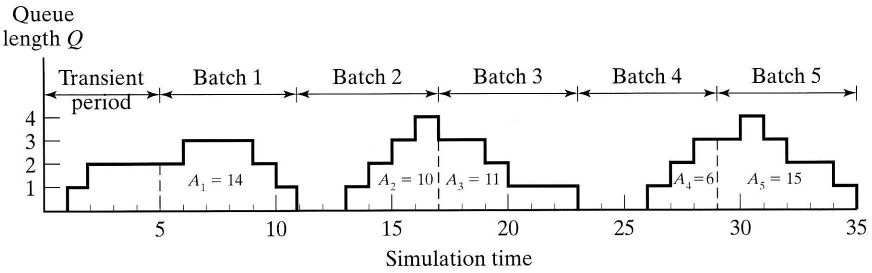


FIGURE 18.11

Change in queue length with simulation time in Example 18.6-1

Let $\overline{Q}_i$ represent the average queue length in batch i. Because the queue length is a time-based variable, we have

$$\overline{Q}_i = \frac{A_i}{t}, \; i = 1, 2, \ldots, 5$$

where A_i is the area under the queue length curve associated with batch (observation) i, and t is the time base per batch. In the present example, $t = 6$ hours.

The data in Figure 18.11 produce the following observations:

Observation i	1	2	3	4	5
A_i	14	10	11	6	15
$\overline{Q}_i$	2.33	1.67	1.83	1.00	2.5
Sample mean = 1.87		Sample standard deviation = .59			

The sample mean and variance can be used to compute a confidence interval, if desired. The computation of the sample variance in Example 18.6-1 is based on the following familiar formula:

$$s = \sqrt{\frac{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2}{n - 1}}$$

This formula is only an approximation of the true variance because it ignores the effect of autocorrelation between the successive batches. The exact formula can be found in Law and Kelton (2000, pp. 249–253).

18.6.2 Replication Method

In the replication method, each observation is represented by an independent simulation run in which the transient period is truncated, as illustrated in Figure 18.12. The computation of the observation averages for each batch is the same as in the subinterval method. The only difference is that the standard variance formula is applicable because the batches are not correlated.

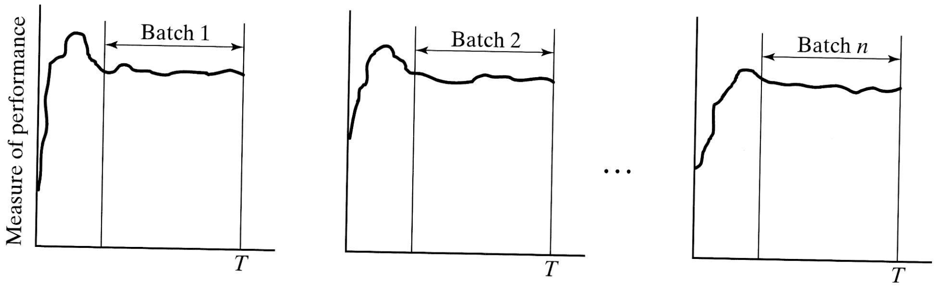


FIGURE 18.12

Collecting simulation data using the replication method

The advantage of the replication method is that each simulation run is driven by a distinct (0, 1) random number stream, which yields observations that are truly statistically independent. The disadvantage is that each observation may be biased by the initial effect of the transient conditions. Such a problem may be alleviated by making the run length sufficiently large.

18.6.3 Regenerative (Cycle) Method

The regenerative method may be regarded as an extended case of the subinterval method. The motivation behind the new method is that it attempts to reduce the effect of autocorrelation that characterizes the subinterval method by requiring similar starting conditions for each batch. For example, if the variable we are dealing with is the queue length, each batch would start at an instant where the queue length is zero. Unlike the subinterval method, the nature of the regenerative method may result in unequal time bases for the different batches.

Although the regenerative method may reduce autocorrelation, it has the disadvantage of yielding a smaller number of batches for a given run length. This follows because we cannot predict when a new batch will start or how long a batch will last. Under steady-state conditions, however, we should expect the starting points for the successive batches to be more or less evenly spaced.

The computation of the average for batch i in the regenerative method is generally defined as the ratio of two random variables a_i and b_i—that is, $x_i = \frac{a_i}{b_i}$. The definitions of a_i and b_i depend on the variable being computed. Specifically, if the variable is *time based*, then a_i would represent the area under the curve and b_i would equal the associated time base. If the variable is *observation based*, then a_i would be the total sum of the observations within batch i and b_i would be the associated number of observations.

Because x_i is the ratio of two random variables, an unbiased estimate of the sample average can be shown to be

$$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$$

where

$$y_i = \frac{n\bar{a}}{\bar{b}} - \frac{(n-1)(n\bar{a} - a_i)}{n\bar{b} - b_i},\ i = 1, 2, \ldots, n$$

$$\bar{a} = \frac{\sum_{i=1}^{n} a_i}{n}$$

$$\bar{b} = \frac{\sum_{i=1}^{n} b_i}{n}$$

In this case, a confidence interval is based on the mean and standard deviation of y_i.

Example 18.6-2

Figure 18.13 represents the number of busy servers in a single facility with three parallel servers. The length of the simulation run is 35 time units, and the length of the tran-

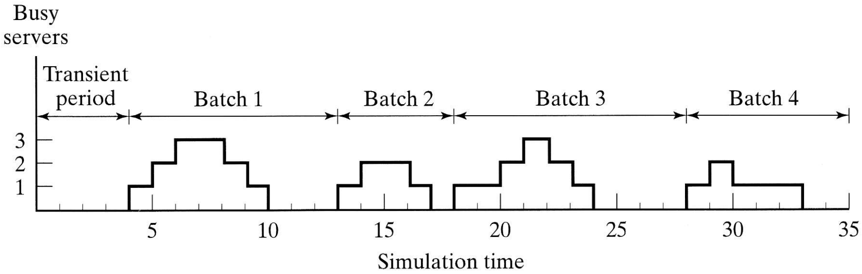


FIGURE 18.13

Changes in the number of busy servers as a function of time in Example 18.6-2 sient period is 4 time units. It is desired to estimate the average utilization of the facility based on the regenerative method.

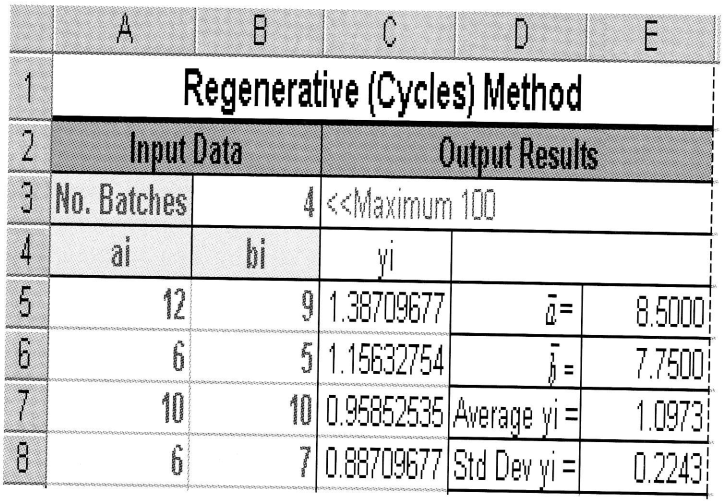
After truncating the transient period, Figure 18.13 yields four batches with the common characteristic of starting with all three servers idle. The associated values of a_i and b_i are given in the following table:

Batch i	a_i	b_i
1	12	9
2	6	5
3	10	10
4	6	7
Averages	$\bar{a} = 8.50$	$\bar{b} = 7.75$

Based on these data, we have

$$y_i = \frac{4 \times 8.5}{7.75} - \frac{(4 - 1) \times (4 \times 8.5 - a_i)}{4 \times 7.75 - b_i} = 4.39 - \frac{102 - 3a_i}{31 - b_i}$$

These computations can be automated readily using Excel template ch18 Regenerative.xls. Figure 18.14 provides the associated output.



	A	B	C	D	E
1	Regenerative (Cycles) Method				
2	Input Data		Output Results		
3	No. Batches	4	<<Maximum 100		
4	ai	bi	yi		
5	12	9	1.38709677	$\bar{a}$=	8.5000
6	6	5	1.15632754	$\bar{b}$=	7.7500
7	10	10	0.95852535	Average yi =	1.0973
8	6	7	0.88709677	Std Dev yi =	0.2243

FIGURE 18.14

Excel calculations of Example 18.6-2 simulation observations based on the regenerative method

PROBLEM SET 18.6A

1. In Example 18.6-1, use the subinterval method to compute the average waiting time in the queue for those who must wait.
2. In a simulation model, the subinterval method is used to compute batch averages. The transient period is estimated to be 100, and each batch has a time base of 100 time units as well. Using the following data that provide the waiting times for customers as a function of the simulation time, estimate the 95% confidence interval for the mean waiting time.

Time interval	Waiting times
0–100	10, 20, 13, 14, 8, 15, 6, 8
100–200	12, 30, 10, 14, 16
200–300	15, 17, 20, 22
300–400	10, 20, 30, 15, 25, 31
400–500	15, 17, 20, 14, 13
500–600	25, 30, 15

3. In Example 18.6-2, suppose that the start point for each observation is the point in time where all the servers have just become idle. Thus, in Figure 18.13, these points correspond to $t = 10, 17, 24$, and 33. Compute the 95% confidence interval for the utilization of the servers based on the new definition of the regenerative points.
4. In a single-server queuing situation, the system is simulated for 100 hours. The results of the simulation show that the server was busy only during the following time intervals: (0, 10), (15, 20), (25, 30), (35, 60), (70, 80), and (90, 95). The length of the transient period is estimated to be 10 hours.
   (a) Define the observation start point needed to implement the regenerative method.
   (b) Compute the 95% confidence interval for the average utilization of the server based on the regenerative method.
   (c) Apply the subinterval method to the same problem using a sample size $n = 5$. Compute the corresponding 95% confidence interval, and compare it with the one obtained from the regenerative method.

18.7 SIMULATION LANGUAGES

Execution of simulation models entails two distinct types of computations: (1) file manipulations that deal with the chronological storage and processing of model events, and (2) arithmetic and bookkeeping computations associated with generation of random samples and collection of model statistics. The first type of computation involves extensive logic development in list processing, and the second type entails tedious and time-consuming calculations. The nature of these computations makes the computer an essential tool for executing simulation models and, in turn, prompts the development of special computer simulation languages for performing these computations conveniently and efficiently.

Available discrete simulation languages fall into two broad categories:

1. Event scheduling
2. Process oriented

In event scheduling languages, the user details the actions associated with the occurrence of each event, in much the same way they are given in Example 18.5-1. The main role of the language in this case is (1) automation of sampling from distributions, (2) storage and retrieval of events in chronological order, and (3) collection of model statistics.

Process-oriented languages use blocks or nodes that can be linked together to form a network that describes the movements of **transactions** or **entities** (i.e., customers) in the system. For example, the three most prominent blocks/nodes in any process simulation language are a *source* from which transactions are created, a *queue* where they can wait if necessary, and a *facility* where service is performed. Each of these blocks/nodes is defined with all the information needed to drive the simulation automatically. For example, once the interarrival time for the source is specified, a process-oriented language automatically "knows" when arrival events will occur. In effect, each block/node of the model has standing instructions that define *how* and *when* transactions are moved in the simulation network.

Process-oriented languages are internally driven by the same actions used in event-scheduling languages. The difference is that these actions are automated to relieve the user of the tedious computational and logical details. In a way, we can regard process-oriented languages as being based on the input-output concept of the "black box" approach. This essentially means that process-oriented languages trade modeling flexibility for simplicity and ease of use.

Prominent event-scheduling languages include SIMSCRIPT, SLAM, and SIMAN. Over the years, these languages have evolved to include process-oriented capabilities. All three languages allow the user to write (a portion of) the model in higher-level language, such as FORTRAN or C. This capability is necessary to allow the user to model complex logic that otherwise cannot be achieved directly by the regular facilities of these languages. A major reason for this limitation is the restrictive and perhaps convoluted manner in which these languages move transactions (or entities) among the model's queues and facilities.

The oldest process-oriented language is GPSS. This language, which was first developed in the early 1960s, has since evolved to accommodate new modeling needs of complex systems. To use this language effectively, the user must master the "inner works" of some 80 different blocks. Despite its long history, GPSS still possesses some modeling peculiarities that are difficult to justify. An example of these peculiarities is the need to approximate continuous distributions by piecewise linear ones. It is true that some recent versions of the language have provided direct capabilities for some of the continuous distributions (e.g., exponential and normal). However, with the present tremendous capability of the computer, it is difficult to understand why such an obstacle has persisted for so long.

Several modern commercial packages currently dominate the simulation market, including Arena, AweSim, and GPSS/H, to mention only a few. These packages use extensive user interface to simplify the process of creating a simulation model. They also provide animation capabilities where changes in the system can be observed visually. However, to the experienced user, these interfaces may appear to reduce the development of a simulation model to a "slow motion" pace. It is not surprising that some users prefer to write simulation models in such general programming languages as C, Basic, and FORTRAN.

PROBLEM SET 18.8A[2]

1. Patrons arrive randomly at a three-clerk post office. The interarrival time is exponential with a mean of 5 minutes. The time a clerk spends with a patron is exponential with a mean of 10 minutes. All arriving patrons form one queue and wait for the first available

[2] Work these problems using a simulation language of your choice, or use BASIC, FORTRAN, or C.

free clerk. Run a simulation model of the system for 480 minutes to determine the following:

(a) The average number of patrons waiting in the queue.

(b) The average utilization of the clerks.

(c) Compare the simulation results with those of the M/M/c queuing model (Chapter 17) and with the spreadsheet ch18MultiServerSimulator.xls.

2. Television units arrive for inspection on a conveyor belt at the constant rate of 5 units per hour. The inspection time takes between 10 and 15 minutes, uniformly distributed. Past experience shows that 20% of inspected units must be adjusted and then sent back for reinspection. The adjustment time is also uniformly distributed between 6 and 8 minutes. Run a simulation model for 480 minutes to compute the following:

(a) The average time a unit takes until it passes inspection.

(b) The average number of times a unit must be reinspected before it exits the system.

3. A mouse is trapped in a maze and desperately "wants out." After trying between 1 and 3 minutes, uniformly distributed, there is a 30% chance that it will find the right path. Otherwise, it will wander around aimlessly for between 2 and 3 minutes, uniformly distributed, and eventually end up where it started, only to try once again. The mouse can "try freedom" as many times as it pleases, but there is a limit to everything. With so much energy expended in trying and retrying, the mouse is certain to expire if it does not make it within a period that is normally distributed, with a mean of 10 minutes and a standard deviation of 2 minutes. Write a simulation model to estimate the probability that the mouse will be free. For the purpose of estimating the probability, assume that 100 mice (replications) will be processed by the model.

4. In the final stage of automobile manufacturing, a car moving on a transporter is situated between two parallel workstations to allow work to be done on both the left and right sides of the car simultaneously. The operation times for the left and right sides are uniform between 15 and 20 minutes and 18 and 22 minutes, respectively. The transporter arrives at the stations area every 20 minutes. Simulate the process for 480 minutes to determine the utilization of the left and right stations.

5. Cars arrive at a one-bay car wash facility where the interarrival time is exponential, with a mean of 10 minutes. Arriving cars line up in a single lane that can accommodate at most five waiting cars. If the lane is full, newly arriving cars will go elsewhere. It takes between 10 and 15 minutes, uniformly distributed, to wash a car. Simulate the system for 960 minutes, and estimate the time a car spends in the facility.

SELECTED REFERENCES

Box, G., and M. Muller, "A Note on the Generation of Random Normal Deviates," *Annals of Mathematical Statistics*, Vol. 29, pp. 610–611, 1958.

Law, A., and W. Kelton, *Simulation Modelling & Analysis*, 3rd ed., McGraw-Hill, New York, 2000.

Ross, S., *A Course in Simulation*, Macmillan, New York, 1990.

Rubenstein, R., B. Melamed, and A. Shapiro, *Modern Simulation and Modeling,* Wiley, New York, 1998.

Taha, H., *Simulation Modeling and SIMNET*, Prentice Hall, Upper Saddle River, N.J., 1988.

CHAPTER 19

Markovian Decision Process

This chapter applies dynamic programming to the solution of a stochastic decision process with a finite number of states. The transition probabilities between the states are described by a Markov chain.[1] The reward structure of the process is described by a matrix that represents the revenue (or cost) associated with movement from one state to another. Both the transition and revenue matrices depend on the decision alternatives available to the decision maker. The objective of the problem is to determine the optimal policy that maximizes the expected revenue over a finite or infinite number of stages.

19.1 SCOPE OF THE MARKOVIAN DECISION PROBLEM: THE GARDENER PROBLEM

We use an example to present the details of the Markovian decision process. The example paraphrases several important applications in the areas of inventory, replacement, cash flow management, and regulation of water reservoir capacity.

Every year, at the beginning of the gardening season (March through September), a gardener uses a chemical test to check the soil condition. Depending on the outcomes of the tests, productivity for the new season falls into one of three states: (1) good, (2) fair, and (3) poor.

Over the years, the gardener observed that prevailing weather conditions during winter (October through February) play an important role in affecting the soil condition, leaving it the same or making it worse, but never improving it. In this regard, last year's soil condition is an important factor in current year's productivity. Using the gardener's test data, the transition probabilities over a 1-year period from one productivity state to another is represented by the following Markov chain:

[1] A review of Markov chains is given in Section 19.5.

$$\mathbf{P}^1 = \begin{array}{c} \\ \text{State of} \\ \text{the system} \\ \text{this year} \end{array} \begin{array}{c} \\ \left\{\begin{array}{c} 1 \\ 2 \\ 3 \end{array}\right. \end{array} \begin{array}{c} \overbrace{\text{State of the system next year}} \\ \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} .2 & .5 & .3 \\ 0 & .5 & .5 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

The transition probabilities in $\mathbf{P}^1$ indicate that the productivity for a current year can be no better than last year's. For example, if this year's soil condition is fair (state 2), next year's productivity may remain fair with probability .5 or become poor (state 3), also with probability .5.

The gardener can alter the transition probabilities $\mathbf{P}^1$ through other courses of action. Typically, fertilizer is applied to boost the soil condition, which yields the following transition matrix:

$$\mathbf{P}^2 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & .30 & .60 & .10 \\ 2 & .10 & .60 & .30 \\ 3 & .05 & .40 & .55 \end{array}$$

To put the decision problem into perspective, the gardener associates a return function (or a reward structure) with the transition from one state to another. The return function expresses the gain or loss during a 1-year period, depending on the states between which the transition is made. Because the gardener has the option of using or not using fertilizer, gain and loss vary depending on the decision made. The matrices $\mathbf{R}^1$ and $\mathbf{R}^2$ summarize the return functions in hundreds of dollars associated with the matrices $\mathbf{P}^1$ and $\mathbf{P}^2$, respectively.

$$\mathbf{R}^1 = \|r_{ij}^1\| = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 7 & 6 & 3 \\ 2 & 0 & 5 & 1 \\ 3 & 0 & 0 & -1 \end{array}$$

$$\mathbf{R}^2 = \|r_{ij}^2\| = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 6 & 5 & -1 \\ 2 & 7 & 4 & 0 \\ 3 & 6 & 3 & -2 \end{array}$$

The elements r_{ij}^2 of $\mathbf{R}^2$ consider the cost of applying fertilizer. For example, if the soil condition was fair last year (state 2) and becomes poor this year (state 3), its gain will be $r_{23}^2 = 0$ compared with $r_{23}^1 = 1$ when no fertilizer is used. In this regard, **R** gives the net reward after the cost of the fertilizer is factored in.

What kind of a decision problem does the gardener have? First, we must know whether the gardening activity will continue for a limited number of years or indefinitely. These situations are referred to as **finite-stage** and **infinite-stage** decision problems. In both cases, the gardener uses the outcome of the chemical tests (state of the system) to determine the *best* course of action (fertilize or do not fertilize) that maximizes expected revenue.

The gardener may also be interested in evaluating the expected revenue resulting from a prespecified course of action for a given state of the system. For example, fertilizer may be applied whenever the soil condition is poor (state 3). The decision-making process in this case is said to be represented by a **stationary policy**.

Each stationary policy will be associated with a different transition and return matrices, which are constructed from the matrices $\mathbf{P}^1$, $\mathbf{P}^2$, $\mathbf{R}^1$, and $\mathbf{R}^2$. For example, for the stationary policy calling for applying fertilizer only when the soil condition is poor (state 3), the resulting transition and return matrices are given as

$$\mathbf{P} = \begin{pmatrix} .20 & .50 & .30 \\ .00 & .50 & .50 \\ .05 & .40 & .55 \end{pmatrix}, \mathbf{R} = \begin{pmatrix} 7 & 6 & 3 \\ 0 & 5 & 1 \\ 6 & 3 & -2 \end{pmatrix}$$

These matrices differ from $\mathbf{P}^1$ and $\mathbf{R}^1$ in the third rows only, which are taken directly from $\mathbf{P}^2$ and $\mathbf{R}^2$, the matrices associated with applying fertilizer.

PROBLEM SET 19.1A

1. In the gardener model, identify the matrices **P** and **R** associated with the stationary policy that calls for using fertilizer whenever the soil condition is fair or poor.
2. Identify all the stationary policies for the gardener model.

19.2 FINITE-STAGE DYNAMIC PROGRAMMING MODEL

Suppose that the gardener plans to "retire" from gardening in N years. We are interested in determining the optimal course of action for each year (to fertilize or not to fertilize) that will return the highest expected revenue at the end of N years.

Let $k = 1$ and 2 represent the two courses of action (alternatives) available to the gardener. The matrices $\mathbf{P}^k$ and $\mathbf{R}^k$ representing the transition probabilities and reward function for alternative k were given in Section 19.1 and are summarized here for convenience.

$$\mathbf{P}^1 = \|r_{ij}^1\| = \begin{pmatrix} .2 & .5 & .3 \\ 0 & .5 & .5 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{R}^1 = \|r_{ij}^1\| = \begin{pmatrix} 7 & 6 & 3 \\ 0 & 5 & 1 \\ 0 & 0 & -1 \end{pmatrix}$$

$$\mathbf{P}^2 = \|p_{ij}^2\| = \begin{pmatrix} .30 & .60 & .10 \\ .10 & .60 & .30 \\ .05 & .40 & .55 \end{pmatrix}, \mathbf{R}^2 = \|r_{ij}^2\| = \begin{pmatrix} 6 & 5 & -1 \\ 7 & 4 & 0 \\ 6 & 3 & -2 \end{pmatrix}$$

The gardener problem is expressed as a finite-stage dynamic programming (DP) model as follows. For the sake of generalization, define

m = Number of states at each stage (year) (= 3 in the gardener problem)

$f_n(i)$ = Optimal *expected* revenue of stages $n, n + 1, \ldots, N$, given i is the state of the system (soil condition) at the beginning of year n

The *backward* recursive equation relating f_n and f_{n+1} is

$$f_n(i) = \max_k \left\{ \sum_{j=1}^{m} p_{ij}^k [r_{ij}^k + f_{n+1}(j)] \right\}, \ n = 1, 2, \ldots, N$$

where $f_{N+1}(j) = 0$ for all j.

A justification for the equation is that the cumulative revenue, $r_{ij}^k + f_{n+1}(j)$, resulting from reaching state j at stage $n + 1$ from state i at stage n occurs with probability p_{ij}^k. Let

$$v_i^k = \sum_{j=1}^{m} p_{ij}^k r_{ij}^k$$

The DP recursive equation can be written as

$$f_N(i) = \max_k \{v_i^k\}$$

$$f_n(i) = \max_k \left\{ v_i^k + \sum_{j=1}^{m} p_{ij}^k f_{n+1}(j) \right\}, \ n = 1, 2 \ldots, N - 1$$

To illustrate the computation of v_i^k, consider the case in which no fertilizer is used ($k = 1$).

$$v_1^1 = .2 \times 7 + .5 \times 6 + .3 \times 3 = 5.3$$

$$v_2^1 = 0 \times 0 + .5 \times 5 + .5 \times 1 = 3$$

$$v_3^1 = 0 \times 0 + 0 \times 0 + 1 \times -1 = -1$$

Thus, if the soil condition is good, a single transition yields 5.3 for that year; if it is fair, the yield is 3; and if it is poor, the yield is -1.

Example 19.2-1

In this example, we solve the gardener problem using the data summarized in the matrices $\mathbf{P}^1$, $\mathbf{P}^2$, $\mathbf{R}^1$, and $\mathbf{R}^2$, given a horizon of 3 years ($N = 3$).

Because the values of v_i^k will be used repeatedly in the computations, they are summarized here for convenience. Recall that $k = 1$ represents "do not fertilize" and $k = 2$ represents "fertilize."

i	v_i^1	v_i^2
1	5.3	4.7
2	3	3.1
3	−1	.4

Stage 3.

	v_i^k		Optimal solution	
i	$k = 1$	$k = 2$	$f_3(i)$	k^*
1	5.3	4.7	5.3	1
2	3	3.1	3.1	2
3	−1	.4	.4	2

Stage 2.

	$v_i^k + p_{i1}^k f_3(1) + p_{i2}^k f_3(2) + p_{i3}^k f_3(3)$		Optimal solution	
i	$k = 1$	$k = 2$	$f_2(i)$	k^*
1	$5.3 + .2 \times 5.3 + .5 \times 3.1$ $+ .3 \times .4 = 8.03$	$4.7 + .3 \times 5.3 + .6 \times 3.1$ $+ .1 \times .4 = 8.19$	8.19	2
2	$3 + 0 \times 5.3 + .5 \times 3.1$ $+ .5 \times .4 = 4.75$	$3.1 + .1 \times 5.3 + .6 \times 3.1$ $+ .3 \times .4 = 5.61$	5.61	2
3	$-1 + 0 \times 5.3 + 0 \times 3.1$ $+ 1 \times .4 = -.6$	$.4 + .05 \times 5.3 + .4 \times 3.1$ $+ .55 \times .4 = 2.13$	2.13	2

Stage 1.

	$v_i^k + p_{i1}^k f_2(1) + p_{i2}^k f_2(2) + p_{i3}^k f_2(3)$		Optimal solution	
i	$k = 1$	$k = 2$	$f_1(i)$	k^*
1	$5.3 + .2 \times 8.19 + .5 \times 5.61$ $+ .3 \times 2.13 = 10.38$	$4.7 + .3 \times 8.19 + .6 \times 5.61$ $+ .1 \times 2.13 = 10.74$	10.74	2
2	$3 + 0 \times 8.19 + .5 \times 5.61$ $+ .5 \times 2.13 = 6.87$	$3.1 + .1 \times 8.19 + .6 \times 5.61$ $+ .3 \times 2.13 = 7.92$	7.92	2
3	$-1 + 0 \times 8.19 + 0 \times 5.61$ $+ 1 \times 2.13 = 1.13$	$.4 + .05 \times 8.19 + .4 \times 5.61$ $+ .55 \times 2.13 = 4.23$	4.23	2

The optimal solution shows that for years 1 and 2, the gardener should apply fertilizer ($k^* = 2$) regardless of the state of the system (soil condition, as revealed by the chemical tests). In year 3, fertilizer should be applied only if the system is in state 2 or 3 (fair or poor soil condition). The total expected revenues for the three years are $f_1(1) = 10.74$ if the state of the system in year 1 is good, $f_1(2) = 7.92$ if it is fair, and $f_1(3) = 4.23$ if it is poor.

The finite-horizon problem can be generalized in two ways. First, the transition probabilities and their return functions need not be the same for all years. Second, a discounting factor can be applied to the expected revenue of the successive stages so that $f_1(i)$ will equal the *present value* of the expected revenues of all the stages.

The first generalization requires the return values r_{ij}^k and transition probabilities p_{ij}^k to be functions of the stage, n, as the following DP recursive equation shows

$$f_N(i) = \max_k \{v_i^{k,N}\}$$

$$f_n(i) = \max_k \left\{ v_i^{k,n} + \sum_{j=1}^{m} p_{ij}^{k,n} f_{n+1}(j) \right\},\ n = 1, 2, \ldots, N - 1$$

where

$$v_i^{k,n} = \sum_{j=1}^{m} p_{ij}^{k,n} r_{ij}^{k,n}$$

In the second generalization, given α (< 1) is the discount factor per year such that D dollars a year from now have a present value of αD dollars, the new recursive equation becomes

$$f_N(i) = \max_k\{v_i^k\}$$

$$f_n(i) = \max_k\left\{v_i^k + \alpha\sum_{j=1}^{m} p_{ij}^k f_{n+1}(j)\right\},\ n = 1, 2, \ldots, N - 1$$

PROBLEM SET 19.2A

1. A company reviews the state of one of its important products annually and decides whether it is successful (state 1) or unsuccessful (state 2). The company must decide whether or not to advertise the product to further promote sales. The following matrices, $\mathbf{P}^1$ and $\mathbf{P}^2$, provide the transition probabilities with and without advertisement during any year. The associated returns are given by the matrices $\mathbf{R}^1$ and $\mathbf{R}^2$. Find the optimal decisions over the next 3 years.

$$\mathbf{P}^1 = \begin{pmatrix} .9 & .1 \\ .6 & .4 \end{pmatrix},\ \mathbf{R}^1 = \begin{pmatrix} 2 & -1 \\ 1 & -3 \end{pmatrix}$$

$$\mathbf{P}^2 = \begin{pmatrix} .7 & .3 \\ .2 & .8 \end{pmatrix},\ \mathbf{R}^2 = \begin{pmatrix} 4 & 1 \\ 2 & -1 \end{pmatrix}$$

2. A company can advertise through radio, TV, or newspaper. The weekly costs of advertisement on the three media are estimated at \$200, \$900, and \$300, respectively. The company can classify its sales volume during each week as (1) fair, (2) good, or (3) excellent. A summary of the transition probabilities associated with each advertisement medium follows.

$$\begin{array}{c} \text{Radio} \\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & .4 & .5 & .1 \\ 2 & .1 & .7 & .2 \\ 3 & .1 & .2 & .7 \end{array} \end{array} \quad \begin{array}{c} \text{TV} \\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & .7 & .2 & .1 \\ 2 & .3 & .6 & .1 \\ 3 & .1 & .7 & .2 \end{array} \end{array} \quad \begin{array}{c} \text{Newspaper} \\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & .2 & .5 & .3 \\ 2 & 0 & .7 & .3 \\ 3 & 0 & .2 & .8 \end{array} \end{array}$$

The corresponding weekly returns (in thousands of dollars) are

$$\begin{array}{c} \text{Radio} \\ \begin{pmatrix} 400 & 520 & 600 \\ 300 & 400 & 700 \\ 200 & 250 & 500 \end{pmatrix} \end{array} \quad \begin{array}{c} \text{TV} \\ \begin{pmatrix} 1000 & 1300 & 1600 \\ 800 & 1000 & 1700 \\ 600 & 700 & 1100 \end{pmatrix} \end{array} \quad \begin{array}{c} \text{Newspaper} \\ \begin{pmatrix} 400 & 530 & 710 \\ 350 & 450 & 800 \\ 250 & 400 & 650 \end{pmatrix} \end{array}$$

Find the optimal advertisement policy over the next 3 weeks.

3. *Inventory Problem.* An appliance store can place orders for refrigerators at the beginning of each month for immediate delivery. A fixed cost of \$100 is incurred every time an order is placed. The storage cost per refrigerator per month is \$5. The penalty for running out of stock is estimated at \$150 per refrigerator per month. The monthly demand is given by the following pdf:

Demand x	0	1	2
$p(x)$	.2	.5	.3

The store's policy is that the maximum stock level should not exceed two refrigerators in any single month. Determine the following:

(a) The transition probabilities for the different decision alternatives of the problem.

(b) The expected inventory cost per month as a function of the state of the system and the decision alternative.

(c) The optimal ordering policy over the next 3 months.

4. Repeat Problem 3 assuming that the pdf of demand over the next quarter changes according to the following table:

	Month		
Demand, x	*1*	*2*	*3*
0	.1	.3	.2
1	.4	.5	.4
2	.5	.2	.4

19.3 INFINITE-STAGE MODEL

There are two methods for solving the infinite-stage problem. The first method calls for evaluating *all* possible stationary policies of the decision problem. This is equivalent to an *exhaustive enumeration* process and can be used only if the number of stationary policies is reasonably small. The second method, called *policy iteration,* is generally more efficient because it determines the optimum policy iteratively.

19.3.1 Exhaustive Enumeration Method

Suppose that the decision problem has S stationary policies, and assume that $\mathbf{P}^s$ and $\mathbf{R}^s$ are the (one-step) transition and revenue matrices associated with the policy, $s = 1, 2, \ldots, S$. The steps of the enumeration method are as follows.

Step 1. Compute v_i^s, the expected one-step (one-period) revenue of policy s given state i, $i = 1, 2, \ldots, m$.

Step 2. Compute π_i^s, the long-run stationary probabilities of the transition matrix $\mathbf{P}^s$ associated with policy s. These probabilities, when they exist, are computed from the equations

$$\boldsymbol{\pi}^s \mathbf{P}^s = \boldsymbol{\pi}^s$$

$$\pi_1^s + \pi_2^s + \cdots + \pi_m^s = 1$$

where $\boldsymbol{\pi}^s = (\pi_1^s, \pi_2^s, \cdots, \pi_m^s)$.

Step 3. Determine E^s, the expected revenue of policy s per transition step (period), by using the formula

$$E^s = \sum_{i=1}^{m} \pi_i^s v_i^s$$

Step 4. The optimal policy s^* is determined such that

$$E^{s^*} = \max_s\{E^s\}$$

We illustrate the method by solving the gardener problem for an infinite-period planning horizon.

Example 19.3-1

The gardener problem has a total of eight stationary policies, as the following table shows:

Stationary policy, s	Action
1	Do not fertilize at all.
2	Fertilize regardless of the state.
3	Fertilize if in state 1.
4	Fertilize if in state 2.
5	Fertilize if in state 3.
6	Fertilize if in state 1 or 2.
7	Fertilize if in state 1 or 3.
8	Fertilize if in state 2 or 3.

The matrices $\mathbf{P}^s$ and $\mathbf{R}^s$ for policies 3 through 8 are derived from those of policies 1 and 2 and are given as

$$\mathbf{P}^1 = \begin{pmatrix} .2 & .5 & .3 \\ 0 & .5 & .5 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{R}^1 = \begin{pmatrix} 7 & 6 & 3 \\ 0 & 5 & 1 \\ 0 & 0 & -1 \end{pmatrix}$$

$$\mathbf{P}^2 = \begin{pmatrix} .3 & .6 & .1 \\ .1 & .6 & .3 \\ .05 & .4 & .55 \end{pmatrix} \qquad \mathbf{R}^2 = \begin{pmatrix} 6 & 5 & -1 \\ 7 & 4 & 0 \\ 6 & 3 & -2 \end{pmatrix}$$

$$\mathbf{P}^3 = \begin{pmatrix} .3 & .6 & .1 \\ 0 & .5 & .5 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{R}^3 = \begin{pmatrix} 6 & 5 & -1 \\ 0 & 5 & 1 \\ 0 & 0 & -1 \end{pmatrix}$$

$$\mathbf{P}^4 = \begin{pmatrix} .2 & .5 & .3 \\ .1 & .6 & .3 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{R}^4 = \begin{pmatrix} 7 & 6 & 3 \\ 7 & 4 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

$$\mathbf{P}^5 = \begin{pmatrix} .2 & .5 & .3 \\ 0 & .5 & .5 \\ .05 & .4 & .55 \end{pmatrix} \qquad \mathbf{R}^5 = \begin{pmatrix} 7 & 6 & 3 \\ 0 & 5 & 1 \\ 6 & 3 & -2 \end{pmatrix}$$

$$\mathbf{P}^6 = \begin{pmatrix} .3 & .6 & .1 \\ .1 & .6 & .3 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{R}^6 = \begin{pmatrix} 6 & 5 & -1 \\ 7 & 4 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

$$\mathbf{P}^7 = \begin{pmatrix} .3 & .6 & .1 \\ 0 & .5 & .5 \\ .05 & .4 & .55 \end{pmatrix} \qquad \mathbf{R}^7 = \begin{pmatrix} 6 & 5 & -1 \\ 0 & 5 & 1 \\ 6 & 3 & -2 \end{pmatrix}$$

$$\mathbf{P}^8 = \begin{pmatrix} .2 & .5 & .3 \\ .1 & .6 & .3 \\ .05 & .4 & .55 \end{pmatrix} \qquad \mathbf{R}^8 = \begin{pmatrix} 7 & 6 & 3 \\ 7 & 4 & 0 \\ 6 & 3 & -2 \end{pmatrix}$$

The values of v_i^s can thus be computed as given in the following table.

	v_i^s		
s	$i = 1$	$i = 2$	$i = 3$
1	5.3	3.0	−1.0
2	4.7	3.1	0.4
3	4.7	3.0	−1.0
4	5.3	3.1	−1.0
5	5.3	3.0	0.4
6	4.7	3.1	−1.0
7	4.7	3.0	0.4
8	5.3	3.1	0.4

The computations of the stationary probabilities are achieved by using the equations

$$\boldsymbol{\pi}^s \mathbf{P}^s = \boldsymbol{\pi}^s$$

$$\pi_1 + \pi_2 + \cdots + \pi_m = 1$$

As an illustration, consider $s = 2$. The associated equations are

$$\begin{aligned} .3\pi_1 + .1\pi_2 + .05\pi_3 &= \pi_1 \\ .6\pi_1 + .6\pi_2 + .4\pi_3 &= \pi_2 \\ .1\pi_1 + .3\pi_2 + .55\pi_3 &= \pi_3 \\ \pi_1 + \pi_2 + \pi_3 &= 1 \end{aligned}$$

(Notice that one of the first three equations is redundant.) The solution yields

$$\pi_1^2 = \tfrac{6}{59},\ \pi_2^2 = \tfrac{31}{59},\ \pi_3^2 = \tfrac{22}{59}$$

In this case, the expected yearly revenue is

$$\begin{aligned} E^2 &= \pi_1^2 v_1^2 + \pi_2^2 v_2^2 + \pi_3^2 v_3^2 \\ &= \left(\tfrac{6}{59}\right) \times 4.7 + \left(\tfrac{31}{59}\right) \times 3.1 + \left(\tfrac{22}{59}\right) \times .4 = 2.256 \end{aligned}$$

The following table summarizes π^s and E^s for all the stationary policies. (Although this will not affect the computations in any way, note that each of policies 1, 3, 4, and 6 has an absorbing state: state 3. This is the reason $\pi_1 = \pi_2 = 0$ and $\pi_3 = 1$ for all these policies.)

s	π_1^s	π_2^s	π_3^s	E^s
1	0	0	1	−1
2	$\frac{6}{59}$	$\frac{31}{59}$	$\frac{22}{59}$	**2.256**
3	0	0	1	0.4
4	0	0	1	−1
5	$\frac{5}{154}$	$\frac{69}{154}$	$\frac{80}{154}$	1.724
6	0	0	1	−1
7	$\frac{5}{137}$	$\frac{62}{137}$	$\frac{70}{137}$	1.734
8	$\frac{12}{135}$	$\frac{69}{135}$	$\frac{54}{135}$	2.216

Policy 2 yields the largest expected yearly revenue. The optimum long-range policy calls for applying fertilizer regardless of the state of the system.

PROBLEM SET 19.3A

1. Solve Problem 2, Set 19.2a for an infinite number of periods using the exhaustive enumeration method.
2. Solve Problem 2, Set 19.2a for an infinite planning horizon using the exhaustive enumeration method.
3. Solve Problem 3, Set 19.2a by the exhaustive enumeration method assuming an infinite horizon.

19.3.2 Policy Iteration Method Without Discounting

To appreciate the difficulty associated with the exhaustive enumeration method, let us assume that the gardener had four courses of action (alternatives) instead of two: (1) do not fertilize, (2) fertilize once during the season, (3) fertilize twice, and (4) fertilize three times. In this case, the gardener would have a total of $4^3 = 256$ stationary policies. By increasing the number of alternatives from 2 to 4, the number of stationary policies "soars" exponentially from 8 to 256. Not only is it difficult to enumerate all the policies explicitly, but the amount of computations may also be prohibitively large. This is the reason we are interested in developing the policy iteration method.

In Section 19.2, we have shown that, for any specific policy, the expected total return at stage n is expressed by the recursive equation

$$f_n(i) = v_i + \sum_{j=1}^{m} p_{ij} f_{n+1}(j),\ i = 1, 2, \ldots, m$$

This recursive equation is the basis for the development of the policy iteration method. However, the present form must be modified slightly to allow us to study the asymptotic behavior of the process. We define η as the number of stages *remaining* for consideration. This is in contrast with n in the equation, which defines stage n. The recursive equation is thus written as

$$f_\eta(i) = v_i + \sum_{j=1}^{m} p_{ij} f_{\eta-1}(j),\ i = 1, 2, 3, \ldots, m$$

Note that f_η is the cumulative expected revenue given that η is the number of stages remaining for consideration. With the new definition, the asymptotic behavior of the process can be studied by letting $\eta \to \infty$.

Given that

$$\pi = (\pi_1, \pi_2, \ldots, \pi_m)$$

is the steady-state probability vector of the transition matrix $\mathbf{P} = \|p_{ij}\|$ and

$$E = \pi_1 v_1 + \pi_2 v_2 + \ldots + \pi_m v_m$$

is the expected revenue per stage as computed in Section 19.3.1, it can be shown that for very large η,

$$f_\eta(i) = \eta E + f(i)$$

where $f(i)$ is a constant term representing the asymptotic intercept of $f_\eta(i)$ given state i.

Because $f_\eta(i)$ is the cumulative optimum return for η remaining stages given state i, and E is the expected revenue *per stage*, we can see intuitively why $f_\eta(i)$ equals ηE plus a correction factor $f(i)$ that accounts for the specific state i. This result assumes that $\eta \to \infty$.

Now, using this information, the recursive equation is written as

$$\eta E + f(i) = v_i + \sum_{j=1}^{m} p_{ij}\{(\eta-1)E + f(j)\},\ i = 1, 2, \ldots, m$$

Simplifying this equation, we get

$$E + f(i) - \sum_{j=1}^{m} p_{ij} f(j) = v_i,\ i = 1, 2, \ldots, m$$

Here, we have m equations in $m + 1$ unknowns, $f(1)$, $f(2)$, $\ldots$, $f(m)$, and E.

As in Section 19.3.1, our objective is to determine the optimum policy that yields the maximum value of E. Because there are m equations in $m + 1$ unknowns, the optimum value of E cannot be determined in one step. Instead, a two-step iterative approach is utilized which, starting with an arbitrary policy, will determine a new policy that yields a better value of E. The iterative process ends when two successive policies are identical.

1. *Value Determination Step.* Choose arbitrary policy s. Using its associated matrices $\mathbf{P}^s$ and $\mathbf{R}^s$ and arbitrarily assuming $f^s(m) = 0$, solve the equations

$$E^s + f^s(i) - \sum_{j=1}^{m} p_{ij}^s f^s(j) = v_i^s,\ i = 1, 2, \ldots, m$$

in the unknowns E^s, $f^s(1)$, $\ldots$, and $f^s(m - 1)$. Go to the policy improvement step.

2. *Policy Improvement Step.* For each state i, determine the policy t that corresponds to

$$\max_{k}\left\{v_i^k + \sum_{j=1}^{m} p_{ij}^k f^s(j)\right\},\ i = 1, 2, \ldots, m$$

The values of $f^s(j)$, $j = 1, 2, \ldots, m$, are those determined in the value determination step. The resulting optimum decisions for states $1, 2, \ldots,$ and m constitute the new policy t. If s and t are identical, t is optimum. Otherwise, set $s = t$ and return to the value determination step.

Example 19.3-2

We solve the gardener problem by the policy iteration method.

Let us start with the arbitrary policy that calls for not applying fertilizer. The associated matrices are

$$\mathbf{P} = \begin{pmatrix} .2 & .5 & .3 \\ 0 & .5 & .5 \\ 0 & 0 & 1 \end{pmatrix},\ \mathbf{R} = \begin{pmatrix} 7 & 6 & 3 \\ 0 & 5 & 1 \\ 0 & 0 & -1 \end{pmatrix}$$

The equations of the value iteration step are

$$\begin{aligned}
E + f(1) - .2f(1) - .5f(2) - .3f(3) &= 5.3 \\
E + f(2) \qquad\qquad - .5f(2) - .5f(3) &= 3 \\
E + f(3) \qquad\qquad\qquad\qquad - f(3) &= -1
\end{aligned}$$

If we arbitrarily let $f(3) = 0$, the equations yield the solution

$$E = -1,\ f(1) = 12.88,\ f(2) = 8,\ f(3) = 0$$

Next, we apply the policy improvement step. The associated calculations are shown in the following tableau.

	$v_i^k + p_{i1}^k f(1) + p_{i2}^k f(2) + p_{i3}^k f(3)$		Optimal solution	
i	$k = 1$	$k = 2$	$f(i)$	k^*
1	$5.3 + .2 \times 12.88 + .5 \times 8 + .3 \times 0 = 11.876$	$4.7 + .3 \times 12.88 + .6 \times 8 + .1 \times 0 = 13.36$	13.36	2
2	$3 + 0 \times 12.88 + .5 \times 8 + .5 \times 0 = 7$	$3.1 + .1 \times 12.88 + .6 \times 8 + .3 \times 0 = 9.19$	9.19	2
3	$-1 + 0 \times 12.88 + 0 \times 8 + 1 \times 0 = -1$	$.4 + .05 \times 12.88 + .4 \times 8 + .55 \times 0 = 4.24$	4.24	2

The new policy calls for applying fertilizer regardless of the state. Because the new policy differs from the preceding one, the value determination step is entered again. The matrices associated with the new policy are

$$\mathbf{P} = \begin{pmatrix} .3 & .6 & .1 \\ .1 & .6 & .3 \\ .05 & .4 & .55 \end{pmatrix},\ \mathbf{R} = \begin{pmatrix} 6 & 5 & -1 \\ 7 & 4 & 0 \\ 6 & 3 & -2 \end{pmatrix}$$

These matrices yield the following equations:

$$\begin{aligned} E + f(1) - .3f(1) - .6f(2) - .1f(3) &= 4.7 \\ E + f(2) - .1f(1) - .6f(2) - .3f(3) &= 3.1 \\ E + f(3) - .05f(1) - .4f(2) - .55f(3) &= .4 \end{aligned}$$

Again, letting $f(3) = 0$, we get the solution

$$E = 2.26,\ f(1) = 6.75,\ f(2) = 3.80,\ f(3) = 0$$

The computations of the policy improvement step are given in the following tableau.

	$v_i^k + p_{i1}^k f(1) + p_{i2}^k f(2) + p_{i3}^k f(3)$		Optimal solution	
i	$k = 1$	$k = 2$	$f(i)$	k^*
1	$5.3 + .2 \times 6.75 + .5 \times 3.80 + .3 \times 0 = 8.55$	$4.7 + .3 \times 6.75 + .6 \times 3.80 + .1 \times 0 = 9.01$	9.01	2
2	$3 + 0 \times 6.75 + .5 \times 3.80 + .5 \times 0 = 4.90$	$3.1 + .1 \times 6.75 + .6 \times 3.80 + .3 \times 0 = 6.06$	6.06	2
3	$-1 + 0 \times 6.75 + 0 \times 3.80 + 1 \times 0 = -1$	$.4 + .05 \times 6.75 + .4 \times 3.80 + .55 \times 0 = 2.26$	2.26	2

The new policy, which calls for applying fertilizer regardless of the state, is identical with the preceding one. Thus the last policy is optimal, and the iterative process ends. This is the same conclusion obtained by the exhaustive enumeration method (Section 19.3.1). Note, however, that the policy iteration method converges quickly to the optimum policy, a typical characteristic of the new method.

PROBLEM SET 19.3B

1. Assume in Problem 1, Set 19.2a that the planning horizon is infinite. Solve the problem by the policy iteration method, and compare the results with those of Problem 1, Set 19.3a.
2. Solve Problem 2, Set 19.2a by the policy iteration method, assuming an infinite planning horizon. Compare the results with those of Problem 2, Set 19.3a.
3. Solve Problem 3, Set 19.2a by the policy iteration method assuming an infinite planning horizon, and compare the results with those of Problem 3, Set 19.3a.

19.3.3 Policy Iteration Method With Discounting

The policy iteration algorithm can be extended to include discounting. Given the discount factor α (< 1), the finite-stage recursive equation can be written as (see Section 19.2)

$$f_\eta(i) = \max_k \left\{ v_i^k + \alpha \sum_{j=1}^{m} p_{ij}^k f_{\eta-1}(j) \right\}$$

(Note that η represents the number of stages *to go.*) It can be proved that as $\eta \to \infty$ (infinite stage model), $f_\eta(i) = f(i)$, where $f(i)$ is the expected present-worth (discounted) revenue given that the system is in state i and operating over an infinite horizon. Thus the long-run behavior of $f_\eta(i)$ as $\eta \to \infty$ is independent of the value of η. This is in contrast with the case of no discounting where $f_\eta(i) = \eta E + f(i)$. This result should be expected because in discounting the effect of future revenues will asymptotically diminish to zero. Indeed, the present worth $f(i)$ should approach a constant value as $\eta \to \infty$.

Based on this information, the steps of the policy iterations are modified as follows.

1. *Value Determination Step.* For an arbitrary policy s with its matrices $\mathbf{P}^s$ and $\mathbf{R}^s$, solve the m equations

$$f^s(i) - \alpha \sum_{j=1}^{m} p_{ij}^s f^s(j) = v_i^s,\ i = 1, 2, \ldots, m$$

in the m unknowns $f^s(1), f^s(2), \ldots, f^s(m)$.

2. *Policy Improvement Step.* For each state i, determine the policy t that corresponds to

$$\max_k \left\{ v_i^k + \alpha \sum_{j=1}^{m} p_{ij}^k f^s(j) \right\},\ i = 1, 2, \ldots, m$$

$f^s(j)$ is obtained from the value determination step. If the resulting policy t is the same as s, stop; t is optimum. Otherwise, set $s = t$ and return to the value determination step.

Example 19.3-3

We will solve Example 19.3-2 using the discounting factor $\alpha = .6$.

Starting with the arbitrary policy, $s = \{1, 1, 1\}$. The associated matrices $\mathbf{P}$ and $\mathbf{R}$ ($\mathbf{P}^1$ and $\mathbf{R}^1$ in Example 19.3-1) yield the equations

$$\begin{aligned} f(1) - .6[.2f(1) + .5f(2) + .3f(3)] &= 5.3 \\ f(2) - .6[\qquad\quad .5f(2) + .5f(3)] &= 3. \\ f(3) - .6[\qquad\qquad\qquad\quad f(3)] &= -1. \end{aligned}$$

The solution of these equations yields

$$f_1 = 6.61,\ f_2 = 3.21,\ f_3 = -2.5$$

A summary of the policy improvement iteration is given in the following tableau:

	$v_i^k + .6[p_{i1}^k f(1) + p_{i2}^k f(2) + p_{i3}^k f(3)]$		Optimal solution	
i	$k = 1$	$k = 2$	$f(i)$	k^*
1	$5.3 + .6[.2 \times 6.61 + .5 \times 3.21 + .3 \times -2.5] = 6.61$	$4.7 + .6[.3 \times 6.61 + .6 \times .21 + .1 \times -2.5] = 6.90$	6.90	2
2	$3 + .6[0 \times 6.61 + .5 \times 3.21 + .5 \times -2.5] = 3.21$	$3.1 + .6[.1 \times 6.61 + .6 \times 3.21 + .3 \times -2.5] = 4.2$	4.2	2
3	$-1 + .6[0 \times 6.61 + 0 \times 3.21 + 1 \times -2.5] = -2.5$	$.4 + .6[.05 \times 6.61 + .4 \times 3.21 + .55 \times -2.5] = .54$	.54	2

The value determination step using $\mathbf{P}^2$ and $\mathbf{R}^2$ (Example 19.3-1) yields the following equations:

$$\begin{aligned} f(1) - .6[\ .3f(1) + .6f(2) + \ .1f(3)] &= 4.7 \\ f(2) - .6[\ .1f(1) + .6f(2) + \ .3f(3)] &= 3.1 \\ f(3) - .6[.05f(1) + .4f(2) + .55f(3)] &= \ .4 \end{aligned}$$

The solution of these equations yields

$$f(1) = 8.89,\ f(2) = 6.62,\ f(3) = 3.37$$

The policy improvement step yields the following tableau:

	$v_i^k + .6[p_{i1}^k f(1) + p_{i2}^k f(2) + p_{i3}^k(3)]$		Optimal solution	
i	$k = 1$	$k = 2$	$f(i)$	k^*
1	$5.3 + .6[.2 \times 8.89 + .5 \times 6.62 + .3 \times 3.37] = 8.96$	$4.7 + .6[.3 \times 8.89 + .6 \times 6.62 + .1 \times 3.37] = 8.89$	8.96	1
2	$3 + .6[0 \times 8.89 + .5 \times 6.62 + .5 \times 3.37] = 6.00$	$3.1 + .6[.1 \times 8.89 + .6 \times 6.62 + .3 \times 3.37] = 6.62$	6.62	2
3	$-1 + .6[0 \times 8.89 + 0 \times 6.62 + 1 \times 3.37] = 1.02$	$.4 + .6[.05 \times 8.89 + .4 \times 6.62 + .55 \times 3.37] = 3.37$	3.37	2

Because the new policy (1, 2, 2) differs from the preceding one, the value determination step is entered again using $\mathbf{P}^3$ and $\mathbf{R}^3$ (Example 19.3-1). This results in the following equations:

$$\begin{aligned} f(1) - .6[\ .2f(1) + .5f(2) + \ .3f(3)] &= 5.3 \\ f(2) - .6[\ .1f(1) + .6f(2) + \ .3f(3)] &= 3.1 \\ f(3) - .6[.05f(1) + .4f(2) + .55f(3)] &= \ .4 \end{aligned}$$

The solution of these equations yields

$$f(1) = 8.97,\ f(2) = 6.63,\ f(3) = 3.38$$

The policy improvement step yields the following tableau:

	$v_i^k + .6[p_{i1}^k f(1) + p_{i2}^k f(2) + p_{i3}^k f(3)]$		Optimal solution	
i	$k = 1$	$k = 2$	$f(i)$	k^*
1	$5.3 + .6[.2 \times 8.97 + .5 \times 6.63 + .3 \times 3.38] = 8.97$	$4.7 + .6[.3 \times 8.97 + .6 \times 6.63 + .1 \times 3.38] = 8.90$	8.98	1
2	$3 + .6[0 \times 8.97 + .5 \times 6.63 + .5 \times 3.38] = 6.00$	$3.1 + .6[.1 \times 8.97 + .6 \times 6.63 + .3 \times 3.38] = 6.63$	6.63	2
3	$-1 + .6[0 \times 8.97 + 0 \times 6.63 + 1 \times 3.38] = 1.03$	$.4 + .6[.05 \times 8.97 + .4 \times 6.63 + .55 \times 3.38] = 3.37$	3.37	2

Because the new policy (1, 2, 2) is identical with the preceding one, it is optimal. Note that discounting has resulted in a different optimal policy that calls for not applying fertilizer if the state of the system is good (state 3).

PROBLEM SET 19.3C

1. Repeat the problems listed, assuming the discount factor $\alpha = .9$.

(a) Problem 1, Set 19.3b

(b) Problem 2, Set 19.3b

(c) Problem 3, Set 19.3b

19.4 LINEAR PROGRAMMING SOLUTION

The infinite-state Markovian decision problems, both with discounting and without, can be formulated and solved as linear programs. We consider the no-discounting case first.

Section 19.3.1 shows that the infinite-state Markovian problem with no discounting ultimately reduces to determining the optimal policy, s^*, which corresponds to

$$\max_{s\varepsilon S}\left\{\sum_{j=1}^{m}\pi_i^s v_i^s \middle| \boldsymbol{\pi}^s\mathbf{P}^s = \boldsymbol{\pi}^s,\ \pi_1^s + \pi_2^s + \ldots + \pi_m^s = 1,\ \pi_i^s \geq 0\ i = 1, 2, \ldots, m\right\}$$

The set S is the collection of all possible policies of the problem. The constraints of the problem ensure that π_i^s, $i = 1, 2, \ldots, m$, represent the steady-state probabilities of the Markov chain $\mathbf{P}^s$.

The problem is solved in Section 19.3.1 by exhaustive enumeration. Specifically, each policy s is specified by a fixed set of actions (as illustrated by the gardener problem in Example 19.3-1). The same problem is the basis for the development of the LP formulation. However, we need to modify the unknowns of the problem such that the optimal solution *automatically* determines the optimal action (alternative) k when the system is in state i. The collection of all the optimal actions will then define s^*, the optimal policy.

Let

q_i^k = Conditional probability of choosing alternative k given that the system is in state i

The problem may thus be expressed as

$$\text{Maximize } E = \sum_{i=1}^{m}\pi_i\left(\sum_{k=1}^{K}q_i^k v_i^k\right)$$

subject to

$$\pi_j = \sum_{i=1}^{m}\pi_i p_{ij},\ j = 1, 2, \ldots, m$$

$$\pi_1 + \pi_2 + \cdots + \pi_m = 1$$

$$q_i^1 + q_i^2 + \cdots + q_i^K = 1,\ i = 1, 2, \ldots, m$$

$$\pi_i \geq 0,\ q_i^k \geq 0,\ \text{for all } i \text{ and } k$$

Note that p_{ij} is a function of the policy selected and hence of the specific alternatives k of the policy.

The problem can be converted into a linear program by making proper substitutions involving q_i^k. Observe that the formulation is equivalent to the original one in Section 19.3.1 only if $q_i^k = 1$ for exactly *one* k for each i, which will reduce the sum $\sum_{k=1}^{K} q_i^k v_i^k$ to $v_i^{k^*}$, where k^* is the optimal alternative chosen. The linear program we develop here does account for this condition automatically.

Define

$$w_{ik} = \pi_i q_i^k, \text{ for all } i \text{ and } k$$

By definition w_{ik} represents the *joint* probability of state i making decision k. From probability theory

$$\pi_i = \sum_{k=1}^{K} w_{ik}$$

Hence,

$$q_i^k = \frac{w_{ik}}{\sum_{k=1}^{K} w_{ik}}$$

Thus the restriction $\sum_{i=1}^{m} \pi_i = 1$ can be written as

$$\sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} = 1$$

Also, the restriction $\sum_{k=1}^{K} q_i^k = 1$ is automatically implied by the way we defined q_i^k in terms of w_{ik}. (Verify!) Thus the problem can be written as

$$\text{Maximize } E = \sum_{i=1}^{m} \sum_{k=1}^{K} v_i^k w_{ik}$$

subject to

$$\sum_{k=1}^{K} w_{jk} - \sum_{i=1}^{m} \sum_{k=1}^{K} p_{ij}^k w_{ik} = 0, \; j = 1, 2, \ldots, m$$

$$\sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} = 1$$

$$w_{ik} \geq 0, \; i = 1, 2, \ldots, m; \quad k = 1, 2, \ldots, K$$

The resulting model is a linear program in w_{ik}. Its optimal solution automatically guarantees that $q_i^k = 1$ for one k for each i. First, note that the linear program has m independent equations (one of the equations associated with $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$ is redundant). Hence, the problem must have m basic variables. It can be shown that w_{ik} must be strictly positive for at least one k for each i. From these two results, we conclude that

$$q_i^k = \frac{w_{ik}}{\sum_{k=1}^{K} w_{ik}}$$

can assume a binary value (0 or 1) only. (As a matter of fact the preceding result also shows that $\pi_i = \sum_{k=1}^{K} w_{ik} = w_{ik^*}$, where k^* is the alternative corresponding to $w_{ik} > 0$.)

Example 19.4-1

The following is an LP formulation of the gardener problem without discounting:

$$\text{Maximize } E = 5.3w_{11} + 4.7w_{12} + 3w_{21} + 3.1w_{22} - w_{31} + .4w_{32}$$

subject to

$$\begin{aligned}
w_{11} + w_{12} - (.2w_{11} + .3w_{12} \qquad\qquad + .1w_{22} \qquad\qquad + 0.5w_{32}) &= 0\\
w_{21} + w_{22} - (.5w_{11} + .6w_{12} + .5w_{21} + .6w_{22} \qquad\qquad + \ .4w_{32}) &= 0\\
w_{31} + w_{32} - (.3w_{11} + .1w_{12} + .5w_{21} + .3w_{22} + w_{31} + .55w_{32}) &= 0\\
w_{11} + w_{12} + w_{21} + w_{22} + w_{31} + w_{32} &= 1\\
w_{ik} \geq 0, \quad \text{for all } i \text{ and } k&
\end{aligned}$$

The optimal solution is $w_{11} = w_{12} = w_{31} = 0$ and $w_{12} = .1017$, $w_{22} = .5254$, and $w_{32} = .3729$. This result means that $q_1^2 = q_2^2 = q_3^2 = 1$. Thus, the optimal policy selects alternative $k = 2$ for $i = 1, 2,$ and 3. The optimal value of E is $4.7(.1017) + 3.1(.5254) + .4(.3729) = 2.256$. The positive values of w_{ik} exactly equal the values of π_i associated with the optimal policy in the exhaustive enumeration procedure of Example 19.3-1, which demonstrates the direct relationship between the two methods.

We next consider the Markovian decision problem with discounting. In Section 19.3.2 the problem is expressed by the recursive equation

$$f(i) = \max_k \left\{ v_i^k + \alpha \sum_{j=1}^{m} p_{ij}^k f(j) \right\}, \ i = 1, 2, \ldots, m$$

These equations are equivalent to

$$f(i) \geq \alpha \sum_{j=1}^{m} p_{ij}^k f(j) + v_i^k, \text{ for all } i \text{ and } k$$

provided that $f(i)$ achieves its minimum value for each i. Now consider the objective function

$$\text{Minimize} \sum_{i=1}^{m} b_i f(i)$$

where b_i (> 0 for all i) is an arbitrary constant. It can be shown that the optimization of this function subject to the inequalities given will result in the minimum value of $f(i)$. Thus, the problem can be written as

$$\text{Minimize} \sum_{i=1}^{m} b_i f(i)$$

subject to

$$f(i) - \alpha \sum_{i=1}^{m} p_{ij}^k f(j) \geq v_i^k, \text{ for all } i \text{ and } k$$

$$f(i) \text{ unrestricted in sign for all } i$$

Now the dual of the problem is

$$\text{Maximize} \sum_{i=1}^{m} \sum_{k=1}^{K} v_i^k w_{ik}$$

subject to

$$\sum_{k=1}^{K} w_{jk} - \alpha \sum_{i=1}^{m} \sum_{k=1}^{K} p_{ij}^k w_{ik} = b_j,\ j = 1, 2, \ldots, m$$

$$w_{ik} \geq 0, \quad \text{for } i = 1, 2, \ldots, m; \quad k = 1, 2, \ldots, K$$

Example 19.4-2

Consider the gardener problem given the discounting factor $\alpha = .6$. If we let $b_1 = b_2 = b_3 = 1$, the dual LP problem may be written as

$$\text{Maximize } 5.3\text{w}_{11} + 4.7w_{12} + 3w_{12} + 3.1w_{22} - w_{31} + 4w_{32}$$

subject to

$$\begin{aligned}
w_{11} + w_{12} - .6[.2w_{11} + .3w_{12} \qquad\qquad + .1w_{22} \qquad\qquad + 0.5w_{32}] &= 1\\
w_{21} + w_{22} - .6[.5w_{11} + .6w_{12} + .5w_{21} + .6w_{22} \qquad\qquad + .4w_{32}] &= 1\\
w_{31} + w_{32} - .6[.3w_{11} + .1w_{12} + .5w_{21} + .3w_{22} + w_{31} + .55w_{32}] &= 1\\
w_{ik} \geq 0, \quad \text{for all } i \text{ and } k&
\end{aligned}$$

The optimal solution is $w_{12} = w_{21} = w_{31} = 0$ and $w_{11} = 1.5678$, $w_{22} = 3.3528$, and $w_{32} = 2.8145$. The solution shows that that optimal policy is (1, 2, 2).

PROBLEM SET 19.4A

1. Formulate the following problems as linear programs.

(a) Problem 1, Set 19.3b

(b) Problem 2, Set 19.3b

(c) Problem 3, Set 19.3b

19.5 APPENDIX: REVIEW OF MARKOV CHAINS

Consider the discrete points in time $\{t_k\}$ for $k = 1, 2, \ldots$, and let ξ_{t_k} be the random variable that characterizes the state of the system at t_k. The family of random variables $\{\xi_{t_k}\}$ forms a **stochastic process**. The states at time t_k actually represent the (exhaustive and mutually exclusive) outcomes of the system at that time. The number of states may thus be finite or infinite. For example, the Poisson distribution

$$P_n(t) = \frac{e^{-\lambda t}(\lambda t)^n}{n!},\ n = 0, 1, 2, \ldots$$

represents a stochastic process with an infinite number of states. The random variable n represents the number of occurrences between 0 and t (assuming that the system starts at time 0). The states of the system at any time t are thus given by $n = 0, 1, 2, \ldots$.

Another example is the coin tossing game with k trials. Each trial may be viewed as a point in time. The resulting sequence of trials forms a stochastic process. The state of the system at any trial is either a head or a tail.

This section presents a summary of a class of stochastic systems that includes **Markov processes** and **Markov chains**. A Markov chain is a special case of Markov processes. It is used to study the short- and long-run behavior of certain stochastic systems.

19.5.1 Markov Processes

The occurrence of a future state in a Markov process depends on the immediately preceding state and only on it. If $t_0 < t_1 < \ldots < t_n$ $(n = 0, 1, 2, \ldots)$ represents points in time, the family of random variables $\{\xi_{t_n}\}$ is a Markov process if it possesses the following **Markovian property**:

$$P\{\xi_{t_n} = x_n | \xi_{t_{n-1}} = x_{n-1}, \ldots, \xi_{t_0} = x_0\} = P\{\xi_{t_n} = x_n | \xi_{t_{n-1}} = x_{n-1}\}$$

for all possible values of $\xi_{t_0}, \xi_{t_1}, \ldots, \xi_{t_n}$.

The probability $p_{x_{n-1}, x_n} = P\{\xi_{t_n} = x_n | \xi_{t_{n-1}} = x_{n-1}\}$ is called the **transition probability**. It represents the *conditional* probability of the system being in x_n at t_n given it was in x_{n-1} at t_{n-1}. This probability is also referred to as the **one-step transition** because it describes the system between t_{n-1} and t_n. An m-step transition probability is thus defined by

$$p_{x_n, x_{n+m}} = P\{\xi_{t_{n+m}} = x_{n+m} | \xi_n = x_n\}$$

19.5.2 Markov Chains

Let E_j, $j = 0, 1, 2, \ldots$, represent the exhaustive and mutually exclusive outcomes (states) of a system at any time. Initially, at time t_0, the system may be in any of these states. Let $a_j^{(0)}$, $j = 0, 1, 2, \ldots$, be the absolute probability that the system is in state E_j at t_0. Assume further that the system is Markovian.

Define

$$p_{ij} = P\{\xi_{t_n} = j | \xi_{t_{n-1}} = i\}$$

as the one-step transition probability of moving from state i at t_{n-1} to state j at t_n, and assume that these probabilities are stationary over time. The transition probabilities from state E_i to state E_j can be more conveniently arranged in a matrix form as follows:

$$\mathbf{P} = \begin{pmatrix} p_{00} & p_{01} & p_{02} & p_{03} & \cdots \\ p_{10} & p_{11} & p_{12} & p_{13} & \cdots \\ p_{20} & p_{21} & p_{22} & p_{23} & \cdots \\ p_{30} & p_{31} & p_{32} & p_{33} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{pmatrix}$$

The matrix $\mathbf{P}$ is called a **homogeneous transition** matrix because all the transition probabilities p_{ij} are fixed and independent of time. The probabilities p_{ij} must satisfy the conditions

$$\sum_j p_{ij} = 1, \text{ for all } i$$

$$p_{ij} \geq 0, \quad \text{for all } i \text{ and } j$$

The transition matrix **P** together with the initial probabilities $\{a_j^{(0)}\}$ associated with the states E_j completely define a Markov chain. One usually thinks of a Markov chain as describing the transitional behavior of a system over equal intervals. Situations exist where the length of the interval depends on the characteristics of the system and hence may not be equal. This case is referred to as an **imbedded Markov chain**.

Absolute and Transition Probabilities. Given $\{a_j^{(0)}\}$ and **P** of a Markov chain, the absolute probabilities of the system after a specified number of transitions are determined as follows. Let $\{a_j^{(n)}\}$ be the absolute probabilities of the system after n transitions—that is, at t_n. The general expression of $\{a_j^{(n)}\}$ in terms of $\{a_j^{(0)}\}$ and **P** are computed as

$$a_j^{(1)} = a_1^{(0)}p_{1j} + a_2^{(0)}p_{2j} + a_3^{(0)}p_{3j} + \cdots = \sum_i a_i^{(0)}p_{ij}$$

Also,

$$a_j^{(2)} = \sum_i a_i^{(1)}p_{ij} = \sum_i \left(\sum_k a_k^{(0)}p_{ki}\right)p_{ij} = \sum_k a_k^{(0)}\left(\sum_i p_{ki}p_{ij}\right) = \sum_k a_k^{(0)}p_{kj}^{(2)}$$

where $p_{kj}^{(2)} = \sum_i p_{ki}p_{ij}$ is the **two-step** or **second-order transition probability**—that is, the probability of going from state k to state j in exactly two transitions.

It can be shown by induction that

$$a_j^{(n)} = \sum_i a_i^{(0)}\left(\sum_k p_{ik}^{(n-1)}p_{kj}\right) = \sum_i a_i^{(0)}p_{ij}^{(n)}$$

where $p_{ij}^{(n)}$ is the n-step or n-order transition probability given by the recursive formula

$$p_{ij}^{(n)} = \sum_k p_{ik}^{(n-1)}p_{kj}$$

In general,

$$p_{ij}^{(n)} = \sum_k p_{ik}^{(n-m)}p_{kj}^{(m)},\ 0 < m < n,\ \text{ for all } i \text{ and } j$$

These equations are known as **Chapman-Kolomogorov** equations.

The elements of a higher transition matrix $\|p_{ij}^{(n)}\|$ can be obtained directly by matrix multiplication. Thus,

$$\|p_{ij}^{(2)}\| = \|p_{ij}\|\,\|p_{ij}\| = \mathbf{P}^2$$

$$\|p_{ij}^{(3)}\| = \|p_{ij}^{(2)}\|\,\|p_{ij}\| = \mathbf{P}^3$$

and, in general,

$$\|p_{ij}^{(n)}\| = \mathbf{P}^{n-1}\mathbf{P} = \mathbf{P}^n$$

Hence, if the absolute probabilities are defined in vector form as

$$\mathbf{a}^{(n)} = (a_1^{(n)}, a_2^{(n)}, a_3^{(n)}, \ldots)$$

then

$$\mathbf{a}^{(n)} = \mathbf{a}^{(0)}\mathbf{P}^n$$

Example 19.5-1

Consider the following Markov chain with two states,

$$\mathbf{P} = \begin{pmatrix} .2 & .8 \\ .6 & .4 \end{pmatrix}$$

with $\mathbf{a}^{(0)} = (.7, .3)$. Determine $\mathbf{a}^{(1)}$, $\mathbf{a}^{(4)}$, and $\mathbf{a}^{(8)}$.

$$\mathbf{P}^2 = \begin{pmatrix} .2 & .8 \\ .6 & .4 \end{pmatrix}\begin{pmatrix} .2 & .8 \\ .6 & .4 \end{pmatrix} = \begin{pmatrix} .52 & .48 \\ .36 & .64 \end{pmatrix}$$

$$\mathbf{P}^4 = \mathbf{P}^2\mathbf{P}^2 = \begin{pmatrix} .52 & .48 \\ .36 & .64 \end{pmatrix}\begin{pmatrix} .52 & .48 \\ .36 & .64 \end{pmatrix} = \begin{pmatrix} .443 & .557 \\ .418 & .582 \end{pmatrix}$$

$$\mathbf{P}^8 = \mathbf{P}^4\mathbf{P}^4 = \begin{pmatrix} .443 & .557 \\ .418 & .582 \end{pmatrix}\begin{pmatrix} .443 & .557 \\ .418 & .582 \end{pmatrix} = \begin{pmatrix} .4291 & .5709 \\ .4284 & .5716 \end{pmatrix}$$

Thus,

$$\mathbf{a}^{(1)} = (.7 \quad .3)\begin{pmatrix} .2 & .8 \\ .6 & .4 \end{pmatrix} = (.32 \quad .68)$$

$$\mathbf{a}^{(4)} = (.7 \quad .3)\begin{pmatrix} .443 & .557 \\ .418 & .582 \end{pmatrix} = (.436 \quad .564)$$

$$\mathbf{a}^{(8)} = (.7 \quad .3)\begin{pmatrix} .4291 & .5709 \\ .4284 & .5716 \end{pmatrix} = (.4289 \quad .5711)$$

The rows of $\mathbf{P}^8$ tend to be identical. Also, $\mathbf{a}^{(8)}$ tends to be identical with the rows of $\mathbf{P}^{(8)}$. This is the result of the long-run properties of Markov chains, which implies that the long-run absolute probabilities are independent of $\mathbf{a}^{(0)}$. In this case the resulting probabilities are known as the **steady-state probabilities**.

Classification of States in Markov Chains. In Markov chains, we may be interested in the behavior of the system over a short period of time. This is represented by the absolute probabilities as shown in the preceding section. An important investigation involves the long-run behavior of the system as the number of transitions tends to infinity. In such a case we need a systematic procedure that will predict the long-run behavior of the system.

Irreducible Markov Chain. A Markov chain is said to be *irreducible* if every state E_i can be reached from every other state E_j after a finite number of transitions—that is,

$$P_{ij}^{(n)} > 0, \, i \neq j, \, 1 \leq n < \infty$$

In this case all the states of the chain *communicate.*

Closed Set and Absorbing States. In a Markov process, a set C of states is said to be closed if the system, once in one of the states of C, will remain there indefinitely. A special example of a closed set is a single state E_j with transition probability $p_{jj} = 1$. In this case, E_j is called an **absorbing state**. All the states of an irreducible chain must form a closed set and no subset can be closed. The closed set C must also satisfy all the conditions of a Markov chain and hence may be studied independently.

Example 19.5-2

Consider the following Markov chain:

$$\mathbf{P} = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 2 & \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ 3 & 0 & 0 & 0 & 1 \end{array}$$

This chain is illustrated graphically in Figure 19.1. The figure shows that the four states do *not* constitute an irreducible chain, because states 0, 1, and 2 cannot be reached from state 3. State 3, by itself, forms a closed set, and hence it is absorbing. One can also say that state 3 forms an irreducible chain.

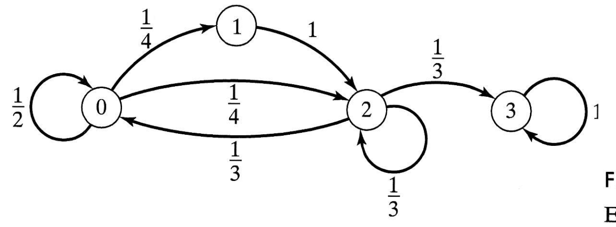


FIGURE 19.1
Example of the states of a Markov chain

First Return Times. An important definition in Markov chains theory is the **first return time**. Given that the system is initially in state E_j, it may return to E_j for *the first time* at the nth step, $n \geq 1$. The number of steps before the system returns to E_j is called the first return time.

Let $f_{ij}^{(n)}$ denote the probability that the first return to E_j occurs at the nth step. Then given the transition matrix

$$\mathbf{P} = \|p_{ij}\|$$

an expression for $f_{ij}^{(n)}$ can be determined as follows:

$$p_{jj} = f_{jj}^{(1)}$$
$$p_{jj}^{(2)} = f_{jj}^{(2)} + f_{jj}^{(1)} p_{jj}$$

or

$$f_{jj}^{(2)} = p_{jj}^{(2)} - f_{jj}^{(1)} p_{jj}$$

By induction

$$f_{jj}^{(n)} = p_{jj}^{(n)} - \sum_{m=1}^{n-1} f_{jj}^{(m)} p_{jj}^{(n-m)}$$

The probability of *at least* one return to state E_j is then given by

$$f_{jj} = \sum_{n=1}^{\infty} f_{jj}^{(n)}$$

Thus, the system is certain to return to j if $f_{jj} = 1$. In this case, if μ_{jj} defines the mean return (recurrence) time,

$$\mu_{jj} = \sum_{n=1}^{\infty} n f_{jj}^{(n)}$$

If $f_{jj} < 1$, it is not certain that the system will return to E_j and, consequently, $\mu_{jj} = \infty$.

The states of a Markov chain can be classified based on the definition of the first return times as follows:

1. A state is **transient** if $f_{jj} < 1$ — that is, $\mu_{jj} = \infty$.
2. A state is **recurrent** (persistent) if $f_{jj} = 1$.
3. A recurrent state is **null** if $\mu_{jj} = \infty$ and **nonnull** if $\mu_{jj} < \infty$.
4. A state is **periodic** with period t if a return is possible only in $t, 2t, 3t, \ldots$ steps. This means that $p_{jj}^{(n)} = 0$ whenever n is not divisible by t.
5. A recurrent state is **ergodic** if it is nonnull and aperiodic (not periodic).

If all the states of a Markov chain are ergodic, then the chain is *irreducible.* In this case, the absolute probabilities

$$\mathbf{a}^{(n)} = \mathbf{a}^{(0)}\mathbf{P}^n$$

always converge uniquely to a limiting distribution as $n \rightarrow \infty$, where the limiting distribution is independent of the initial probabilities $\mathbf{a}^{(0)}$.

The following theorem is now in order:

Theorem 19.5-1. *All the states in an irreducible infinite Markov chain may belong to one, and only one, of three states: transient, recurrent null, and recurrent nonnull. In every case all the states communicate, and they have the same period. For the special case where the chain has a finite number of states, the chain cannot consist of transient states only nor can it contain any null states.*

Limiting Distribution of Irreducible Chains. Example 19.5-1 shows that as the number of transitions increases, the absolute probability becomes independent of the initial distribution. This is the long-run property of Markov chains. In this section determination of the limiting (long-run) distribution of an irreducible chain is presented. The discussion will be restricted to the aperiodic type, because it is the only type needed in this text.

The existence of a limiting distribution in an irreducible aperiodic chain depends on the class of its states. Thus, considering the three classes given in Theorem 19.5-1, the following theorem can be stated:

Theorem 19.5-2. *In an irreducible aperiodic Markov chain,*

(a) *If all the states are transient or null, then $p_{ij}^{(n)} \rightarrow 0$ as $n \rightarrow \infty$ for all i and j and no limiting distribution exists.*

(b) *If all the states are ergodic, then*

$$\lim_{n\rightarrow\infty} a_j^{(n)} = \pi_j,\ j = 0, 1, 2, \ldots$$

where π_j is the limiting (steady-state) distribution. The probabilities π_j exist uniquely and are independent of $a_j^{(0)}$. In this case, π_j can be determined from the set of equations[2]

$$\pi_j = \sum_i \pi_i p_{ij}$$

$$1 = \sum_j \pi_j$$

The mean recurrence time for state j is then given by

$$\mu_{jj} = \frac{1}{\pi_j}$$

Example 19.5-3

To determine the steady-state probability distribution in Example 19.5-1, consider

$$\pi_1 = .2\pi_1 + .6\pi_2$$

$$\pi_2 = .8\pi_1 + .4\pi_2$$

$$\pi_1 + \pi_2 = 1$$

The solution yields $\pi_1 = .4286$ and $\pi_2 = .5714$. These results are very close to the row values of $\mathbf{a}^{(8)}$ in Example 19.5-1.

The mean recurrence time for states 1 and 2 are

$$\mu_{11} = \frac{1}{\pi_1} = 2.3$$

$$\mu_{22} = \frac{1}{\pi_2} = 1.75$$

Example 19.5-4

Consider the following Markov chain with three states:

$$\mathbf{P} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \end{array}$$

This is called a **doubly stochastic matrix**, because

$$\sum_{i=0}^{2} p_{ij} = \sum_{j=0}^{2} p_{ij} = 1$$

In such cases, the steady-state probabilities are

$$\pi_0 = \pi_1 = \pi_2 = \tfrac{1}{3}$$

[2] One of the equations $\pi_j = \sum_i \pi_i p_{ij}$ is redundant.

PROBLEM SET 19.5A

1. Classify the following Markov chains and find their stationary distributions.

(a) $$\begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{4} & \frac{3}{4} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$$

(b) $$\begin{pmatrix} q & p & 0 & 0 & 0 \\ q & 0 & p & 0 & 0 \\ q & 0 & 0 & p & 0 \\ q & 0 & 0 & 0 & p \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, p + q = 1$$

2. Find the mean recurrence time for each state of the following Markov chain:

$$\begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{5} & \frac{3}{5} & \frac{1}{5} \end{pmatrix}$$

SELECTED REFERENCES

Derman, C., *Finite State Markovian Decision Processes,* Academic Press, New York, 1970.

Howard, R., *Dynamic Programming and Markov Processes,* MIT Press, Cambridge, MA, 1960.

CHAPTER 20

Classical Optimization Theory

Classical optimization theory uses differential calculus to determine points of maxima and minima (extrema) for unconstrained and constrained functions. The methods may not be suitable for efficient numerical computations, but the underlying theory provides the basis for most nonlinear programming algorithms (see Chapter 21).

This chapter develops necessary and sufficient conditions for determining unconstrained extrema, the *Jacobian* and *Lagrangean* methods for problems with equality constraints, and the *Karush-Kuhn-Tucker* conditions for problems with inequality constraints.

20.1 UNCONSTRAINED PROBLEMS

An extreme point of a function $f(\mathbf{X})$ defines either a maximum or a minimum of the function. Mathematically, a point $\mathbf{X}_0 = (x_1^0, \ldots, x_j^0, \ldots, x_n^0)$ is a maximum if

$$f(\mathbf{X}_0 + \mathbf{h}) \leq f(\mathbf{X}_0)$$

for all $\mathbf{h} = (h_1, \ldots, h_j, \ldots, h_n)$ and $|h_j|$ is sufficiently small for all j. In other words, $\mathbf{X}_0$ is a maximum if the value of f at every point in the neighborhood of $\mathbf{X}_0$ does not exceed $f(\mathbf{X}_0)$. In a similar manner, $\mathbf{X}_0$ is a minimum if

$$f(\mathbf{X}_0 + \mathbf{h}) \geq f(\mathbf{X}_0)$$

Figure 20.1 illustrates the maxima and minima of a single-variable function $f(x)$ over the interval $[a,b]$. The points x_1, x_2, x_3, x_4, and x_6 are all extrema of $f(x)$, with x_1, x_3, and x_6 as maxima and x_2 and x_4 as minima. Because

$$f(x_6) = \max\{f(x_1), f(x_3), f(x_6)\}$$

$f(x_6)$ is a **global** or **absolute** maximum, and $f(x_1)$ and $f(x_3)$ are **local** or **relative** maxima. Similarly, $f(x_4)$ is a local minimum and $f(x_2)$ is a global minimum.

Although x_1 (in Figure 20.1) is a maximum point, it differs from remaining local maxima in that the value of f corresponding to at least one point in the neighborhood

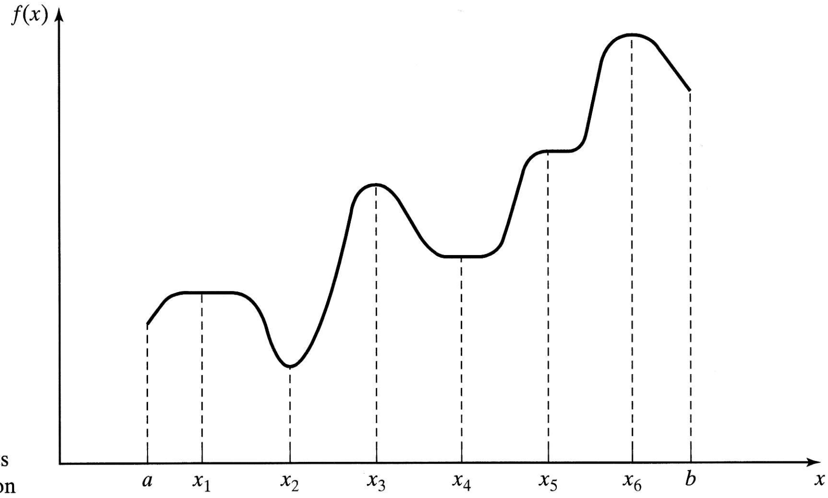


FIGURE 20.1
Examples of extreme points for a single-variable function

of x_1 is equal to $f(x_1)$. In this respect, x_1 is a **weak maximum**, whereas x_3 and x_6 are **strong maxima**. In general, $\mathbf{X}_0$ is a weak maximum if $f(\mathbf{X}_0 + \mathbf{h}) \leq f(\mathbf{X}_0)$ and a strong maximum if $f(\mathbf{X}_0 + \mathbf{h}) < f(\mathbf{X}_0)$, where $\mathbf{h}$ is as defined earlier.

In Figure 20.1, the first derivative (slope) of f equals zero at all extrema. This property is also satisfied at **inflection** and **saddle** points, such as x_5. If a point with zero slope (gradient) is not an extremum (maximum or minimum), then it must be an inflection or a saddle point.

20.1.1 Necessary and Sufficient Conditions

This section develops the necessary and sufficient conditions for an n-variable function $f(\mathbf{X})$ to have extrema. It is assumed that the first and second partial derivatives of $f(\mathbf{X})$ are continuous at every $\mathbf{X}$.

Theorem 20.1-1. *A necessary condition for $\mathbf{X}_0$ to be an extreme point of $f(\mathbf{X})$ is that*

$$\nabla f(\mathbf{X}_0) = \mathbf{0}$$

Proof. By Taylor's theorem, for $0 < \theta < 1$,

$$f(\mathbf{X}_0 + \mathbf{h}) - f(\mathbf{X}_0) = \nabla f(\mathbf{X}_0)\mathbf{h} + \tfrac{1}{2}\mathbf{h}^T\mathbf{H}\mathbf{h}|_{\mathbf{X}_0+\theta\mathbf{h}}$$

where $\mathbf{h}$ is as defined earlier. For sufficiently small $|h_j|$, the remainder term $\frac{1}{2}\mathbf{h}^T\mathbf{H}\mathbf{h}$ is of the order h_j^2 hence

$$f(\mathbf{X}_0 + \mathbf{h}) - f(\mathbf{X}_0) = \nabla f(\mathbf{X}_0)\mathbf{h} + 0(h_j^2) \approx \nabla f(\mathbf{X}_0)\mathbf{h}$$

It can be shown by contradiction that $\nabla f(\mathbf{X}_0)$ must vanish at a minimum point $\mathbf{X}_0$. Otherwise if it does not, then for a specific j the following condition will hold:

$$\frac{\partial f(\mathbf{X}_0)}{\partial x_j} < 0 \text{ or } \frac{\partial f(\mathbf{X}_0)}{\partial x_j} > 0$$

By selecting h_j with appropriate sign, it is always possible to have

$$h_j \frac{\partial f(\mathbf{X}_0)}{\partial x_j} < 0$$

Setting all other h_j equal to zero, Taylor's expansion yields

$$f(\mathbf{X}_0 + \mathbf{h}) < f(\mathbf{X}_0)$$

This result contradicts the assumption that $\mathbf{X}_0$ is a minimum point. Consequently, $\nabla f(\mathbf{X}_0)$ must equal zero. A similar proof can be established for the maximization case.

Because the necessary condition is also satisfied for inflection and saddle points, the points obtained from the solution of

$$\nabla f(\mathbf{X}_0) = \mathbf{0}$$

are referred to as **stationary** points. The next theorem establishes the sufficiency conditions for $\mathbf{X}_0$ to be an extreme point.

Theorem 20.1-2. *A sufficient condition for a stationary point* $\mathbf{X}_0$ *to be an extremum is that the Hessian matrix* **H** *evaluated at* $\mathbf{X}_0$ *satisfy the following conditions*:

1. **H** *is positive definite if* $\mathbf{X}_0$ *is a minimum point.*
2. **H** *is negative definite if* $\mathbf{X}_0$ *is a maximum point.*

Proof. By Taylor's theorem, for $0 < \theta < 1$,

$$f(\mathbf{X}_0 + \mathbf{h}) - f(\mathbf{X}_0) = \nabla f(\mathbf{X}_0)\mathbf{h} + \tfrac{1}{2}\mathbf{h}^T\mathbf{H}\mathbf{h}|_{\mathbf{X}_0+\theta\mathbf{h}}$$

Given $\mathbf{X}_0$ is a stationary point, then $\nabla f(\mathbf{X}_0) = \mathbf{0}$ (Theorem 20.2-1). Thus,

$$f(\mathbf{X}_0 + \mathbf{h}) - f(\mathbf{X}_0) = \tfrac{1}{2}\mathbf{h}^T\mathbf{H}\mathbf{h}|_{\mathbf{X}_0+\theta\mathbf{h}}$$

If $\mathbf{X}_0$ is a minimum point, then

$$f(\mathbf{X}_0 + \mathbf{h}) > f(\mathbf{X}_0),\ \mathbf{h} \neq \mathbf{0}$$

Thus, for $\mathbf{X}_0$ to be a minimum point, it must be true that

$$\tfrac{1}{2}\mathbf{h}^T\mathbf{H}\mathbf{h}|_{\mathbf{X}_0+\theta\mathbf{h}} > 0$$

Given that the second partial derivative is continuous, the expression $\frac{1}{2}\mathbf{h}^T\mathbf{H}\mathbf{h}$ must have the same sign at both $\mathbf{X}_0$ and $\mathbf{X}_0 + \theta\mathbf{h}$. Because $\mathbf{h}^T\mathbf{H}\mathbf{h}|_{\mathbf{X}_0}$ defines a quadratic form (see Section A.3), this expression (and hence $\mathbf{h}^T\mathbf{H}\mathbf{h}|_{\mathbf{X}_0+\theta\mathbf{h}}$) is positive if, and only if, $\mathbf{H}|_{\mathbf{X}_0}$ is positive-definite. This means that a sufficient condition for the stationary point $\mathbf{X}_0$ to be a minimum is that the Hessian matrix, **H,** evaluated at the same point is positive-definite. A similar proof for the maximization case shows that the corresponding Hessian matrix must be negative-definite.

Example 20.1-1

Consider the function

$$f(x_1, x_2, x_3) = x_1 + 2x_3 + x_2x_3 - x_1^2 - x_2^2 - x_3^2$$

The necessary condition

$$\nabla f(\mathbf{X}_0) = 0$$

gives

$$\frac{\partial f}{\partial x_1} = 1 - 2x_1 = 0$$

$$\frac{\partial f}{\partial x_2} = x_3 - 2x_2 = 0$$

$$\frac{\partial f}{\partial x_3} = 2 + x_2 - 2x_3 = 0$$

The solution of these simultaneous equations is given by

$$\mathbf{X}_0 = (\tfrac{1}{2}, \tfrac{2}{3}, \tfrac{4}{3})$$

To establish sufficiency, consider

$$\mathbf{H}|_{\mathbf{x}_0} = \begin{pmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_3} \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \dfrac{\partial^2 f}{\partial x_2 \partial x_3} \\ \dfrac{\partial^2 f}{\partial x_3 \partial x_1} & \dfrac{\partial^2 f}{\partial x_3 \partial x_2} & \dfrac{\partial^2 f}{\partial x_3^2} \end{pmatrix}_{\mathbf{x}_0} = \begin{pmatrix} -2 & 0 & 0 \\ 0 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix}$$

The principal minor determinants of $\mathbf{H}|_{\mathbf{x}_0}$ have the values -2, 4, and -6, respectively. Thus, as shown in Section A.3, $\mathbf{H}|_{\mathbf{x}_0}$ is negative-definite and $\mathbf{X}_0 = (\frac{1}{2}, \frac{2}{3}, \frac{4}{3})$ represents a maximum point.

In general, if $\mathbf{H}|_{\mathbf{x}_0}$ is indefinite, $\mathbf{X}_0$ must be a saddle point. For nonconclusive cases, $\mathbf{X}_0$ may or may not be an extremum and the sufficiency condition becomes rather involved because higher-order terms in Taylor's expansion must be considered.

The sufficiency condition established by Theorem 20.1-2 applies to single-variable functions as follows. Given y_0 is a stationary point, then

1. y_0 is a maximum if $f'(y_0) < 0$.
2. y_0 is a minimum if $f'(y_0) > 0$.

If in the single-variable case $f''(y_0) = 0$, higher-order derivatives must be investigated as the following theorem requires.

Theorem 20.1-3. *Given y_0, a stationary point of $f(y)$, if the first $(n - 1)$ derivatives are zero and $f^{(n)}(y_0) \neq 0$, then*

1. *y_0 is an inflection point if n is odd.*
2. *y_0 is a minimum if n is even and $f^{(n)}(y_0) > 0$.*
3. *y_0 is a maximum if n is even and $f^{(n)}(y_0) < 0$.*

Example 20.1-2

Figure 20.2 graphs the following two functions:

$$f(y) = y^4$$
$$g(y) = y^3$$

For $f(y) = y^4$

$$f'(y) = 4y^3 = 0$$

which yields the stationary point $y_0 = 0$. Now

$$f'(0) = f''(0) = f^{(3)}(0) = 0,\ f^{(4)}(0) = 24 > 0$$

Hence, $y_0 = 0$ is a minimum point (see Figure 20.2).

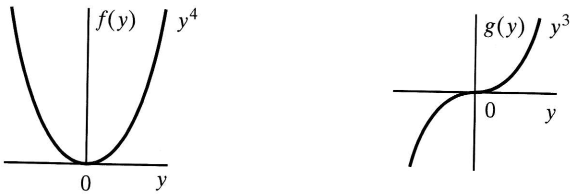


FIGURE 20.2

Extreme points of $f(y) = y^4$ and $g(y) = y^3$

For $g(y) = y^3$,

$$g'(y) = 3y^2 = 0$$

This yields $y_0 = 0$ as a stationary point. Also

$$g'(0) = g''(0),\ g^{(3)}(0) = 6 \neq 0$$

Thus, $y_0 = 0$ is an inflection point.

PROBLEM SET 20.1A

1. Examine the following functions for extreme points.

(a) $f(x) = x^3 + x$

(b) $f(x) = x^4 + x^2$

(c) $f(x) = 4x^4 - x^2 + 5$

(d) $f(x) = (3x - 2)^2(2x - 3)^2$

(e) $f(x) = 6x^5 - 4x^3 + 10$

2. Examine the following functions for extreme points.

(a) $f(\mathbf{X}) = x_1^3 + x_2^3 - 3x_1x_2$

(b) $f(\mathbf{X}) = 2x_1^2 + x_2^2 + x_3^2 + 6(x_1 + x_2 + x_3) + 2x_1x_2x_3$

3. Verify that the function

$$f(x_1, x_2, x_3) = 2x_1x_2x_3 - 4x_1x_3 - 2x_2x_3 + x_1^2 + x_2^2 + x_3^2 - 2x_1 - 4x_2 + 4x_3$$

has the stationary points $(0, 3, 1)$, $(0, 1, -1)$, $(1, 2, 0)$, $(2, 1, 1)$, and $(2, 3, -1)$. Use the sufficiency condition to find the extreme points.

4. Solve the following simultaneous equations by converting the system to a nonlinear objective function with no constraints.

$$x_2 - x_1^2 = 0$$
$$x_2 - x_1 = 2$$

(*Hint:* min $f^2(x_1, x_2)$ occurs at $f(x_1, x_2) = 0$.)

5. Prove Theorem 20.1-3.

20.1.2 The Newton-Raphson Method

In general, the necessary condition equations, $\nabla f(\mathbf{X}) = \mathbf{0}$, may be difficult to solve numerically. The Newton-Raphson method is an iterative procedure for solving simultaneous nonlinear equations. Although the method is presented here in this context, it is actually part of the **gradient methods** for optimizing unconstrained functions numerically (see Section 21.1.2).

Consider the simultaneous equations

$$f_i(\mathbf{X}) = 0, \quad i = 1, 2, \ldots, m$$

Let $\mathbf{X}^k$ be a given point. Then by Taylor's expansion

$$f_i(\mathbf{X}) \approx f_i(\mathbf{X}^k) + \nabla f_i(\mathbf{X}^k)(\mathbf{X} - \mathbf{X}^k), \quad i = 1, 2, \ldots, m$$

Thus, the original equations, $f_i(\mathbf{X}) = 0, \quad i = 1, 2, \ldots, m$, may be approximated as

$$f_i(\mathbf{X}^k) + \nabla f_i(\mathbf{X}^k)(\mathbf{X} - \mathbf{X}^k) = 0, \; i = 1, 2, \ldots, m$$

These equations may be written in matrix notation as

$$\mathbf{A}_k + \mathbf{B}_k(\mathbf{X} - \mathbf{X}^k) = \mathbf{0}$$

If $\mathbf{B}_k$ is nonsingular, then

$$\mathbf{X} = \mathbf{X}^k - \mathbf{B}_k^{-1}\mathbf{A}_k$$

The idea of the method is to start from an initial point $\mathbf{X}^0$. By using the foregoing equation, a new point $\mathbf{X}^{k+1}$ is determined from $\mathbf{X}^k$. The procedure ends with $\mathbf{X}^m$ as the solution when $\mathbf{X}^m \approx \mathbf{X}^{m-1}$.

A geometric interpretation of the method is illustrated by a single-variable function in Figure 20.3. The relationship between x^k and x^{k+1} for a single-variable function $f(x)$ reduces to

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}$$

or

$$f'(x^k) = \frac{f(x^k)}{x^k - x^{k+1}}$$

The figure shows that x^{k+1} is determined from the slope of $f(x)$ at x^k, where $\tan \theta = f'(x^k)$.

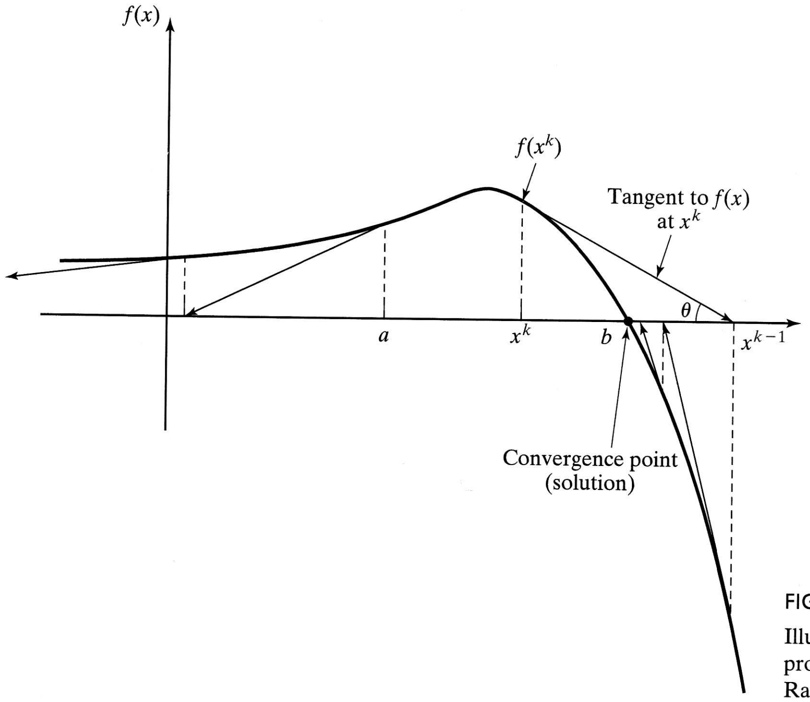


FIGURE 20.3

Illustration of the iterative process in the Newton-Raphson Method

One difficulty with the method is that convergence is not always guaranteed unless the function f is well behaved. In Figure 20.3, if the initial point is a, the method will diverge. There is no easy way for locating a "good" initial point.

Example 20.1-3

To demonstrate the use of the Newton-Raphson method, consider determining the stationary points of the function

$$f(x) = (3x - 2)^2(2x - 3)^2$$

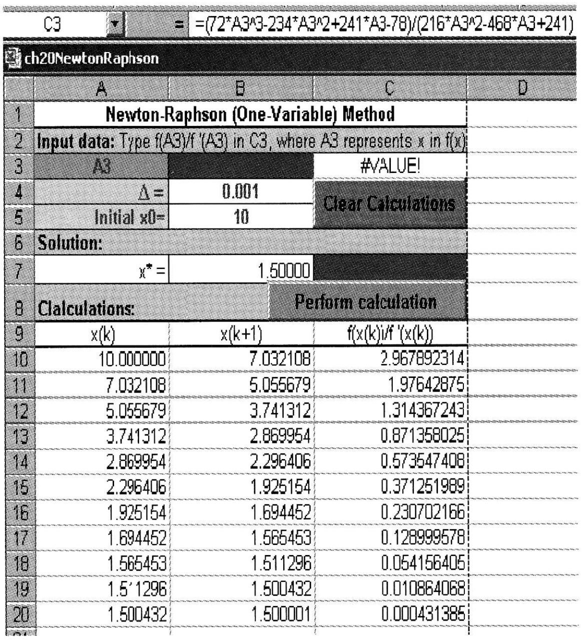
The equation we need to solve to determine the stationary points is $f'(x) = 0$, which gives

$$72x^3 - 234x^2 + 241x - 78 = 0$$

Excel template ch20NewtonRaphson.xls can be used to solve any single-variable equation. Figure 20.4 provides the iterations for solving $f'(x) = 0$. The template requires entering the following ratio in cell C3, with the variable x replaced with A3:

$$\frac{72x^3 - 234x^2 + 241x - 78}{216x^2 - 468x + 241}$$

Note that the denominator is the first derivative of the numerator, as required by the Newton-Raphson method. We set tolerance limit $\Delta = .001$ and select initial starting point $x_0 = 10$. The tolerance limit specifies the allowable difference between x^k and x^{k+1} that signals the termination of the iterations. The method converges to $x = 1.5$.



C3 = =(72*A3^3-234*A3^2+241*A3-78)/(216*A3^2-468*A3+241)

ch20NewtonRaphson

	A	B	C	D
1	Newton-Raphson (One-Variable) Method			
2	**Input data:** Type f(A3)/f '(A3) in C3, where A3 represents x in f(x)			
3	A3		#VALUE!	
4	Δ=	0.001	Clear Calculations	
5	Initial x0=	10		
6	**Solution:**			
7	x* =	1.50000		
8	**Clalculations:**		Perform calculation	
9	x(k)	x(k+1)	f(x(k))/f '(x(k))	
10	10.000000	7.032108	2.967892314	
11	7.032108	5.055679	1.97642875	
12	5.055679	3.741312	1.314367243	
13	3.741312	2.869954	0.871358025	
14	2.869954	2.296406	0.573547408	
15	2.296406	1.925154	0.371251989	
16	1.925154	1.694452	0.230702166	
17	1.694452	1.565453	0.128999578	
18	1.565453	1.511296	0.054156405	
19	1.5'1296	1.500432	0.010864068	
20	1.500432	1.500001	0.000431385	

FIGURE 20.4

Excel solution of $72x^3 - 234x^2 + 241x - 78 = 0$ by the Newton-Raphson Method

Actually, $f(x)$ has three stationary points at $x = \frac{2}{3}$, $x = \frac{13}{12}$, and $x = \frac{3}{2}$. The remaining two points can be found by selecting different values for initial x_0. In fact, $x_0 = .5$ and $x_0 = 1$ should yield the missing stationary points. You are encouraged to use different initial x_0 to get a feel of how the method works.

In general, the Newton-Raphson method requires making several attempts before "all" the solutions can be found. In the present example, we know beforehand that the equation has three roots. This will not be the case with complex or multivariable functions, however.

PROBLEM SET 20.1B

1. Use ch20NewtonRaphson.xls to solve Problem 1(c), Set 20.1a.
2. Solve Problem 2(b), Set 20.1a by the Newton-Raphson method.

20.2 CONSTRAINED PROBLEMS

This section deals with the optimization of constrained continuous functions. Section 20.2.1 introduces the case of equality constraints, and Section 20.2.2 deals with inequality constraints. The presentation in Section 20.2.1 is covered for the most part in Beightler and associates (1979, pp. 45–55).

20.2.1 Equality Constraints

This section presents two methods: The **Jacobian** and the **Lagrangean.** The Lagrangean method can be developed logically from the Jacobian method. This relationship provides an interesting economic interpretation of the Lagrangean method.

Constrained Derivatives (Jacobian) Method. Consider the problem

$$\text{Minimize } z = f(\mathbf{X})$$

subject to

$$\mathbf{g}(\mathbf{X}) = \mathbf{0}$$

where

$$\mathbf{X} = (x_1, x_2, \ldots, x_n)$$

$$\mathbf{g} = (g_1, g_2, \ldots, g_m)^T$$

The functions $f(\mathbf{X})$ and $g_i(\mathbf{X})$, $i = 1, 2, \ldots, m$, are twice continuously differentiable.

The idea of using constrained derivatives is to develop a closed-form expression for the first partial derivatives of $f(\mathbf{X})$ at all points that satisfy the constraints $\mathbf{g}(\mathbf{X}) = \mathbf{0}$. The corresponding stationary points are identified as the points at which these partial derivatives vanish. The sufficiency conditions introduced in Section 20.1 can then be used to check the identity of stationary points.

To clarify the proposed concept, consider $f(x_1, x_2)$ illustrated in Figure 20.5. This function is to be minimized subject to the constraint

$$g_1(x_1, x_2) = x_2 - b = 0$$

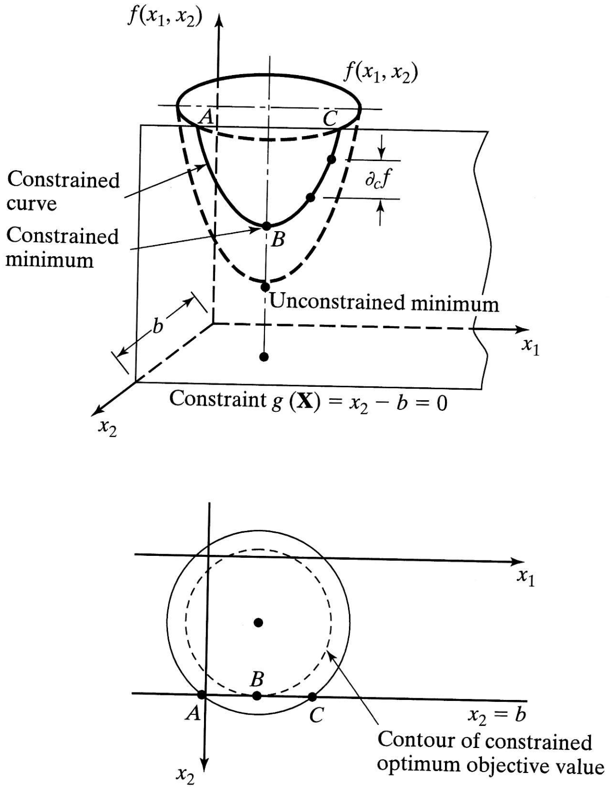


FIGURE 20.5

Demonstration of the idea of the Jacobian Method where b is a constant. From Figure 20.5, the curve designated by the three points A, B, and C represents the values of $f(x_1, x_2)$ for which the constraint is always satisfied. The constrained derivatives method defines the gradient of $f(x_1, x_2)$ at any point on the curve ABC. Point B at which the constrained derivative vanishes is a stationary point for the constrained problem.

The method is now developed mathematically. By Taylor's theorem, for $\mathbf{X} + \Delta\mathbf{X}$ in the feasible neighborhood of $\mathbf{X}$, we have

$$f(\mathbf{X} + \Delta\mathbf{X}) - f(\mathbf{X}) = \nabla f(\mathbf{X})\Delta\mathbf{X} + O(\Delta x_j^2)$$

and

$$\mathbf{g}(\mathbf{X} + \Delta\mathbf{X}) - \mathbf{g}(\mathbf{X}) = \nabla\mathbf{g}(\mathbf{X})\Delta\mathbf{X} + O(\Delta\mathbf{x}_j^2)$$

As $\Delta x_j \rightarrow 0$, the equations reduce to

$$\partial f(\mathbf{X}) = \nabla f(\mathbf{X})\partial\mathbf{X}$$

and

$$\partial\mathbf{g}(\mathbf{X}) = \nabla\mathbf{g}(\mathbf{X})\partial\mathbf{X}$$

For feasibility, we must have $\mathbf{g}(\mathbf{X}) = \mathbf{0}$, $\partial\mathbf{g}(\mathbf{X}) = \mathbf{0}$, and it follows that

$$\partial f(\mathbf{X}) - \nabla f(\mathbf{X})\,\partial\mathbf{X} = \mathbf{0}$$
$$\nabla\mathbf{g}(\mathbf{X})\partial\mathbf{X} = \mathbf{0}$$

This gives $(m + 1)$ equations in $(n + 1)$ unknowns, $\partial f(\mathbf{X})$ and $\partial\mathbf{X}$. Note that $\partial f(\mathbf{X})$ is a dependent variable and, hence, is determined once $\partial\mathbf{X}$ is known. This means that we have m equations in n unknowns.

If $m > n$, at least $(m - n)$ equations are redundant. Eliminating redundancy, the system reduces to $m \leq n$. If $m = n$, the solution is $\partial\mathbf{X} = \mathbf{0}$, and $\mathbf{X}$ has no feasible neighborhood, which means that the solution space consists of one point only. The remaining case, where $m < n$, requires further elaboration.

Define

$$\mathbf{X} = (\mathbf{Y}, \mathbf{Z})$$

such that

$$\mathbf{Y} = (y_1, y_2, \ldots, y_m),\ \mathbf{Z} = (z_1, z_2, \ldots, z_{n-m})$$

The vectors $\mathbf{Y}$ and $\mathbf{Z}$ are called the *dependent* and *independent* variables, respectively. Rewriting the gradient vectors of f and g in terms of $\mathbf{Y}$ and $\mathbf{Z}$, we get

$$\nabla f(\mathbf{Y}, \mathbf{Z}) = (\nabla_{\mathbf{Y}} f, \nabla_{\mathbf{Z}} f)$$
$$\nabla\mathbf{g}(\mathbf{Y}, \mathbf{Z}) = (\nabla_{\mathbf{Y}}\mathbf{g}, \nabla_{\mathbf{Z}}\mathbf{g})$$

Define

$$\mathbf{J} = \nabla_{\mathbf{Y}}\mathbf{g} = \begin{pmatrix} \nabla_{\mathbf{Y}} g_1 \\ \vdots \\ \nabla_{\mathbf{Y}} g_m \end{pmatrix}$$

$$\mathbf{C} = \nabla_{\mathbf{z}}\mathbf{g} = \begin{pmatrix} \nabla_{\mathbf{z}}g_1 \\ \vdots \\ \nabla_{\mathbf{z}}g_m \end{pmatrix}$$

$\mathbf{J}_{m\times m}$ is called the **Jacobian matrix** and $\mathbf{C}_{m\times n-m}$ the **control matrix**. The Jacobian **J** is assumed nonsingular. This is always possible because the given m equations are independent by definition. The components of the vector **Y** must thus be selected such that the matrix **J** is nonsingular.

The original set of equations in $\partial f(\mathbf{X})$ and $\partial\,\mathbf{X}$ may be written as

$$\partial f(\mathbf{Y},\mathbf{Z}) = \nabla_{\mathbf{Y}} f \partial\mathbf{Y} + \nabla_{\mathbf{z}} f \partial\mathbf{Z}$$

and

$$\mathbf{J}\partial\mathbf{Y} = -\mathbf{C}\,\partial\,\mathbf{Z}$$

Because **J** is nonsingular, its inverse $\mathbf{J}^{-1}$ exists. Hence,

$$\partial\,\mathbf{Y} = -\,\mathbf{J}^{-1}\mathbf{C}\partial\,\mathbf{Z}$$

Substituting for $\partial\,\mathbf{Y}$ in the equation for $\partial\,f(\mathbf{X})$ gives ∂f as a function of $\partial\,\mathbf{Z}$—that is,

$$\partial\,f(\mathbf{Y},\mathbf{Z}) = (\nabla_{\mathbf{z}} f - \nabla_{\mathbf{Y}} f\mathbf{J}^{-1}\mathbf{C})\partial\mathbf{Z}$$

From this equation, the constrained derivative with respect to the independent vector **Z** is given by

$$\nabla_c f = \frac{\partial_c f(\mathbf{Y},\mathbf{Z})}{\partial_c\mathbf{Z}} = \nabla_z f - \nabla_{\mathbf{Y}} f\mathbf{J}^{-1}\mathbf{C}$$

where $\nabla_c f$ is the **constrained gradient** vector of f with respect to **Z**. Thus, $\nabla_c f(\mathbf{Y},\mathbf{Z})$ must be null at the stationary points.

The sufficiency conditions are similar to those developed in Section 20.1. The Hessian matrix will correspond to the independent vector **Z**, and the elements of the Hessian matrix must be the *constrained* second derivatives. To show how this is obtained, let

$$\nabla_c f = \nabla_z f - \mathbf{WC}$$

It thus follows the ith row of the (constrained) Hessian matrix is $\partial\,\nabla_c f/\partial\,z_i$. Notice that **W** is a function of **Y** and **Y** is a function of **Z**. Thus, the partial derivative of $\nabla_c f$ with respect to z_i is based on the following chain rule:

$$\frac{\partial\,w_j}{\partial\,z_i} = \frac{\partial\,w_j}{\partial\,y_j}\frac{\partial\,y_j}{\partial\,z_i}$$

Example 20.2-1

Consider the following problem:

$$f(\mathbf{X}) = x_1^2 + 3x_2^2 + 5x_1x_3^2$$

$$g_1(\mathbf{X}) = x_1x_3 + 2x_2 + x_2^2 - 11 = 0$$

$$g_2(\mathbf{X}) = x_1^2 + 2x_1x_2 + x_3^2 - 14 = 0$$

Given the feasible point $\mathbf{X}^0 = (1, 2, 3)$, we wish to study the variation in $f(=\partial_c f)$ in the feasible neighborhood of $\mathbf{X}^0$.

Let

$$\mathbf{Y} = (x_1, x_3) \quad \text{and} \quad \mathbf{Z} = x_2$$

Thus,

$$\nabla_{\mathbf{Y}} f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_3}\right) = (2x_1 + 5x_3^2, 10x_1x_3)$$

$$\nabla_{\mathbf{Z}} f = \frac{\partial f}{\partial x_2} = 6x_2$$

$$\mathbf{J} = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_3} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_3} \end{pmatrix} = \begin{pmatrix} x_3 & x_1 \\ 2x_1 + 2x_2 & 2x_3 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} \frac{\partial g_1}{\partial x_2} \\ \frac{\partial g_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2x_2 + 2 \\ 2x_1 \end{pmatrix}$$

Suppose that we need to estimate $\partial_c f$ in the feasible neighborhood of the feasible point $\mathbf{X}^0 = (1, 2, 3)$ given a small change $\partial x_2 = .01$ in the independent variable x_2. We have

$$\mathbf{J}^{-1}\mathbf{C} = \begin{pmatrix} 3 & 1 \\ 6 & 6 \end{pmatrix}^{-1} \begin{pmatrix} 6 \\ 2 \end{pmatrix} = \begin{pmatrix} \frac{6}{12} & -\frac{1}{12} \\ -\frac{6}{12} & \frac{3}{12} \end{pmatrix} \begin{pmatrix} 6 \\ 2 \end{pmatrix} \approx \begin{pmatrix} 2.83 \\ -2.50 \end{pmatrix}$$

Hence, the incremental value of constrained f is given as

$$\partial_c f = (\nabla_{\mathbf{Z}} f - \nabla_{\mathbf{Y}} f \mathbf{J}^{-1}\mathbf{C})\partial Z = \left(6(2) - (47, 30)\begin{pmatrix} 2.83 \\ -2.50 \end{pmatrix}\right)\partial x_2 = -46.01\,\partial x_2$$

By specifying the value of ∂x_2 for the *independent* variable x_2, feasible values of ∂x_1 and ∂x_2 are determined for the dependent variables x_1 and x_3 using the formula

$$\partial \mathbf{Y} = -\mathbf{J}^{-1}\mathbf{C}\,\partial \mathbf{Z}$$

Thus, for $\partial x_2 = .01$,

$$\begin{pmatrix} \partial x_1 \\ \partial x_3 \end{pmatrix} = -\mathbf{J}^{-1}\mathbf{C}\,\partial x_2 = \begin{pmatrix} -.0283 \\ .0250 \end{pmatrix}$$

We now compare the value of $\partial_c f$ as computed above with the difference $f(\mathbf{X}^0 + \partial\mathbf{X}) - f(\mathbf{X}^0)$, given $\partial x_2 = .01$.

$$\mathbf{X}^0 + \partial\mathbf{X} = (1 - .0283, 2 + .01, 3 + .025) = (.9717, 2.01, 3.025)$$

This yields

$$f(\mathbf{X}^0) = 58, f(\mathbf{X}^0 + \partial\mathbf{X}) = 57.523$$

or

$$f(\mathbf{X}^0 + \partial\mathbf{X}) - f(\mathbf{X}^0) = -.477$$

The amount $-.477$ compares favorably with $\partial_c f = -46.01\,\partial x_2 = -.4601$. The difference between the two values is the result of the linear approximation in computing $\partial_c f$ at $\mathbf{X}^0$.

PROBLEM SET 20.2A

1. Consider Example 20.2-1.

(a) Compute $\partial_c f$ by the two methods presented in the example, using $\partial x_2 = .001$ instead of $\partial x_2 = .01$. Does the effect of linear approximation become more negligible with the decrease in the value of ∂x_2?

(b) Specify a relationship among ∂x_1, ∂x_2, and ∂x_3 at the feasible point $\mathbf{X}^0 = (1, 2, 3,)$ that will keep the point $(x_1^0 + \partial x_1, x_2^0 + \partial x_2, x_3^0 + \partial x_3)$ feasible.

(c) If $\mathbf{Y} = (x_2, x_3)$ and $\mathbf{Z} = x_1$, what is the value of ∂x_1 that will produce the same value of $\partial_c f$ given in the example?

Example 20.2-2

This example illustrates the use of constrained derivatives. Consider the problem

$$\text{Minimize } f(\mathbf{X}) = x_1^2 + x_2^2 + x_3^2$$

subject to

$$g_1(\mathbf{X}) = x_1 + x_2 + 3x_3 - 2 = 0$$

$$g_2(\mathbf{X}) = 5x_1 + 2x_2 + x_3 - 5 = 0$$

To determine the constrained extreme points, let

$$\mathbf{Y} = (x_1, x_2) \text{ and } \mathbf{Z} = x_3$$

Thus,

$$\nabla_{\mathbf{Y}} f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right) = (2x_1, 2x_2), \; \nabla_{\mathbf{Z}} f = \frac{\partial f}{\partial x_3} = 2x_3$$

$$\mathbf{J} = \begin{pmatrix} 1 & 1 \\ 5 & 2 \end{pmatrix}, \; \mathbf{J}^{-1} = \begin{pmatrix} -\frac{2}{3} & \frac{1}{3} \\ \frac{5}{3} & -\frac{1}{3} \end{pmatrix}, \; \mathbf{C} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

Hence,

$$\nabla_c f = \frac{\partial_c f}{\partial_c x_3} = 2x_3 - (2x_1, 2x_2)\begin{pmatrix} -\frac{2}{3} & \frac{1}{3} \\ \frac{5}{3} & -\frac{1}{3} \end{pmatrix}\begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

$$= \tfrac{10}{3} x_1 - \tfrac{28}{3} x_2 + 2x_3$$

The equations for determining the stationary points are thus given as

$$\nabla_c f = 0$$

$$g_1(\mathbf{X}) = 0$$

$$g_2(\mathbf{X}) = 0$$

or

$$\begin{pmatrix} 10 & -28 & 6 \\ 1 & 1 & 3 \\ 5 & 2 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 5 \end{pmatrix}$$

The solution is

$$\mathbf{X}^0 \approx (.81, .35, .28)$$

The identity of $\mathbf{X}^0$ is checked using the sufficiency condition. Given x_3 is the independent variable, it follows from $\nabla_c f$ that

$$\frac{\partial_c^2 f}{\partial_c x_3^2} = \tfrac{10}{3}\left(\frac{dx_1}{dx_3}\right) - \tfrac{28}{3}\left(\frac{dx_2}{dx_3}\right) + 2 = \left(\tfrac{10}{3}, -\tfrac{28}{3}\right)\begin{pmatrix} \dfrac{dx_1}{dx_3} \\ \dfrac{dx_2}{dx_3} \end{pmatrix} + 2$$

From the Jacobian method,

$$\begin{pmatrix} \dfrac{dx_1}{dx_3} \\ \dfrac{dx_2}{dx_3} \end{pmatrix} = -\mathbf{J}^{-1}\mathbf{C} = \begin{pmatrix} \frac{5}{3} \\ -\frac{14}{3} \end{pmatrix}$$

Substitution gives $\partial_c^2 f / \partial_c x_3^2 = \frac{460}{9} > 0$. Hence, $\mathbf{X}^0$ is the minimum point.

Sensitivity Analysis in the Jacobian Method. The Jacobian method can be used to study the effect of small changes in the right-hand side of the constraints on the optimal value of f. Specifically, what is the effect of changing $g_i(\mathbf{X}) = 0$ to $g_i(\mathbf{X}) = \partial g_i$ on the optimal value of f? This type of investigation is called **sensitivity analysis** and is similar to that carried out in linear programming (see Chapter 4). However, sensitivity analysis in nonlinear programming is valid only in the immediate neighborhood of the extreme point. The development will be helpful in studying the Lagrangean method.

We have shown previously that

$$\partial f(\mathbf{Y}, \mathbf{Z}) = \nabla_{\mathbf{Y}} f \partial \mathbf{Y} + \nabla_{\mathbf{Z}} f \partial \mathbf{Z}$$
$$\partial \mathbf{g} = \mathbf{J} \partial \mathbf{Y} + \mathbf{C} \partial \mathbf{Z}$$

Given $\partial \mathbf{g} \neq \mathbf{0}$, then

$$\partial \mathbf{Y} = \mathbf{J}^{-1} \partial \mathbf{g} - \mathbf{J}^{-1}\mathbf{C} \partial \mathbf{Z}$$

Substituting in the equation for $\partial f(\mathbf{Y}, \mathbf{Z})$ gives

$$\partial f(\mathbf{Y}, \mathbf{Z}) = \nabla_{\mathbf{Y}} f \mathbf{J}^{-1} \partial \mathbf{g} + \nabla_c f \partial \mathbf{Z}$$

where

$$\nabla_c f = \nabla_{\mathbf{Z}} f - \nabla_{\mathbf{Y}} f \mathbf{J}^{-1}\mathbf{C}$$

as defined previously. The expression for $\partial f(\mathbf{Y}, \mathbf{Z})$ can be used to study variation in f in the feasible neighborhood of a feasible point $\mathbf{X}^0$ resulting from making small changes $\partial \mathbf{g}$ and $\partial \mathbf{Z}$.

At the extreme (indeed, any stationary) point $\mathbf{X}_0 = (\mathbf{Y}_0, \mathbf{Z}_0)$ the constrained gradient $\nabla_c f$ must vanish. Thus

$$\partial f(\mathbf{Y}_0, \mathbf{Z}_0) = \nabla_{\mathbf{Y}_0} f \mathbf{J}^{-1} \partial \mathbf{g}(\mathbf{Y}_0, \mathbf{Z}_0)$$

or

$$\frac{\partial f}{\partial \mathbf{g}} = \nabla_{Y_0} f \mathbf{J}^{-1}$$

The effect of the small change $\partial \mathbf{g}$ on the *optimum* value of f can be studied by evaluating the rate of change of f with respect to $\mathbf{g}$. These rates are usually referred to as **sensitivity coefficients**.

Example 20.2-3

Consider the same problem of Example 20.2-2. The optimum point is given by $\mathbf{X}_0 = (x_1^0, x_2^0, x_3^0) = (.81, .35, .28)$. Given $\mathbf{Y}_0 = (x_1^0, x_2^0)$, then

$$\nabla_{\mathbf{Y}_0} f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right) = (2x_1^0, 2x_2^0) = (1.62, .70)$$

Consequently,

$$\left(\frac{\partial f}{\partial g_1}, \frac{\partial f}{\partial g_2}\right) = \nabla_{\mathbf{Y}_0} f\mathbf{J}^{-1} = (1.62, .7)\begin{pmatrix} -\frac{2}{3} & \frac{1}{3} \\ \frac{5}{3} & -\frac{1}{3} \end{pmatrix} = (.0876, .3067)$$

This means that for $\partial g_2 = 1$, f will increase *approximately* by .0867. Similarly, for $\partial g_1 = 1$, f will increase *approximately* by .3067.

Application of the Jacobian Method to an LP Problem. Consider the linear programming problem

$$\text{Maximize } z = 2x_1 + 3x_2$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 \quad &= 5 \\ x_1 - x_2 \quad + x_4 &= 3 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

To account for the nonnegativity constraints $x_j \geq 0$, substitute $x_j = w_j^2$. With this substitution, the nonnegativity conditions become implicit and the original problem becomes

$$\text{Maximize } z = 2w_1^2 + 3w_2^2$$

subject to

$$\begin{aligned} w_1^2 + w_2^2 + w_3^2 &= 5 \\ w_1^2 - w_2^2 + w_4^2 &= 3 \end{aligned}$$

To apply the Jacobian method, let

$$\mathbf{Y} = (w_1, w_2), \mathbf{Z} = (w_3, w_4)$$

(In the terminology of linear programming, $\mathbf{Y}$ and $\mathbf{Z}$ correspond to the basic and nonbasic variables, respectively.) Thus

$$\mathbf{J} = \begin{pmatrix} 2w_1 & 2w_2 \\ 2w_1 & -2w_2 \end{pmatrix}, \mathbf{J}^{-1} = \begin{pmatrix} \frac{1}{4w_1} & \frac{1}{4w_1} \\ \frac{1}{4w_2} & \frac{-1}{4w_2} \end{pmatrix}, w_1 \text{ and } w_2 \neq 0$$

$$\mathbf{C} = \begin{pmatrix} 2w_3 & 0 \\ 0 & 2w_4 \end{pmatrix}, \nabla_Y f = (4w_1, 6w_2), \nabla_Z f = (0, 0)$$

so that

$$\nabla_c f = (0,\ 0) - (4w_1,\ 6w_2)\begin{pmatrix} \frac{1}{4w_1} & \frac{1}{4w_1} \\ \frac{1}{4w_2} & \frac{-1}{4w_2} \end{pmatrix}\begin{pmatrix} 2w_3 & 0 \\ 0 & 2w_4 \end{pmatrix} = (-5w_3,\ w_4)$$

The solution of the equations comprised of $\nabla_c f = \mathbf{0}$ and the constraints of the problem yield the stationary point ($w_1 = 2,\ w_2 = 1,\ w_3 = 0,\ w_4 = 0$). The Hessian is given by

$$\mathbf{H}_c = \begin{pmatrix} \frac{\partial_c^2 f}{\partial_c w_3^2} & \frac{\partial_c^2 f}{\partial_c w_3\, \partial_c w_4} \\ \frac{\partial_c^2 f}{\partial_c w_3\, \partial_c w_4} & \frac{\partial_c^2 f}{\partial_c w_4^2} \end{pmatrix} = \begin{pmatrix} -5 & 0 \\ 0 & 1 \end{pmatrix}$$

Because $\mathbf{H_c}$ is indefinite, the stationary point does not yield a maximum.

The reason the preceding solution does not yield the optimum solution is that the specific choices of **Y** and **Z** are not optimum. In fact, to find the optimum, we need to keep on altering our choices of **Y** and **Z** until the sufficiency condition is satisfied. This will be equivalent to locating the optimum extreme point of the linear programming solution space. For example, consider $\mathbf{Y} = (w_2,\ w_4)$ and $\mathbf{Z} = (w_1,\ w_3)$. The corresponding constrained gradient vector becomes

$$\nabla_c f = (4w_1,\ 0) - (6w_2,\ 0)\begin{pmatrix} \frac{1}{2w_2} & 0 \\ \frac{1}{2w_4} & \frac{1}{2w_4} \end{pmatrix}\begin{pmatrix} 2w_1 & 2w_3 \\ 2w_1 & 0 \end{pmatrix} = (-2w_1,\ 6w_3)$$

The corresponding stationary point is given by $w_1 = 0,\ w_2 = \sqrt{5},\ w_3 = 0,\ w_4 = \sqrt{8}$. Because

$$\mathbf{H}_c = \begin{pmatrix} -2 & 0 \\ 0 & -6 \end{pmatrix}$$

is negative-definite, the solution is a maximum point.

The result is verified graphically in Figure 20.6. The first solution ($x_1 = 4,\ x_2 = 1$) is not optimal, and the second ($x_1 = 0,\ x_2 = 5$) is. You can verify that the remaining two extreme points of the solution space are not optimal. In fact, the extreme point ($x_1 = 0,\ x_2 = 0$) can be shown by the sufficiency condition to yield a minimum point.

The sensitivity coefficients $\nabla_{\mathbf{Y}_0} f \mathbf{J}^{-1}$ when applied to linear programming yield the dual values. To illustrate this point for the given numerical example, let u_1 and u_2 be

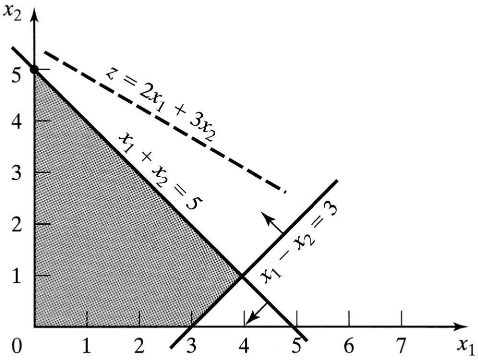


FIGURE 20.6
Extreme points of the solution space of the linear program

the corresponding dual variables. At the optimum point ($w_1 = 0$, $w_2 = \sqrt{5}$, $w_3 = 0$, $w_4 = \sqrt{8}$), these dual variables are given by

$$(u_1, u_2) = \nabla_{\mathbf{Y}_0}\mathbf{J}^{-1} = (6w_2, 0)\begin{pmatrix} \frac{1}{2w_2} & 0 \\ \frac{1}{2w_4} & \frac{1}{2w_4} \end{pmatrix} = (3, 0)$$

The corresponding dual objective value is $5u_1 + 3u_2 = 15$, which equals the optimal primal objective value. The given solution also satisfies the dual constraints and hence is optimal and feasible. This shows that the sensitivity coefficients are indeed the LP dual variables. In fact, both have the same interpretation.

We can draw some general conclusions from the application of the Jacobian method to the linear programming problem. From the numerical example, the necessary conditions require the independent variables to equal zero. Also, the sufficiency conditions indicate that the Hessian is a diagonal matrix. Thus, all its diagonal elements must be positive for a minimum and negative for a maximum. The observations demonstrate that the necessary condition is equivalent to specifying that only basic (feasible) solutions are needed to locate the optimum solution. In this case the independent variables are equivalent to the nonbasic variables in the linear programming problem. Also, the sufficiency condition demonstrates the strong relationship between the diagonal elements of the Hessian matrix and the optimality indicator $z_j - c_j$ (see Section 7.2) in the simplex method.[1]

PROBLEM SET 20.2B

1. Suppose that Example 20.2-2 is solved in the following manner. First, solve the constraints expressing x_1 and x_2 in terms of x_3; then use the resulting equations to express the objective function in terms of x_3 only. By taking the derivative of the new objective function with respect to x_3, we can determine the points of maxima and minima.

(a) Would the derivative of the new objective function (expressed in terms of x_3) be different from that obtained by the Jacobian method?

(b) How does the suggested procedure differ from the Jacobian method?

2. Apply the Jacobian method to Example 20.2-1 by selecting $\mathbf{Y} = (x_2, x_3)$ and $\mathbf{Z} = (x_1)$.

3. Solve by the Jacobian method:

$$\text{Minimize } f(\mathbf{X}) = \sum_{i=1}^{n} x_i^2$$

subject to

$$\prod_{i=1}^{n} x_i = C$$

where C is a positive constant. Suppose that the right-hand side of the constraint is changed to $C + \delta$, where δ is a small positive quantity. Find the corresponding change in the optimal value of f.

[1] For a formal proof of the validity of these results for the general linear programming problem, see H. Taha and G. Curry, "Classical Derivation of the Necessary and Sufficient Conditions for Optimal Linear Programs," *Operations Research,* Vol. 19, 1971, pp. 1045–1049. The paper shows that the key ideas of the simplex method can be derived by the Jacobian method.

4. Solve by the Jacobian method

$$\text{Minimize } f(\mathbf{X}) = 5x_1^2 + x_2^2 + 2x_1x_2$$

subject to

$$g(\mathbf{X}) = x_1x_2 - 10 = 0$$

(a) Find the change in the optimal value of $f(\mathbf{X})$ if the constraint is replaced by $x_1x_2 - 9.99 = 0$.

(b) Find the change in value of $f(\mathbf{X})$ in the neighborhood of the feasible point $(2, 5)$ given that $x_1x_2 = 9.99$ and $\partial x_1 = .01$.

5. Consider the problem:

$$\text{Maximize } f(\mathbf{X}) = x_1^2 + 2x_2^2 + 10x_3^2 + 5x_1x_2$$

subject to

$$g_1(\mathbf{X}) = x_1 + x_2^2 + 3x_2x_3 - 5 = 0$$

$$g_2(\mathbf{X}) = x_1^2 + 5x_1x_2 + x_3^2 - 7 = 0$$

Apply the Jacobian method to find $\partial f(\mathbf{X})$ in the feasible neighborhood of the feasible point $(1, 1, 1)$. Assume that this feasible neighborhood is specified by $\partial g_1 = -.01$, $\partial g_2 = .02$, and $\partial x_1 = .01$.

6. Consider the problem

$$\text{Minimize } f(\mathbf{X}) = x_1^2 + x_2^2 + x_3^2 + x_4^2$$

subject to

$$g_1(\mathbf{X}) = x_1 + 2x_2 + 3x_3 + 5x_4 - 10 = 0$$

$$g_2(\mathbf{X}) = x_1 + 2x_2 + 5x_3 + 6x_4 - 15 = 0$$

(a) Show that by selecting x_3 and x_4 as independent variables, the Jacobian method fails to provide a solution and state the reason.

(b) Now solve the problem using x_1 and x_3 as independent variables and apply the sufficiency condition to determine the type of the resulting stationary point.

(c) Determine the sensitivity coefficients given the solution in (b).

7. Consider the linear programming problem.

$$\text{Maximize } f(\mathbf{X}) = \sum_{j=1}^{n} c_jx_j$$

subject to

$$g_i(\mathbf{X}) = \sum_{j=1}^{n} a_{ij}x_j - b_i = 0 \qquad i = 1, 2, \ldots, m$$

$$x_j \geq 0, \quad j = 1, 2, \ldots, n$$

Neglecting the nonnegativity constraint, show that the constrained derivatives $\nabla_c f(\mathbf{X})$ for this problem yield the same expression for $\{z_j - c_j\}$ defined by the optimality condition of the linear programming problem (Section 7.2)—that is,

$$\{z_j - c_j\} = \{\mathbf{C}_B\mathbf{B}^{-1}\mathbf{P}_j - c_j\}, \text{ for all } j$$

Can the constrained-derivative method be applied directly to the linear programming problem? Why or why not?

Lagrangean Method. In the Jacobian method, let the vector $\boldsymbol{\lambda}$ represent the sensitivity coefficients—that is

$$\boldsymbol{\lambda} = \nabla_{\mathbf{Y}_0}\mathbf{J}^{-1} = \frac{\partial f}{\partial \mathbf{g}}$$

Thus,

$$\partial f - \boldsymbol{\lambda}\ \partial \mathbf{g} = 0$$

This equation satisfies the necessary conditions for stationary points because $\frac{\partial f}{\partial \mathbf{g}}$ is computed such that $\nabla_c f = \mathbf{0}$. A more convenient form for presenting these equations is to take their partial derivatives with respect to all x_j. This yields

$$\frac{\partial}{\partial x_j}(f - \boldsymbol{\lambda}\mathbf{g}) = 0, \quad j = 1, 2, \ldots, n$$

The resulting equations together with the constraint equations $\mathbf{g}(\mathbf{X}) = \mathbf{0}$ yield the feasible values of $\mathbf{X}$ and $\boldsymbol{\lambda}$ that satisfy the *necessary* conditions for stationary points.

The given procedure defines the *Lagrangean method* for identifying the stationary points of optimization problems with *equality* constraints. The procedure can be developed formally as follows. Let

$$L(\mathbf{X}, \boldsymbol{\lambda}) = f(\mathbf{X}) - \boldsymbol{\lambda}\mathbf{g}(\mathbf{X})$$

The function L is called the **Lagrangean function** and the parameters $\boldsymbol{\lambda}$ the **Lagrange multipliers**. By definition, these multipliers have the same interpretation as the sensitivity coefficients of the Jacobian method.

The equations

$$\frac{\partial L}{\partial \boldsymbol{\lambda}} = \mathbf{0}, \frac{\partial L}{\partial \mathbf{X}} = \mathbf{0}$$

give the necessary conditions for determining stationary points of $f(\mathbf{X})$ subject to $\mathbf{g}(\mathbf{X}) = \mathbf{0}$. The sufficiency conditions for the Lagrangean method will be stated without proof. Define

$$\mathbf{H}^B = \left(\begin{array}{c|c} \mathbf{0} & \mathbf{P} \\ \hline \mathbf{P}^T & \mathbf{Q} \end{array}\right)_{(m+n)\times(m+n)}$$

where

$$\mathbf{P} = \begin{pmatrix} \nabla g_1(\mathbf{X}) \\ \vdots \\ \nabla g_m(\mathbf{X}) \end{pmatrix}_{m\times n}, \quad \mathbf{Q} = \left\|\frac{\partial^2 L(\mathbf{X}, \boldsymbol{\lambda})}{\partial x_i \partial x_j}\right\|_{n\times n}, \quad \text{for all } i \text{ and } j$$

The matrix $\mathbf{H}^B$ is the **bordered Hessian matrix.**

Given the stationary point $(\mathbf{X}_0, \boldsymbol{\lambda}_0)$ for the Lagrangean function $L(\mathbf{X}, \boldsymbol{\lambda})$ and the bordered Hessian matrix $\mathbf{H}^B$ evaluated at $(\mathbf{X}_0, \boldsymbol{\lambda}_0)$, then $\mathbf{X}_0$ is

1. A maximum point if, starting with the principal major determinant of order $(2m + 1)$, the *last* $(n - m)$ principal minor determinants of $\mathbf{H}^B$ form an alternating sign pattern starting with $(-1)^{m+1}$.
2. A minimum point if, starting with the principal minor determinant of order $(2m + 1)$, the *last* $(n - m)$ principal minor determinants of $\mathbf{H}^B$ have the sign of $(-1)^m$.

These conditions are sufficient, but not necessary, for identifying an extreme point. This means that a stationary point may be an extreme point without satisfying these conditions.

Other conditions exist that are both necessary and sufficient for identifying extreme points. However, the procedure may be computationally intractable. Define the following matrix at the stationary point $(\mathbf{X}_0, \boldsymbol{\lambda}_0)$:

$$\Delta = \left(\begin{array}{c|c} \mathbf{0} & \mathbf{P} \\ \hline \mathbf{P}^T & \mathbf{Q} - \mu\mathbf{I} \end{array}\right)$$

where μ is an unknown parameter. Consider the determinant $|\boldsymbol{\Delta}|$; then each of the real $(n - m)$ roots μ of the polynomial

$$|\Delta| = 0$$

must be

1. Negative if $\mathbf{X}_0$ is a maximum point.
2. Positive if $\mathbf{X}_0$ is a minimum point.

Example 20.2-4

Consider the problem of Example 20.2-2. The Lagrangean function is

$$L(\mathbf{X}, \boldsymbol{\lambda}) = x_1^2 + x_2^2 + x_3^2 - \lambda_1(x_1 + x_2 + 3x_3 - 2) - \lambda_2(5x_1 + 2x_2 + x_3 - 5)$$

This yields the following necessary conditions:

$$\frac{\partial L}{\partial x_1} = 2x_1 - \lambda_1 - 5\lambda_2 = 0$$

$$\frac{\partial L}{\partial x_2} = 2x_2 - \lambda_1 - 2\lambda_2 = 0$$

$$\frac{\partial L}{\partial x_3} = 2x_3 - 3\lambda_1 - \lambda_2 = 0$$

$$\frac{\partial L}{\partial \lambda_1} = -(x_1 + x_2 + 3x_3 - 2) = 0$$

$$\frac{\partial L}{\partial \lambda_2} = -(5x_1 + 2x_2 + x_3 - 5) = 0$$

The solution to these simultaneous equations yields

$$\mathbf{X}_0 = (x_1, x_2, x_3) = (.8043, .3478, .2826)$$

$$\boldsymbol{\lambda} = (\lambda_1, \lambda_2) = (.0870, .3043)$$

This solution combines the results of Examples 20.2-2 and 20.2-3. The values of the Lagrange multipliers $\boldsymbol{\lambda}$ equal the sensitivity coefficients obtained in Example 20.2-3 (allowing for the roundoff error). The result shows that these coefficients are independent of the choice of the dependent vector $\mathbf{Y}$ in the Jacobian method.

To show that the given point is a minimum, consider

$$\mathbf{H}^B = \left(\begin{array}{cc|ccc} 0 & 0 & 1 & 1 & 3 \\ 0 & 0 & 5 & 2 & 1 \\ \hline 1 & 5 & 2 & 0 & 0 \\ 1 & 2 & 0 & 2 & 0 \\ 3 & 1 & 0 & 0 & 2 \end{array}\right)$$

Because $n = 3$ and $m = 2$, $n - m = 1$, and we need to check the determinant of $\mathbf{H}^B$ only, which must have the sign of $(-1)^2$ for the stationary point $\mathbf{X}_0$ to be a minimum. Because det $\mathbf{H}^B = 460 > 0$, $\mathbf{X}^0$ is a minimum point.

Example 20.2-5

Consider the problem

$$\text{Minimize } z = x_1^2 + x_2^2 + x_3^2$$

subject to

$$4x_1 + x_2^2 + 2x_3 - 14 = 0$$

The Lagrangean function is

$$L(\mathbf{X}, \lambda) = x_1^2 + x_2^2 + x_3^2 - \lambda(4x_1 + x_2^2 + 2x_3 - 14)$$

The associated necessary conditions are given as:

$$\frac{\partial L}{\partial x_1} = 2x_1 - 4\lambda = 0$$

$$\frac{\partial L}{\partial x_2} = 2x_2 - 2\lambda x_2 = 0$$

$$\frac{\partial L}{\partial x_3} = 2x_3 - 2\lambda = 0$$

$$\frac{\partial L}{\partial \lambda} = -(4x_1 + x_2^2 + 2x_3 - 14) = 0$$

These equations yield infinity of solutions because $\frac{\partial L}{\partial x_2} = 0$ is independent of x_2. For the sake of the example, we will consider the following three solutions:

$$(\mathbf{X}_0, \lambda_0)_1 = (2, 2, 1, 1)$$

$$(\mathbf{X}_0, \lambda_0)_2 = (2, -2, 1, 1)$$

$$(\mathbf{X}_0, \lambda_0)_3 = (2.8, 0, 1.4, 1.4)$$

The sufficiency conditions yields

$$\mathbf{H}^B = \begin{pmatrix} 0 & 4 & 2x_2 & 2 \\ 4 & 2 & 0 & 0 \\ 2x_2 & 0 & 2 - 2\lambda & 0 \\ 2 & 0 & 0 & 2 \end{pmatrix}$$

Because $m = 1$ and $n = 3$, for a stationary point to be a minimum, the sign of the last $(3 - 1) = 2$ principal minor determinants must be that of $(-1)^m = -1$. Thus, for $(\mathbf{X}_0, \lambda_0)_1 = (2, 2, 1, 1)$

$$\begin{vmatrix} 0 & 4 & 4 \\ 4 & 2 & 0 \\ 4 & 0 & 0 \end{vmatrix} = -32 < 0, \quad \begin{vmatrix} 0 & 4 & 4 & 2 \\ 4 & 2 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 \end{vmatrix} = -64 < 0$$

For $(\mathbf{X}_0, \lambda_0)_2 = (2, -2, 1, 1)$,

$$\begin{vmatrix} 0 & 4 & -4 \\ 4 & 2 & 0 \\ -4 & 0 & 0 \end{vmatrix} = -32 < 0, \quad \begin{vmatrix} 0 & 4 & -4 & 2 \\ 4 & 2 & 0 & 0 \\ -4 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 \end{vmatrix} = -64 < 0$$

Finally, for $(\mathbf{X}_0, \lambda_0)_3 = (2.8, 0, 1.4, 1.4)$

$$\begin{vmatrix} 0 & 4 & 0 \\ 4 & 2 & 0 \\ 0 & 0 & -.8 \end{vmatrix} = 12.8 > 0, \quad \begin{vmatrix} 0 & 4 & 0 & 2 \\ 4 & 2 & 0 & 0 \\ 0 & 0 & -.8 & 0 \\ 2 & 0 & 0 & 2 \end{vmatrix} = 32 > 0$$

This shows that $(\mathbf{X}_0)_1$ and $(\mathbf{X}_0)_2$ are minimum points. $(\mathbf{X}_0)_3$ does not satisfy the sufficiency conditions of either a maximum or a minimum. This does not mean that it is not an extreme point because the given conditions are sufficient only.

To illustrate the use of the other sufficiency condition that employs the roots of polynomial, consider

$$\Delta = \begin{pmatrix} 0 & 4 & 2x_2 & 2 \\ 4 & 2 - \mu & 0 & 0 \\ 2x_2 & 0 & 2 - 2\lambda - \mu & 0 \\ 2 & 0 & 0 & 2 - \mu \end{pmatrix}$$

Now, for $(\mathbf{X}_0, \lambda_0)_1 = (2, 2, 1, 1)$

$$|\Delta| = 9\mu^2 - 26\mu + 16 = 0$$

This gives $\mu = 2$ or $\frac{8}{9}$. Because all $\mu > 0$, $(\mathbf{X}_0)_1 = (2, 2, 1)$ is a minimum point. For $(\mathbf{X}_0, \lambda_0)_2 = (2, -2, 1, 1)$,

$$|\Delta| = 9\mu^2 - 26\mu + 16 = 0$$

which is the same as in the previous case. Hence $(\mathbf{X}_0)_2 = (2, -2, 1)$ is a minimum point. Finally, for $(\mathbf{X}_0, \lambda_0)_3 = (2.8, 0, 1.4, 1.4)$,

$$|\Delta| = 5\mu^2 - 6\mu - 8 = 0$$

This gives $\mu = 2$ and $-.8$, which means that the identity of $(\mathbf{X}_0)_3 = (2.8, 0\,1.4)$ is not known.

PROBLEM SET 20.2C

1. Solve the following linear programming problem by both the Jacobian and the Lagrangean methods:

$$\text{Maximize } f(\mathbf{X}) = 5x_1 + 3x_2$$

subject to

$$g_1(\mathbf{X}) = x_1 + 2x_2 + x_3 - 6 = 0$$
$$g_2(\mathbf{X}) = 3x_1 + x_2 + x_4 - 9 = 0$$
$$x_1, x_2, x_3, x_4 \geq 0$$

2. Find the optimal solution to the problem

$$\text{Minimize } f(\mathbf{X}) = x_1^2 + 2x_2^2 + 10x_3^2$$

subject to

$$g_1(\mathbf{X}) = x_1 + x_2^2 + x_3 - 5 = 0$$
$$g_2(\mathbf{X}) = x_1 + 5x_2 + x_3 - 7 = 0$$

Suppose that $g_1(\mathbf{X}) = .01$ and $g_2(\mathbf{X}) = .02$. Find the corresponding change in the optimal value of $f(\mathbf{X})$.

3. Solve Problem 6, Set 20.2b by the Lagrangean method and verify that the values of the Lagrange multipliers are the same as the sensitivity coefficients obtained in Problem 6, Set 20.2b.

20.2.2 Inequality Constraints

This section extends the Lagrangean method to handle inequality constraints. The main contribution of the section is the development of the general Karush-Kuhn-Tucker (KKT) conditions, which provide the basic theory for nonlinear programming.

Extension of the Lagrangean Method. Consider

$$\text{Maximize } z = f(\mathbf{X})$$

subject to

$$g_i(\mathbf{X}) \leq 0, \quad i = 1, 2, \ldots, m$$

The nonnegativity constraints $\mathbf{X} \geq 0$, if any, are included in the m constraints.

If the *un*constrained optimum of $f(\mathbf{X})$ does not satisfy all constraints, the constrained optimum must occur at a boundary point of the solution space. This means that at least one constraint must be satisfied in equation form. The procedure thus involves the following steps.

Step 1. Solve the unconstrained problem

$$\text{Maximize } z = f(\mathbf{X})$$

If the resulting optimum satisfies all the constraints, stop because all constraints are redundant. Otherwise, set k = 1 and go to step 2.

Step 2. Activate any k constraints (i.e., convert them into equalities) and optimize $f(\mathbf{X})$ subject to the k active constraints using the Lagrangean method. If the resulting solution is feasible with respect to the remaining constraints, stop;

it is a *local* optimum.[2] Otherwise, activate another set of k constraints and repeat the step. If *all* sets of active constraints taken k at a time are considered without encountering a feasible solution, go to step 3.

Step 3. If $k = m$, stop; no feasible solution exists. Otherwise, set $k = k + 1$ and go to step 2.

An important point often neglected in presenting the procedure is that it does *not* guarantee global optimality even when the problem is well behaved (possesses a *unique* optimum). Another important point is the implicit misconception that, for $p < q$, the optimum of $f(\mathbf{X})$ subject to p equality constraints is always better than its optimum subject to q equality constraints. This is true, in general, only if the q constraints form a subset of the p constraints. The following example is designed to illustrate these points.

Example 20.2-6

$$\text{Maximize } z = -(2x_1 - 5)^2 - (2x_2 - 1)^2$$

subject to

$$x_1 + 2x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

The graphical representation in Figure 20.7 should assist in understanding the analytic procedure. Observe that the problem is well behaved (concave objective function subject to a convex solution space), which means that a reasonably well defined algorithm should guarantee global optimality. Yet, as will be shown, the extended Lagrangean method produces a local maximum only.

The unconstrained optimum is obtained by solving

$$\frac{\partial z}{\partial x_1} = -4(2x_1 - 5) = 0$$

$$\frac{\partial z}{\partial x_2} = -4(2x_2 - 1) = 0$$

This gives $(x_1, x_2) = (\frac{5}{2}, \frac{1}{2})$, which does not satisfy the constraint $x_1 + 2x_2 \leq 2$. Thus, the constraints are activated one at a time. Consider $x_1 = 0$. The Lagrangean function is

$$L(x_1, x_2, \lambda) = -(2x_1 - 5)^2 - (2x_2 - 1)^2 - \lambda x_1$$

Thus,

$$\frac{\partial L}{\partial x_1} = -4(2x_1 - 5) - \lambda = 0$$

$$\frac{\partial L}{\partial x_2} = -4(2x_2 - 1) \qquad = 0$$

[2] A *local* optimum is defined from among all the optima resulting from optimizing $f(\mathbf{X})$ subject to *all* combinations of k *equality* constraints, $k = 1, 2, \ldots, m$.

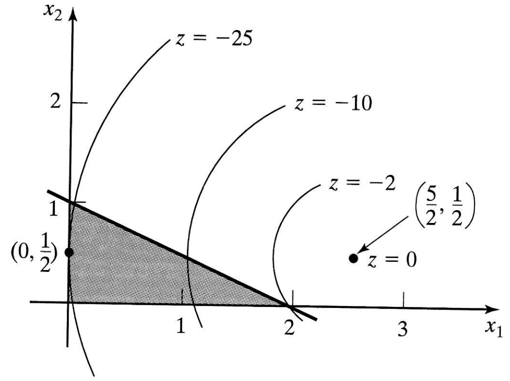


FIGURE 20.7
Solution space of Example 20.2-6

$$\frac{\partial L}{\partial \lambda} = -x_1 = 0$$

This gives the solution point $(x_1, x_2) = (0, \frac{1}{2})$, which can be shown by the sufficiency condition to be a maximum. Because this point satisfies all other constraints, the procedure terminates with $(x_1, x_2) = (0, \frac{1}{2})$ as a local optimal solution to the problem. The objective value is $z = -25$. (The remaining constraints $x_2 \geq 0$ and $x_1 = 2x_2 \leq 2$, activated one at a time, yield infeasible solutions.)

In Figure 20.7, the feasible solution $(x_1, x_2) = (2, 0)$, which is the point of intersection of the *two* constraints $x_2 = 0$ and $x_1 + 2x_2 = 2$, yields the objective value $z = -2$. This value is better than the one obtained with one active constraint.

The procedure just described illustrates that the best to be hoped for in using the extended Lagrangean method is a (possibly) good feasible solution. This is particularly true if the objective function is not unimodal. If the functions of the problem are well behaved (e.g., the problem possesses a unique constrained optimum as in Example 20.2-6), the procedure can be rectified to locate the global optimum. Specifically, consider the unconstrained optimum and the constrained optima subject to *all* sets of one active constraint, then two active constraints, and so on, until all m constraints are activated. The best of *all* the *feasible* optima is the global optimum.

If this procedure is followed by Example 20.2-6, it will be necessary to solve seven problems before global optimality is verified. This indicates the limited use of the method in solving problems of any practical size.

The Karush-Kuhn-Tucker (KKT) Conditions.[3] This section develops the KKT *necessary* conditions for identifying stationary points of a nonlinear constrained problem subject to inequality constraints. The development is based on the Lagrangean method. These conditions are also sufficient under certain rules that will be stated later.

[3] Historically, W. Karush was the first to develop the KKT conditions in 1939 as part of his M.S. thesis at the University of Chicago. The same conditions were developed independently in 1951 by W. Kuhn and A. Tucker.

Consider the problem

$$\text{Maximize } z = f(\mathbf{X})$$

subject to

$$\mathbf{g}(\mathbf{X}) \leq \mathbf{0}$$

The inequality constraints may be converted into equations by using *nonnegative* slack variables. Let S_i^2 (≥ 0) be the slack quantity added to the ith constraint $g_i(\mathbf{X}) \leq 0$ and define

$$\mathbf{S} = (S_1, S_2, \ldots, S_m)^T,\ \mathbf{S}^2 = (S_1^2, S_2^2, \ldots, S_{m)}^2)^T$$

where m is the total number of inequality constraints. The Lagrangean function is thus given by

$$L(\mathbf{X}, \mathbf{S}, \boldsymbol{\lambda}) = f(\mathbf{X}) - \boldsymbol{\lambda}[\mathbf{g}(\mathbf{X}) + \mathbf{S}^2]$$

Given the constraints

$$\mathbf{g}(\mathbf{X}) \leq \mathbf{0}$$

a necessary condition for optimality is that λ be nonnegative (nonpositive) for maximization (minimization) problems. This result is justified as follows. The vector $\boldsymbol{\lambda}$ measures the rate of variation of f with respect to $\mathbf{g}$—that is,

$$\boldsymbol{\lambda} = \frac{\partial f}{\partial \mathbf{g}}$$

In the maximization case, as the right-hand side of the constraint $\mathbf{g}(\mathbf{X}) \leq \mathbf{0}$ changes from $\mathbf{0}$ to $\partial\mathbf{g}(> \mathbf{0})$, the solution space becomes less constrained and hence f cannot decrease. This means that $\boldsymbol{\lambda} \geq \mathbf{0}$. Similarly for minimization, as the right-hand side of the constraints increases, f cannot increase, which implies that $\boldsymbol{\lambda} \leq \mathbf{0}$. If the constraints are equalities, that is, $g(\mathbf{X}) = \mathbf{0}$, then λ becomes unrestricted in sign (see Problem 2, Set 20.2d).

The restrictions on $\boldsymbol{\lambda}$ are part of the KKT necessary conditions. The remaining conditions will now be derived.

Taking the partial derivatives of L with respect to $\mathbf{X}$, $\mathbf{S}$, and $\boldsymbol{\lambda}$, we obtain

$$\frac{\partial L}{\partial \mathbf{X}} = \nabla f(\mathbf{X}) - \boldsymbol{\lambda}\nabla\mathbf{g}(\mathbf{X}) = \mathbf{0}$$

$$\frac{\partial L}{\partial S_i} = -2\lambda_i S_i = 0,\ i = 1, 2, \ldots, m$$

$$\frac{\partial L}{\partial \boldsymbol{\lambda}} = -(\mathbf{g}(\mathbf{X}) + \mathbf{S}^2) = \mathbf{0}$$

The second set of equations reveals the following results:

1. If $\lambda_i \neq 0$, then $S_i^2 = 0$, which means that the corresponding resource is scarce and, hence, it is consumed completely (equality constraint).
2. If $S_i^2 > 0$, then $\lambda_i = 0$. This means resource i is not scarce and, consequently, it has no effect on the value of f (i.e., $\lambda_i = \frac{\partial f}{\partial g_i} = 0$).

From the second and third sets of equations, we obtain

$$\lambda_i g_i(\mathbf{X}) = 0, \quad i = 1, 2, \ldots, m$$

This new condition essentially repeats the foregoing argument, because if $\lambda_i > 0$, $g_i(\mathbf{X}) = 0$ or $S_i^2 = 0$; and if $g_i(\mathbf{X}) < 0$, $S_i^2 > 0$, and $\lambda_i = 0$.

The KKT necessary conditions for the maximization problem can now be summarized as follows:

$$\boldsymbol{\lambda} \geq \mathbf{0}$$

$$\nabla f(\mathbf{X}) - \boldsymbol{\lambda}\nabla \mathbf{g}(\mathbf{X}) = \mathbf{0}$$

$$\lambda_i g_i(\mathrm{X}) = 0, \quad i = 1, 2, \ldots, m$$

$$\mathbf{g}(\mathbf{X}) \leq \mathbf{0}$$

These conditions apply to the minimization case as well, except that $\boldsymbol{\lambda}$ must be nonpositive (verify!). In both maximization and minimization, the Lagrange multipliers corresponding to equality constraints must be unrestricted in sign.

Sufficiency of the KKT Conditions. The Kuhn-Tucker necessary conditions are also sufficient if the objective function and the solution space satisfy the conditions in Table 20.1.

TABLE 20.1

Sense of optimization	Required conditions	
	Objective function	*Solution space*
Maximization	Concave	Convex set
Minimization	Convex	Convex set

It is simpler to verify that a function is convex or concave than to prove that a solution space is a convex set. For this reason, we provide a list of conditions that are easier to apply in practice in the sense that the convexity of the solution space can be established by checking the convexity or concavity of the constraint functions. To provide these conditions, we define the generalized nonlinear problems as

$$\text{Maximize or minimize } z = f(\mathbf{X})$$

subject to

$$g_i(\mathbf{X}) \leq 0, \quad i = 1, 2, \ldots, r$$

$$g_i(\mathbf{X}) \geq 0, \quad i = r + 1, \ldots, p$$

$$g_i(\mathbf{X}) = 0, \quad i = p + 1, \ldots, m$$

$$L(\mathbf{X}, \mathbf{S}, \boldsymbol{\lambda}) = f(\mathbf{X}) - \sum_{i=1}^{r}\lambda_i[g_i(\mathbf{X}) + S_i^2] - \sum_{i=r+1}^{p}\lambda_i[g_i(\mathbf{X}) - S_i^2] - \sum_{i=p+1}^{m}\lambda_i g_i(\mathbf{X})$$

where λ_i is the Lagrangean multiplier associated with constraint i. The conditions for establishing the sufficiency of the KKT conditions are summarized in Table 20.2.

TABLE 20.2

Sense of optimization	Required conditions			
	$f(\mathbf{X})$	$g_i(\mathbf{X})$	λ_i	
Maximization	Concave	Convex	≥ 0	$(1 \leq i \leq r)$
		Concave	≤ 0	$(r + 1 \leq i \leq p)$
		Linear	Unrestricted	$(p + 1 \leq i \leq m)$
Minimization	Convex	Convex	≤ 0	$(1 \leq i \leq r)$
		Concave	≥ 0	$(r + 1 \leq i \leq p)$
		Linear	Unrestricted	$(p + 1 \leq i \leq m)$

The conditions in Table 20.2 represent only a subset of the conditions in Table 20.1. The reason is that a solution space may be convex without satisfying the conditions in Table 20.2.

Table 20.2 is valid because the given conditions yield a concave Lagrangean function $L(\mathbf{X}, \mathbf{S}, \boldsymbol{\lambda})$ in case of maximization and a convex $L(\mathbf{X}, \mathbf{S}, \boldsymbol{\lambda})$ in case of minimization. This result is verified by noticing that if $g_i(x)$ is convex, then $\lambda_i g_i(x)$ is convex if $\lambda_i \geq 0$ and concave if $\lambda_i \leq 0$. Similar interpretations can be established for all the remaining conditions. Observe that a linear function is both convex and concave. Also, if a function f is concave, then $(-f)$ is convex, and vice versa.

Example 20.2-7

Consider the following minimization problem:

$$\text{Minimize } f(\mathbf{X}) = x_1^2 + x_2^2 + x_3^2$$

subject to

$$\begin{aligned} g_1(\mathbf{X}) &= 2x_1 + x_2 - 5 \leq 0 \\ g_2(\mathbf{X}) &= x_1 + x_3 - 2 \leq 0 \\ g_3(\mathbf{X}) &= 1 - x_1 \leq 0 \\ g_4(\mathbf{X}) &= 2 - x_2 \leq 0 \\ g_5(\mathbf{X}) &= -x_3 \leq 0 \end{aligned}$$

This is a minimization problem; hence $\boldsymbol{\lambda} \leq \mathbf{0}$. The KKT conditions are thus given as

$$(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) \leq \mathbf{0}$$

$$(2x_1, 2x_2, 2x_3) - (\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)\begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} = \mathbf{0}$$

$$\lambda_1 g_1 = \lambda_2 g_2 = \cdots = \lambda_5 g_5 = 0$$

$$\mathbf{g}(\mathbf{X}) \leq \mathbf{0}$$

These conditions reduce to

$$\begin{aligned}
\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5 &\leq 0\\
2x_1 - 2\lambda_1 - \lambda_2 + \lambda_3 &= 0\\
2x_2 - \lambda_1 + \lambda_4 &= 0\\
2x_3 - \lambda_2 + \lambda_5 &= 0\\
\lambda_1(2x_1 + x_2 - 5) &= 0\\
\lambda_2(x_1 + x_3 - 2) &= 0\\
\lambda_3(1 - x_1) &= 0\\
\lambda_4(2 - x_2) &= 0\\
\lambda_5 x_3 &= 0\\
2x_1 + x_2 &\leq 5\\
x_1 + x_3 &\leq 2\\
x_1 \geq 1, x_2 \geq 2, x_3 &\geq 0
\end{aligned}$$

The solution is $x_1 = 1, x_2 = 2, x_3 = 0, \lambda_1 = \lambda_2 = \lambda_5 = 0, \lambda_3 = -2, \lambda_4 = -4$. Because both $f(\mathbf{X})$ and the solution space $g(\mathbf{X}) \leq 0$ are convex, $L(\mathbf{X}, \mathbf{S}, \boldsymbol{\lambda})$ must be convex and the resulting stationary point yields a global constrained minimum. The example shows that the procedure is not suitable for numerical computations because it may be difficult to solve the resulting conditions explicitly. The KKT conditions are central to the development of the nonlinear programming algorithms in Chapter 21.

PROBLEM SET 20.2D

1. Consider the problem:

$$\text{Maximize } f(\mathbf{X})$$

subject to

$$\mathbf{g}(\mathbf{X}) \geq 0$$

Show that the KKT conditions are the same as in Section 20.2.2, except that the Lagrange multipliers $\boldsymbol{\lambda}$ are nonpositive.

2. Consider the following problem:

$$\text{Maximize } f(\mathbf{X})$$

subject to

$$\mathbf{g}(\mathbf{X}) = 0$$

Show that the KKT conditions are

$$\nabla f(\mathbf{X}) - \boldsymbol{\lambda}\nabla \mathbf{g}(\mathbf{X}) = \mathbf{0}$$

$$\mathbf{g}(\mathbf{X}) = \mathbf{0}$$

$$\boldsymbol{\lambda} \text{ unrestricted in sign}$$

3. Write the KKT necessary conditions for the following problems.

(a) Maximize $f(\mathbf{X}) = x_1^3 - x_2^2 + x_1x_3^2$
subject to

$$x_1 + x_2^2 + x_3 = 5$$
$$5x_1^2 - x_2^2 - x_3 \geq 0$$
$$x_1, x_2, x_3 \geq 0$$

(b) Minimize $f(\mathbf{X}) = x_1^4 + x_2^2 + 5x_1x_2x_3$
subject to

$$x_1^2 - x_2^2 + x_3^3 \leq 10$$
$$x_1^3 + x_2^2 + 4x_3^2 \geq 20$$

4. Consider the problem

$$\text{Maximize } f(\mathbf{X})$$

subject to

$$\mathbf{g}(\mathbf{X}) = \mathbf{0}$$

Given $f(\mathbf{X})$ is concave and $g_i(\mathbf{X})$ $(i = 1, 2, \ldots, m)$ is a *linear* function, show that the KKT necessary conditions are also sufficient. Is this result true if $g_i(\mathbf{X})$ is a convex *non*linear function for all i? Why?

5. Consider the problem

$$\text{Maximize } f(\mathbf{X})$$

subject to

$$g_1(\mathbf{X}) \geq 0, \ g_2(\mathbf{X}) = 0, \ g_3(\mathbf{X}) \leq 0$$

Develop the KKT conditions and give the stipulations under which the conditions are sufficient.

SELECTED REFERENCES

Bazarra, M., H. Shrali, and C. Shetty, *Nonlinear Programming Theory and Algorithms,* 2nd ed., Wiley, New York, 1993.

Beightler, C., D. Phillips, and D. Wilde, *Foundations of Optimization,* 2nd ed., Prentice Hall, NJ, 1979.

Rardin, R. *Optimization in Operations Research,* Prentice Hall, NJ, 1998.

CHAPTER 21

Nonlinear Programming Algorithms

The solution methods of nonlinear programming generally can be classified as either *direct* or *indirect* algorithms. Examples of direct methods are the gradient algorithms, where the maximum (minimum) of a problem is sought by following the fastest rate of increase (decrease) of the objective function. In indirect methods, the original problem is replaced by another from which the optimum is determined. Examples of these situations include quadratic programming, separable programming, and stochastic programming.

21.1 UNCONSTRAINED ALGORITHMS

This section presents two algorithms for the unconstrained problem: the *direct search* algorithm and the *gradient* algorithm.

21.1.1 Direct Search Method

Direct search methods apply primarily to strictly unimodal single-variable functions. Although the case may appear trivial, Section 21.1.2 shows that optimization of single-variable functions plays a key role in the development of the more general multivariable algorithms.

The idea of direct search methods is to identify the **interval of uncertainty** that includes the optimum solution point. The procedure locates the optimum by iteratively narrowing the interval of uncertainty to any desired level of accuracy.

Two closely related algorithms are presented in this section: Dichotomous and golden section search methods. Both algorithms seek the maximization of a unimodal function $f(x)$ over the interval $a \leq x \leq b$, which is known to include the optimum point x^*. The two methods start with $I_0 = (a, b)$ representing the initial interval of uncertainty.

General Step i. Let $I_{i-1} = (x_L, x_R)$ be the current interval of uncertainty (at iteration 0, $x_L = a$ and $x_R = b$). Next, define x_1 and x_2 such that

$$x_L < x_1 < x_2 < x_R$$

The next interval of uncertainty, I_i, is determined in the following manner:

1. If $f(x_1) > f(x_2)$, then $x_L < x^* < x_2$. Set $x_R = x_2$ and $I_i = (x_L, x_2)$ (see Figure 21.1[a]).
2. If $f(x_1) < f(x_2)$, then $x_1 < x^* < x_R$. Set $x_L = x_1$ and $I_i = (x_1, x_R)$ (see Figure 21.1[b]).
3. If $f(x_1) = f(x_2)$, then $x_1 < x^* < x_2$. Set $x_L = x_1$, $x_R = x_2$, and $I_i = (x_1, x_2)$.

The manner in which x_1 and x_2 are determined guarantees that $I_i < I_{i-1}$, as will be shown shortly. The algorithm terminates at iteration k if $I_k \leq \Delta$, where Δ is a user-specified level of accuracy.

The difference between the dichotomous and golden section methods occurs in the manner x_1 and x_2 are computed. The following table provides the formulas.

Dichotomous method	Golden section method
$x_1 = \frac{1}{2}(x_R + x_L - \Delta)$	$x_1 = x_R - (\frac{\sqrt{5}-1}{2})(x_R - x_L)$
$x_2 = \frac{1}{2}(x_R + x_L + \Delta)$	$x_2 = x_L + (\frac{\sqrt{5}-1}{2})(x_R - x_L)$

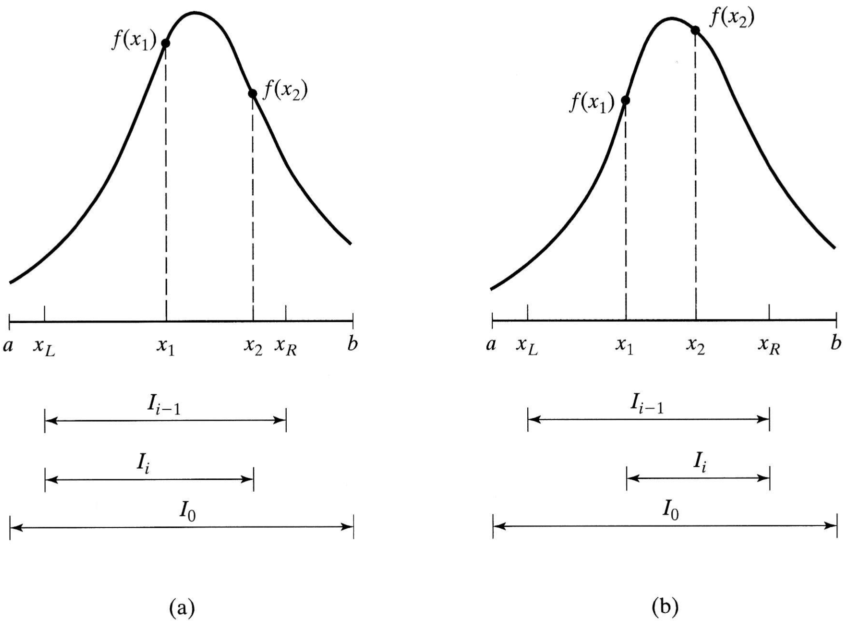


FIGURE 21.1
Illustration of the general step of the dichotomous/golden section search methods

In the dichotomous method, the values x_1 and x_2 sit symmetrically around the midpoint of the current interval of uncertainty. This means that

$$I_i = .5(I_{i-1} + \Delta)$$

Repeated application of the algorithm guarantees that the length of the interval of uncertainty will approach the desired accuracy, Δ.

In the golden section method, the idea is more involved. We notice that each iteration of the dichotomous method requires calculating the two values $f(x_1)$ and $f(x_2)$, but ends up discarding one of them. What the golden section proposes is to save computations by reusing the discarded value in the immediately succeeding iteration.

Define for $0 < \alpha < 1$,

$$x_1 = x_R - \alpha(x_R - x_L)$$
$$x_2 = x_L + \alpha(x_R - x_L)$$

Then the interval of uncertainty I_i at iteration i equals (x_L, x_2) or (x_1, x_R). Consider the case $I_i = (x_L, x_2)$, which means that x_1 is included in I_i. In iteration $i + 1$, we select x_2 equal to x_1 in iteration i, which leads to the following equation:

$$x_2(\text{iteration } i + 1) = x_1(\text{iteration } i)$$

Substitution yields

$$x_L + \alpha[x_2(\text{iteration } i) - x_L] = x_R - \alpha(x_R - x_L)$$

or

$$x_L + \alpha[x_L + \alpha(x_R - x_L) - x_L] = x_R - \alpha(x_R - x_L)$$

which finally simplifies to

$$\alpha^2 + \alpha - 1 = 0$$

This equation yields $\alpha = \frac{-1 \pm \sqrt{5}}{2}$. Because $0 \le \alpha \le 1$, we select the positive root $\alpha = \frac{-1 + \sqrt{5}}{2} \approx .681$.

The design of the golden section computations guarantees an α-reduction in successive intervals of uncertainty; that is

$$I_{i+1} = \alpha I_i$$

Compared to the dichotomous method, the golden section method converges more quickly to the desired level of accuracy. In addition, each iteration in the golden section method requires half the computations because the method always recycles one set of computations from the immediately preceding iteration.

Example 21.1-1

$$\text{Maximize } f(x) = \begin{cases} 3x, & 0 \le x \le 2 \\ \frac{1}{3}(-x + 20), & 2 \le x \le 3 \end{cases}$$

The maximum value of $f(x)$ occurs at $x = 2$. The following table demonstrates the calculations for iterations 1 and 2 using the dichotomous and the golden section methods. We will assume $\Delta = .1$.

Dichotomous method	Golden section method
Iteration 1	*Iteration 1*
$I_0 = (0, 3) \equiv (x_L, x_R)$	$I_0 = (0, 3) \equiv (x_L, x_R)$
$x_1 = .5(3 + 0 - .1) = 1.45, f(x_1) = 4.35$	$x_1 = 3 - .618(3 - 0) = 1.146, f(x_1) = 3.438$
$x_2 = .5(3 + 0 + .1) = 1.55, f(x_2) = 4.65$	$x_2 = 0 + .618(3 - 0) = 1.854, f(x_2) = 5.562$
$f(x_2) > f(x_1) \Rightarrow x_L = 1.45, I_1 = (1.45, 3)$	$f(x_2) > f(x_1) \Rightarrow x_L = 1.146, I_1 = (1.146, 3)$
Iteration 2	*Iteration 2*
$I_1 = (1.45, 3) \equiv (x_L, x_R)$	$I_1 = (1.146, 3) \equiv (x_L, x_R)$
$x_1 = .5(3 + 1.45 - .1) = 2.175, f(x_1) = 5.942$	$x_1 = x_2$ in iteration $0 = 1.854, f(x_1) = 5.562$
$x_2 = .5(3 + 1.45 + .1) = 2.275, f(x_2) = 5.908$	$x_2 = 1.146 + .618(3 - 1.146) = 2.292, f(x_2) = 5.903$
$f(x_1) > f(x_2) \Rightarrow x_R = 2.275, I_2 = (1.45, 2.275)$	$f(x_2) > f(x_1) \Rightarrow x_L = 1.854, I_1 = (1.854, 3)$

Continuing in the same manner, the interval of uncertainty will eventually narrow down to the desired Δ-tolerance.

Excel template ch21DichotomousGoldenSection.xls is designed to handle either method automatically. The input data include $f(x)$, a, b, and Δ. The function $f(x)$ is entered in cell E3 as

```
=IF(C3<=2,3*C3,(-C3+20)/3)
```

Note that C3 plays the role of x in $f(x)$. Limits a and b are entered in cells B4 and D4 to represent the admissible search range for $f(x)$. Also, the tolerance limit, Δ, is entered in cell B3. The search method is selected by entering **x** in either D5 (dichotomous) or F5 (golden section).

Figure 21.2 compares the two methods. Not only does the golden section method requires 40% less iterations, it also involves less calculations per iteration as we explained previously.

PROBLEM SET 21.1A

1. Use Excel template ch21DichotomousGoldenSection.xls to solve Example 21.1-1 assuming that $\Delta = .01$. Compare the amount of computations and the accuracy of the results with those in Figure 21.2.
2. Find the maximum of each of the following functions by dichotomous search. Assume that $\Delta = .05$.

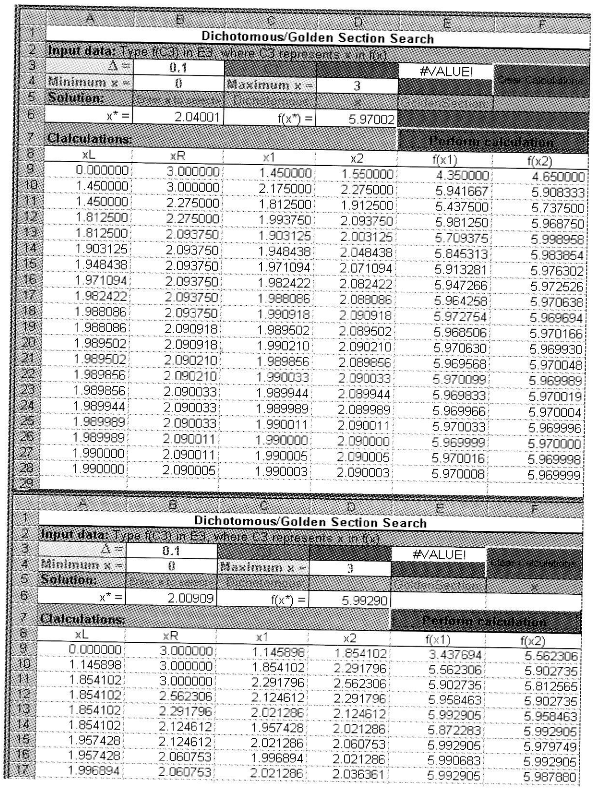
   (a) $f(x) = \dfrac{1}{|(x - 3)^3|}, \quad 2 \le x \le 4$

   (b) $f(x) = x\cos x, \quad 0 \le x \le \pi$

   (c) $f(x) = x\sin \pi x, \quad 1.5 \le x \le 2.5$

   (d) $f(x) = -(x - 3)^2, \quad 2 \le x \le 4$

   (e) $f(x) = \begin{cases} 4x, & 0 \le x \le 2 \\ 4 - x, & 2 \le x \le 4 \end{cases}$
3. Develop an expression for determining the maximum number of iterations needed to terminate the dichotomous search method for a given value of Δ and an initial interval of uncertainty $I_0 = b - a$.



	A	B	C	D	E	F
1	Dichotomous/Golden Section Search					
2	**Input data:** Type f(C3) in E3, where C3 represents x in f(x)					
3	Δ =	0.1			#VALUE!	Clear Calculations
4	Minimum x =	0	Maximum x =	3		
5	**Solution:**	Enter x to select→	Dichotomous	x	GoldenSection	
6	x* =	2.04001	f(x*) =	5.97002		
7	**Clalculations:**				Perform calculation	
8	xL	xR	x1	x2	f(x1)	f(x2)
9	0.000000	3.000000	1.450000	1.550000	4.350000	4.650000
10	1.450000	3.000000	2.175000	2.275000	5.941667	5.908333
11	1.450000	2.275000	1.812500	1.912500	5.437500	5.737500
12	1.812500	2.275000	1.993750	2.093750	5.981250	5.968750
13	1.812500	2.093750	1.903125	2.003125	5.709375	5.998958
14	1.903125	2.093750	1.948438	2.048438	5.845313	5.983854
15	1.948438	2.093750	1.971094	2.071094	5.913281	5.976302
16	1.971094	2.093750	1.982422	2.082422	5.947266	5.972526
17	1.982422	2.093750	1.988086	2.088086	5.964258	5.970638
18	1.988086	2.093750	1.990918	2.090918	5.972754	5.969694
19	1.988086	2.090918	1.989502	2.089502	5.968506	5.970166
20	1.989502	2.090918	1.990210	2.090210	5.970630	5.969930
21	1.989502	2.090210	1.989856	2.089856	5.969568	5.970048
22	1.989856	2.090210	1.990033	2.090033	5.970099	5.969989
23	1.989856	2.090033	1.989944	2.089944	5.969833	5.970019
24	1.989944	2.090033	1.989989	2.089989	5.969966	5.970004
25	1.989989	2.090033	1.990011	2.090011	5.970033	5.969996
26	1.989989	2.090011	1.990000	2.090000	5.969999	5.970000
27	1.990000	2.090011	1.990005	2.090005	5.970016	5.969998
28	1.990000	2.090005	1.990003	2.090003	5.970008	5.969999
29						

	A	B	C	D	E	F
1	Dichotomous/Golden Section Search					
2	**Input data:** Type f(C3) in E3, where C3 represents x in f(x)					
3	Δ =	0.1			#VALUE!	Clear Calculations
4	Minimum x =	0	Maximum x =	3		
5	**Solution:**	Enter x to select→	Dichotomous		GoldenSection	x
6	x* =	2.00909	f(x*) =	5.99290		
7	**Clalculations:**				Perform calculation	
8	xL	xR	x1	x2	f(x1)	f(x2)
9	0.000000	3.000000	1.145898	1.854102	3.437694	5.562306
10	1.145898	3.000000	1.854102	2.291796	5.562306	5.902735
11	1.854102	3.000000	2.291796	2.562306	5.902735	5.812565
12	1.854102	2.562306	2.124612	2.291796	5.958463	5.902735
13	1.854102	2.291796	2.021286	2.124612	5.992905	5.958463
14	1.854102	2.124612	1.957428	2.021286	5.872283	5.992905
15	1.957428	2.124612	2.021286	2.060753	5.992905	5.979749
16	1.957428	2.060753	1.996894	2.021286	5.990683	5.992905
17	1.996894	2.060753	2.021286	2.036361	5.992905	5.987880

FIGURE 21.2

Excel output of the dichotomous and golden section methods applied to Example 21.1-1

21.1.2 Gradient Method

This section develops a method for optimizing functions that are twice continuously differentiable. The idea is to generate successive points in the direction of the gradient of the function.

The Newton-Raphson method presented in Section 20.1.2 is a gradient method for solving simultaneous equations. This section presents another technique, called the **steepest ascent** method.

Termination of the gradient method occurs at the point where the gradient vector becomes null. This is only a necessary condition for optimality. Optimality cannot be verified unless it is known a priori that $f(\mathbf{X})$ is concave or convex.

Suppose that $f(\mathbf{X})$ is maximized. Let $\mathbf{X}^0$ be the initial point from which the procedure starts and define $\nabla f(\mathbf{X}^k)$ as the gradient of f at point $\mathbf{X}^k$. The idea is to determine a particular path p along which $\frac{\partial f}{\partial p}$ is maximized at a given point. This result is achieved if successive points $\mathbf{X}^k$ and $\mathbf{X}^{k+1}$ are selected such that

$$\mathbf{X}^{k+1} = \mathbf{X}^k + r^k \nabla f(\mathbf{X}^k)$$

where r^k is the optimal **step size** at $\mathbf{X}^k$.

The step size r^k is determined such that the next point, $\mathbf{X}^{k+1}$, leads to the largest improvement in f. This is equivalent to determining $r = r^k$ that maximizes the function

$$h(r) = f[\mathbf{X}^k + r\nabla f(\mathbf{X}^k)]$$

Because $h(r)$ is a single-variable function, the search method in Section 21.1.1 may be used to find the optimum, provided that $h(r)$ is strictly unimodal.

The proposed procedure terminates when two successive trial points $\mathbf{X}^k$ and $\mathbf{X}^{k+1}$ are approximately equal. This is equivalent to having

$$r^k\nabla f(\mathbf{X}^k) \approx \mathbf{0}$$

Because $r^k \neq 0$, the necessary condition $\nabla f(\mathbf{X}^k) = \mathbf{0}$ is satisfied at $\mathbf{X}^k$.

Example 21.1-2

Consider the following problem:

$$\text{Maximize } f(x_1, x_2) = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

The exact optimum occurs at $(x_1^*, x_2^*) = (\frac{1}{3}, \frac{4}{3})$.

To solve the problem by the steepest ascent method, consider

$$\nabla f(\mathbf{X}) = (4 - 4x_1 - 2x_2, 6 - 2x_1 - 4x_2)$$

The quadratic nature of the function dictates that the gradients at any two successive points are orthogonal (perpendicular to one another). Using the initial point $\mathbf{X}^0 = (1, 1)$, Figure 21.3 shows the successive solution points.

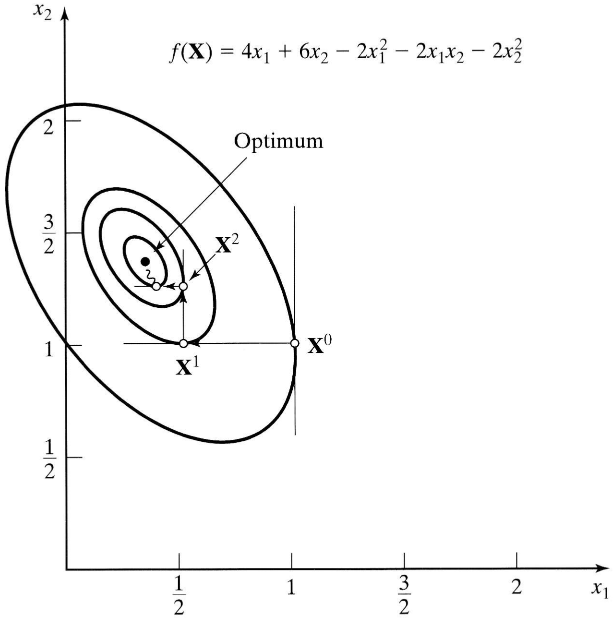


FIGURE 21.3

Maximization of $f(x_1, x_2) = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$ by the steepest ascent method

Iteration 1.

$$\nabla f(\mathbf{X}^0) = (-2, 0)$$

The next point $\mathbf{X}^1$ is obtained by considering

$$\mathbf{X} = (1, 1) + r(-2, 0) = (1 - 2r, 1)$$

Thus,

$$h(r) = f(1 - 2r, 1) = -2(1 - 2r)^2 + 2(1 - 2r) + 4$$

The optimal step size is obtained using the classical necessary conditions in Chapter 20 (you may also use the search algorithms in Section 21.1.1 to determine the optimum). The maximum value of $h(r)$ is $r^1 = \frac{1}{4}$, which yields the next solution point as $\mathbf{X}^1 = (\frac{1}{2}, 1)$.

Iteration 2.

$$\nabla f(\mathbf{X}^1) = (0, 1)$$

$$\mathbf{X} = (\tfrac{1}{2}, 1) + r(0, 1) = (\tfrac{1}{2}, 1 + r)$$

$$h(r) = -2(1 + r)^2 + 5(1 + r) + \tfrac{3}{2}$$

This gives $r^2 = \frac{1}{4}$ and $\mathbf{X}^2 = (\frac{1}{2}, \frac{5}{4})$.

Iteration 3.

$$\nabla f(\mathbf{X}^2) = (-\tfrac{1}{2}, 0)$$

$$\mathbf{X} = (\tfrac{1}{2}, \tfrac{5}{4}) + r(-\tfrac{1}{2}, 0) = (\tfrac{1 - r}{2}, \tfrac{5}{4})$$

$$h(r) = -\tfrac{1}{2}(1 - r)^2 + \tfrac{3}{4}(1 - r) + \tfrac{35}{8}$$

Hence, $r^3 = \frac{1}{4}$ and $\mathbf{X}^3 = (\frac{3}{8}, \frac{5}{4})$.

Iteration 4.

$$\nabla f(\mathbf{X}^3) = (0, \tfrac{1}{4})$$

$$\mathbf{X} = (\tfrac{3}{8}, \tfrac{5}{4}) + r(0, \tfrac{1}{4}) = (\tfrac{3}{8}, \tfrac{5 + r}{4})$$

$$h(r) = -\tfrac{1}{8}(5 + r)^2 + \tfrac{21}{16}(5 + r) + \tfrac{39}{32}$$

Thus, $r^4 = \frac{1}{4}$ and $\mathbf{X}^4 = (\frac{3}{8}, \frac{21}{16})$.

Iteration 5.

$$\nabla f(\mathbf{X}^4) = (-\tfrac{1}{8}, 0)$$

$$\mathbf{X} = (\tfrac{3}{8}, \tfrac{21}{16}) + r(-\tfrac{1}{8}, 0) = (\tfrac{3 - r}{8}, \tfrac{21}{16})$$

$$h(r) = -\tfrac{1}{32}(3 - r)^2 + \tfrac{11}{64}(3 - r) + \tfrac{567}{128}$$

This gives $r^5 = \frac{1}{4}$ and $\mathbf{X}^5 = (\frac{11}{32}, \frac{21}{16})$.

Iteration 6.

$$\nabla f(\mathbf{X}^5) = (0, \tfrac{1}{16})$$

Because $\nabla f(\mathbf{X}^5) \approx \mathbf{0}$, the process can be terminated at this point. The *approximate* maximum point is given by $\mathbf{X}^5 = (.3438, 1.3125)$. The exact optimum is $\mathbf{X}^* = (.3333, 1.3333)$.

PROBLEM SET 21.1B

1. Show that, in general, the Newton-Raphson method (Section 20.1.2) when applied to a strictly concave quadratic function will converge in exactly one step. Apply the method to the maximization of

$$f(\mathbf{X}) = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

2. Carry out at most five iterations for each of the following problems using the method of steepest descent (ascent). Assume that $\mathbf{X}^0 = 0$ in each case.

(a) $\min f(\mathbf{X}) = (x_2 - x_1^2)^2 + (1 - x_1)$

(b) $\max f(\mathbf{X}) = \mathbf{cX} + \mathbf{X}^T\mathbf{AX}$

where

$$\mathbf{c} = (1, 3, 5)$$

$$\mathbf{A} = \begin{pmatrix} -5 & -3 & -\frac{1}{2} \\ -3 & -2 & 0 \\ -\frac{1}{2} & 0 & -\frac{1}{2} \end{pmatrix}$$

(c) $\min f(\mathbf{X}) = x_1 - x_2 + x_1^2 - x_1x_2$

21.2 CONSTRAINED ALGORITHMS

The general constrained nonlinear programming problem is defined as

$$\text{Maximize (or minimize) } z = f(\mathbf{X})$$

subject to

$$\mathbf{g}(\mathbf{X}) \leq \mathbf{0}$$

The nonnegativity conditions, $\mathbf{X} \geq \mathbf{0}$, form part of the constraints. Also, at least one of the functions $f(\mathbf{X})$ and $\mathbf{g}(\mathbf{X})$ is nonlinear, and all the functions are continuously differentiable.

No general algorithm exists for handling nonlinear models, because of the erratic behavior of the nonlinear functions. Perhaps the most general result applicable to the problem is the KKT conditions (Section 20.2.2). Table 20.2 shows that unless $f(\mathbf{X})$ and $\mathbf{g}(\mathbf{X})$ are well-behaved (convexity and concavity conditions), the KKT conditions are only necessary for realizing optimality.

This section presents a number of algorithms that may be classified generally as *indirect* and *direct* methods. Indirect methods solve the nonlinear problem by dealing with one or more *linear* programs derived from the original program. Direct methods deal with the problem in its original form.

The indirect methods presented in this section include separable, quadratic, geometric, and stochastic programming. The direct methods include the method of linear combinations and a brief discussion of the sequential unconstrained maximization technique. Other important nonlinear techniques can be found in the Selected References at the end of the chapter.

21.2.1 Separable Programming

A function $f(x_1, x_2, \ldots, x_n)$ is **separable** if it can be expressed as the sum of n single-variable functions $f_1(x_1), f_2(x_2), \ldots, f_n(x_n)$—that is,

$$f(x_1, x_1 \ldots, x_n) = f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n)$$

For example, the linear function

$$h(x_1, x_2, \ldots, x_n) = a_1x_1 + a_2x_2 + \ldots + a_nx_n$$

is separable (the parameters a_i, $i = 1, 2, \ldots, n$, are constants). Conversely, the function

$$h(x_1, x_2, x_3) = x_1^2 + x_1 \sin(x_2 + x_3) + x_2e^{x3}$$

is not separable.

Some nonlinear functions are not directly separable but can be made so by appropriate substitutions. Consider, for example, the case of maximizing $z = x_1x_2$. Letting $y = x_1x_2$, then $\ln y = \ln x_1 + \ln x_2$ and the problem becomes

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

subject to

$$\ln y = \ln x_1 + \ln x_2$$

which is separable. The substitution assumes that x_1 and x_2 are *positive* variables; otherwise, the logarithmic function is undefined.

The case where x_1 and x_2 assume zero values (i.e., $x_1, x_2 \geq 0$) may be handled in the following manner. Let δ_1 and δ_2 be positive constants and define

$$w_1 = x_1 + \delta_1$$

$$w_2 = x_2 + \delta_2$$

The new variables w_1 and w_2 are strictly positive. Now

$$x_1x_2 = w_1w_2 - \delta_2w_1 - \delta_1w_2 + \delta_1\delta_2$$

Letting $y = w_1w_2$, the problem is expressed as

$$\text{Maximize } z = y - \delta_2w_1 - \delta_1w_2 + \delta_1\delta_2$$

subject to

$$\ln y = \ln w_1 + \ln w_2$$

$$y \geq 0,\ w_1 \geq \delta_1,\ w_2 \geq \delta_2$$

The new problem is separable.

Examples of other functions that can be made separable using substitution are $e^{x_1+x_2}$ and $x_1^{x_2}$. A variant of the procedure just presented can be applied to such cases to effect separability.

Separable programming deals with nonlinear problems in which the objective function and the constraints are separable. This section shows how an approximate solution can be obtained for *any* separable problem by linear approximation and the simplex method of linear programming.

The single-variable function $f(\mathrm{x})$ can be approximated by a piecewise linear function using mixed integer programming (Chapter 9). Suppose that $f(x)$ is to be approximated over an interval $[a, b]$. Define a_k, $k = 1, 2, \ldots, K$, as the kth breaking point on the x-axis such that $a_1 < a_2 < \ldots < a_K$. The points a_1 and a_K coincide with end points a and b of the interval under investigation. Thus, $f(\mathrm{x})$ is approximated as follows:

$$f(x) \approx \sum_{k=1}^{K} f(a_k)t_k$$

$$x = \sum_{k=1}^{K} a_k t_k$$

where t_k is a nonnegative weight associated with the kth breaking point such that

$$\sum_{k=1}^{K} t_k = 1$$

Mixed integer programming ensures the validity of the approximation. Specifically, the piecewise linear approximation is valid if

1. At most two t_k are positive.
2. If t_k is a positive, then only an adjacent t_{k+1} or t_{k-1} can assume a positive value.

To show how these conditions are satisfied, consider the separable problem

$$\text{Maximize (or minimize) } z = \sum_{i=1}^{n} f_i(x_i)$$

subject to

$$\sum_{i=1}^{n} g_i^j(x_i) \le b_j,\ j = 1, 2, \ldots, m$$

This problem can be approximated by a mixed integer program as follows. Let the number of breaking points for the ith variable x_i equal K_i and let a_i^k be its kth breaking value. Let t_i^k be the weight associated with the kth breaking point of variable i.[1] Then the equivalent mixed problem is

$$\text{Maximize (or minimize) } z = \sum_{i=1}^{n} \sum_{k=1}^{K_i} f_i(a_i^k)t_i^k$$

subject to

$$\sum_{i=1}^{n} \sum_{k=1}^{K_i} g_i^j(a_i^k)t_i^k \le b_j, \quad j = 1, 2, \ldots, m$$

$$0 \le t_i^1 \le y_i^1,\ i = 1, 2, \ldots, n$$

$$0 \le t_i^k \le y_i^{k-1} + y_i^k, \quad k = 2, 3, \ldots, K_i - 1$$

[1] It would be more accurate to replace the index k with k_i to correspond uniquely to variable i. However, we will not do so for the sake of simplifying the notation.

$$0 \leq t_i^{K_i} \leq y_i^{K_i - 1}, \; i = 1, 2, \ldots, n$$

$$\sum_{k=1}^{K_i - 1} y_i^k = 1$$

$$\sum_{k=1}^{K_i} t_i^k = 1$$

$$y_i^k = (0, 1), \; k = 1, 2, \ldots, K_i, \; i = 1, 2, \ldots, n$$

The variables for the approximating problem are t_i^k and y_i^k.

This formulation shows how any separable problem can be solved, at least in principle, by mixed integer programming. The difficulty is that the number of constraints increases rather rapidly with the number of breaking points. In particular, the computational feasibility of the procedure is questionable because there are no reliable computer codes for solving large mixed integer programming problems.

Another method for solving the approximate model is the regular simplex method (Chapter 3) using **restricted basis**. In this case the additional constraints involving y_i^k are not needed. The restricted basis specifies that *no more* than two *positive* t_i^k can appear in the basis. Moreover, two t_i^k can be positive only if they are adjacent. Thus, the strict optimality condition of the simplex method is used to select the entering variable t_i^k *only if* it satisfies the foregoing conditions. Otherwise, the variable t_i^k having the next best optimality indicator $(z_i^k - c_i^k)$ is considered for entering the solution. The process is repeated until the optimality condition is satisfied or until it is impossible to introduce new t_i^k without violating the restricted basis condition, whichever occurs first. The last tableau gives the approximate optimal solution to the problem.

The mixed integer programming method yields a global optimum to the approximate problem, but the restricted basis method can only guarantee a local optimum. Additionally, in the two methods, the approximate solution may not be feasible for the original problem. In fact, the approximate model may give rise to additional extreme points that are not part of the solution space of the original problem.

Example 21.2-1

Consider the problem

$$\text{Maximize} \quad z = x_1 + x_2^4$$

subject to

$$3x_1 + 2x_2^2 \leq 9$$

$$x_1, x_2 \geq 0$$

The exact optimum solution to this problem, obtained by inspection, is $x_1 = 0$, $x_2 = 2.12$, and $z^* = 20.2$. To show how the approximating method is used, consider the separable functions

$$f_1(x_1) = x_1$$

$$f_2(x_2) = x_2^4$$

$$g_1^1(x_1) = 3x_1$$
$$g_1^2(x_2) = 2x_2^2$$

The functions $f_1(x_1)$ and $g_1^1(x_1)$ are left in their present form because they are already linear. In this case, x_1 is treated as one of the variables. Considering $f_2(x_2)$ and $g_1^2(x_2)$, we assume that there are four breaking points ($K_2 = 4$). Because the value of x_2 cannot exceed 3, it follows that

k	a_2^k	$f_2(a_2^k)$	$g_1^2(a_2^k)$
1	0	0	0
2	1	1	2
3	2	16	8
4	3	81	18

This yields

$$\begin{aligned} f_2(x_2) &\approx t_2^1 f_2(a_2^1) + t_2^2 f_2(a_2^2) + t_2^3 f_2(a_2^3) + t_2^4 f_2(a_2^4) \\ &\approx 0t_2^1 + 1t_2^2 + 16t_2^3 + 81t_2^4 = t_2^2 + 16t_2^3 + 81t_2^4 \end{aligned}$$

Similarly,

$$g_1^2(x_2) \approx 2t_2^2 + 8t_2^3 + 18t_2^4$$

The approximating problem thus becomes

$$\text{Maximize } z = x_1 + t_2^2 + 16t_2^3 + 81t_2^4$$

subject to

$$\begin{aligned} 3x_1 + 2t_2^2 + 8t_2^3 + 18t_2^4 &\leq 9 \\ t_2^1 + t_2^2 + t_2^3 + t_2^4 &= 1 \\ t_2^k \geq 0,\ k &= 1, 2, 3, 4 \\ x_1 &\geq 0 \end{aligned}$$

The solution must satisfy the restricted basis condition.

The initial simplex tableau (with rearranged columns to give a starting solution) is given by

Basic	x_1	t_2^2	t_2^3	t_2^4	s_1	t_2^1	Solution
z	-1	-1	-16	-81	0	0	0
s_1	3	2	8	18	1	0	9
t_2^1	0	1	1	1	0	1	1

The variable $s_1(\geq 0)$ is a slack. (This problem happened to have an obvious starting solution. In general, one may have to use the artificial variables techniques, Section 3.4.)

From the z-row coefficients, t_2^4 is the entering variable. Because t_2^1 is currently basic at positive level, the restricted basis condition dictates that it must leave before t_2^4 can enter the solution. By the feasibility condition, s_1 must be the leaving variable. This means that t_2^4 cannot enter the solution. The next best entering variable, t_2^3, requires t_2^1

to leave the basic solution, a condition that happens to be guaranteed by the feasibility condition. The new tableau thus becomes

Basic	x_1	t_2^2	t_2^3	t_2^4	s_1	t_2^1	Solution
z	-1	15	0	-65	0	16	16
s_1	3	-6	0	10	1	-8	1
t_2^3	0	1	1	1	0	1	1

Next, t_2^4 is the entering variable, which is admissible because t_2^3 is positive. The simplex method shows that s_1 will leave. Thus,

Basic	x_1	t_2^2	t_2^3	t_2^4	s_1	t_2^1	Solution
Z	$\frac{37}{2}$	-24	0	0	$\frac{13}{2}$	-36	$22\frac{1}{2}$
t_2^4	$\frac{3}{10}$	$-\frac{6}{10}$	0	1	$\frac{1}{10}$	$-\frac{8}{10}$	$\frac{1}{10}$
t_2^3	$-\frac{3}{10}$	$\frac{16}{10}$	1	0	$-\frac{1}{10}$	$\frac{18}{10}$	$\frac{9}{10}$

The tableau shows that t_2^1 and t_2^2 are candidates for the entering variable. Because t_2^1 is not adjacent to basic t_2^3 or t_2^4, it cannot enter. Similarly, t_2^2 cannot enter because t_2^4 cannot leave. The process ends at this point, and the solution given is the best feasible solution for the approximate problem.

The optimum solution to the original problem is

$$x_1 = 0$$

$$x_2 \approx 2t_2^3 + 3t_2^4 = 2(\tfrac{9}{10}) + 3(\tfrac{1}{10}) = 2.1$$

$$z = 0 + 2.1^4 = 19.45$$

The approximate optimum value of $x_2(= 2.1)$ approximately equals the true optimum value $(= 2.12)$.

Separable Convex Programming. A special case of separable programming occurs when $g_i^j(x_i)$ is convex for all i and j, thus ensuring a convex solution space. Additionally, if $f_i(x_i)$ is convex (minimization) or concave (maximization) for all i, then the problem has a global optimum (see Table 20.2, Section 20.2.2). Under such conditions, a simplified approximation can be used.

Consider a minimization problem and let $f_i(x_i)$ be as shown in Figure 21.4. The breaking points of the function $f_i(x_i)$ are $x_i = a_{ki}$, $k = 0, 1, \ldots, K_i$. Let x_{ki} define the increment of the variable x_i in the range $(a_{k-1,i}, a_{ki})$, $k = 1, 2, \ldots, K_i$; and let ρ_{ki} be the corresponding slope of the line segment in the same range. Then

$$f_i(x_i) \approx \sum_{k=1}^{K_i} \rho_{ki} x_{ki} + f_i(a_{0i})$$

$$x_i = \sum_{k=1}^{K_i} x_{ki}$$

$$0 \le x_{ki} \le a_{ki} - a_{k-1,\,i},\ k = 1, 2, \ldots, K_i$$

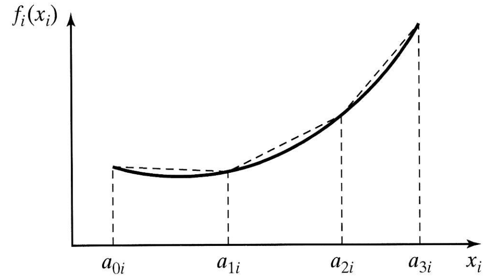


FIGURE 21.4

Piecewise linear approximation of a convex function

The fact that $f_i(x_i)$ is convex ensures that $\rho_{1i} < \rho_{2i} < \cdots < \rho_{K_1 i}$. Thus, in a minimization problem, for $p < q$, the variable x_{pi} is more attractive than x_{qi}, which means that x_{pi} will always enter the solution before x_{qi}.

The convex constraint functions $g_i^j(x_i)$ are approximated in essentially the same way. Let ρ_{ki}^j be the slope of the kth line segment corresponding to $g_i^j(x_i)$. It follows that the ith function is approximated as

$$g_i^j(x_i) \approx \sum_{k=1}^{K_i} \rho_{ki}^j x_{ki} + g_i^j(a_{0i})$$

The complete problem is thus given by

$$\text{Minimize } z = \sum_{i=1}^{n}\left(\sum_{k=1}^{K_1} \rho_{ki} x_{ki} + f_i(a_{0i})\right)$$

subject to

$$\sum_{i=1}^{n}\left(\sum_{k=1}^{K_i} \rho_{ki}^j x_{ki} + g_i^j(a_{0i})\right) \le b_j, \quad j = 1, 2, \ldots, m$$

$$0 \le x_{ki} \le a_{ki} - a_{k-1,\, i},\ k = 1, 2, \ldots, K_i;\ i = 1, 2, \ldots, n$$

where

$$\rho_{ki} = \frac{f_i(a_{ki}) - f_i(a_{k-1,i})}{a_{ki} - a_{k-1,i}}$$

$$\rho_{ki}^j = \frac{g_i^j(a_{ki}) - g_i^j(a_{k-1,i})}{a_{ki} - a_{k-1,i}}$$

The maximization problem is treated essentially the same way. In this case, $\rho_{1i} > \rho_{2i} > \cdots > \rho_{K_i i}$, which means that, for $p < q$, the variable x_{pi} will always enter the solution before x_{qi} (see Problem 7, Set 21.2a for proof).

The new problem can be solved by the simplex method with upper bounded variables (Section 7.3). The restricted basis concept is not needed because the convexity (concavity) of the functions guarantees correct selection of basic variables.

Example 21.2-2

Consider the problem

$$\text{Maximize } z = x_1 - x_2$$

subject to

$$\begin{aligned} 3x_1^4 + x_2 &\le 243 \\ x_1 + 2x_2^2 &\le 32 \\ x_1 \qquad &\ge 2.1 \\ x_2 &\ge 3.5 \end{aligned}$$

The separable functions of this problem are

$$\begin{aligned} f_1(x_1) &= x_1, \quad f_2(x_2) = -x_2 \\ g_1^1(x_1) &= 3x_1^4, \quad g_2^1(x_2) = x_2 \\ g_1^2(x_1) &= x_1, \quad g_2^2(x_2) = 2x_2^2 \end{aligned}$$

These functions satisfy the convexity condition required for the minimization problem.

The ranges of the variables x_1 and x_2 (estimated from the constraints) are $0 \le x_1 \le 3$ and $0 \le x_2 \le 4$. Let $K_1 = 3$ and $K_2 = 4$ with $a_{01} = a_{02} = 0$. The slopes corresponding to the separable functions are determined as follows.

For $i = 1$,

k	a_{k1}	$f_1(a_{k1}) = a_{k1}$	ρ_{k1}	$g_1^1(a_{k1}) = 3a_{k1}^4$	ρ_{k1}^1	$g_1^2(a_{k1}) = a_{k1}$	ρ_{k1}^2	x_{k1}
0	0	0	—	0	—	0	—	—
1	1	1	1	3	3	1	1	x_{11}
2	2	2	1	48	45	2	1	x_{21}
3	3	3	1	243	195	3	1	x_{31}

For $i = 2$,

k	a_{k2}	$f_2(a_{k2}) = -a_{k2}$	ρ_{k2}	$g_2^1(a_{k2}) = a_{k2}$	ρ_{k2}^1	$g_2^2(a_{k2}) = 2a_{k2}^2$	ρ_{k2}^2	x_{k1}
0	0	0	—	0	—	0	—	—
1	1	−1	−1	1	1	2	2	x_{11}
2	2	−2	−1	2	1	8	6	x_{21}
3	3	−3	−1	3	1	18	10	x_{31}
4	4	−4	−1	4	1	32	14	x_{42}

The complete problem then becomes

$$\text{Maximize } z \approx x_{11} + x_{21} + x_{31} - x_{12} - x_{22} - x_{32} - x_{42}$$

subject to

$$\begin{aligned} 3x_{11} + 45x_{21} + 195x_{31} + x_{12} + x_{22} + x_{32} + x_{42} &\le 243 \\ x_{11} + x_{21} + x_{31} + 2x_{12} + 6x_{22} + 10x_{32} + 14x_{42} &\le 32 \\ x_{11} + x_{21} + x_{31} &\ge 2.1 \\ x_{12} + x_{22} + x_{32} + x_{42} &\ge 3.5 \\ 0 \le x_{k1} \le 1, \quad k &= 1,2,3 \\ 0 \le x_{k2} \le 1, \quad k &= 1,2,3,4 \end{aligned}$$

TORA optimum solution is

$$z = -.52,\ x_{11} = x_{12} = 1,\quad x_{13} = .98,\ x_{21} = x_{22} = x_{23} = 1,\ x_{24} = .5$$

The solution translates to $(x_1, x_2) = (2.98, 3.5)$.

PROBLEM SET 21.2A

1. Approximate the following problem as a mixed integer program.

$$\text{Maximize } z = e^{-x_1} + x_1 + (x_2 + 1)^2$$

subject to

$$x_1^2 + x_2 \leq 3$$

$$x_1, x_2 \geq 0$$

2. Repeat Problem 1 using the restricted basis method. Then find the optimal solution.

3. Consider the problem

$$\text{Maximize } z = x_1 x_2 x_3$$

subject to

$$x_1^2 + x_2 + x_3 \leq 4$$

$$x_1, x_2, x_3 \geq 0$$

Approximate the problem as a linear program for use with the restricted basis method.

4. Show how the following problem can be made separable.

$$\text{Maximize } z = x_1 x_2 + x_3 + x_1 x_3$$

subject to

$$x_1 x_2 + x_2 + x_1 x_3 \leq 10$$

$$x_1, x_2, x_3 \geq 0$$

5. Show how the following problem can be made separable.

$$\text{Minimize } z = e^{2x_1 + x_2^2} + (x_3 - 2)^2$$

subject to

$$x_1 + x_2 + x_3 \leq 6$$

$$x_1, x_2, x_3 \geq 0$$

6. Show how the following problem can be made separable.

$$\text{Maximize } z = e^{x_1 x_2} + x_2^2 x_3 + x_4$$

subject to

$$x_1 + x_2 x_3 + x_3 \leq 10$$

$$x_1, x_2, x_3 \geq 0$$

$$x_4 \text{ unrestricted in sign}$$

7. Show that in separable convex programming, it is never optimal to have $x_{ki} > 0$ when $x_{k-1,i}$ is not at its upper bound.

8. Solve as a separable convex programming problem.

$$\text{Minimize } z = x_1^4 + 2x_2 + x_3^2$$

subject to

$$x_1^2 + x_2 + x_3^2 \leq 4$$
$$|x_1 + x_2| \leq 0$$
$$x_1, x_3 \geq 0$$
$$x_2 \text{ unrestricted in sign}$$

9. Solve the following as a separate convex programming problem.

$$\text{Minimize } z = (x_1 - 2)^2 + 4(x_2 - 6)^2$$

subject to

$$6x_1 + 3(x_2 + 1)^2 \leq 12$$
$$x_1, x_2 \geq 0$$

21.2.2 Quadratic Programming

A quadratic programming model is defined as

$$\text{Maximize } z = \mathbf{CX} + \mathbf{X}^T\mathbf{DX}$$

subject to

$$\mathbf{AX} \leq \mathbf{b},\ \mathbf{X} \geq \mathbf{0}$$

where

$$\mathbf{X} = (x_1, x_2, \ldots, x_n)^T$$
$$\mathbf{C} = (c_1, c_2, \ldots, c_n)$$
$$\mathbf{b} = (b_1, b_2, \ldots, b_m)^T$$
$$\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$
$$\mathbf{D} = \begin{pmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \vdots & \vdots \\ d_{n1} & \cdots & d_{nn} \end{pmatrix}$$

The function $\mathbf{X}^T\mathbf{DX}$ defines a quadratic from (Section A.3). The matrix $\mathbf{D}$ is assumed symmetric and negative-definite. This means that z is strictly concave. The constraints are linear, which guarantees a convex solution space.

The solution to this problem is based on the KKT necessary conditions. Because z is strictly concave and the solution space is convex, these conditions (as shown in Table 20.2, Section 20.2.2) are also sufficient for a global optimum.

The quadratic programming problem will be treated for the maximization case. It is trivial to change the formulation to minimization. The problem may be written as

$$\text{Maximize } z = \mathbf{CX} + \mathbf{X}^T\mathbf{DX}$$

subject to

$$\mathbf{G}(\mathbf{X}) = \begin{pmatrix} \mathbf{A} \\ -\mathbf{I} \end{pmatrix}\mathbf{X} - \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \leq \mathbf{0}$$

Let

$$\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T$$

$$\mathbf{U} = (\mu_1, \mu_2, \ldots, \mu_n)^T$$

be the Lagrange multipliers corresponding to the two sets of constraints $\mathbf{AX} - \mathbf{b} \leq \mathbf{0}$ and $-\mathbf{X} \leq \mathbf{0}$, respectively. Application of the KKT conditions yields

$$\boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{U} \geq \mathbf{0}$$

$$\nabla z - (\boldsymbol{\lambda}^T, \mathbf{U}^T)\nabla\mathbf{G}(\mathbf{X}) = \mathbf{0}$$

$$\lambda_i\left(b_i - \sum_{j=1}^{n} a_{ij}x_j\right) = 0, \ i = 1, 2, \ldots, m$$

$$\mu_j x_j = 0, \quad j = 1, 2, \ldots, n$$

$$\mathbf{AX} \leq \mathbf{b}$$

$$-\mathbf{X} \leq \mathbf{0}$$

Now

$$\nabla z = \mathbf{C} + 2\mathbf{X}^T\mathbf{D}$$

$$\nabla\mathbf{G}(\mathbf{X}) = \begin{pmatrix} \mathbf{A} \\ -\mathbf{I} \end{pmatrix}$$

Let $\mathbf{S} = \mathbf{b} - \mathbf{AX} \geq \mathbf{0}$ be the slack variables of the constraints. The conditions reduce to

$$-2\mathbf{X}^T\mathbf{D} + \boldsymbol{\lambda}^T\mathbf{A} - \mathbf{U}^T = \mathbf{C}$$

$$\mathbf{AX} + \mathbf{S} = \mathbf{b}$$

$$\mu_j x_j = 0 = \lambda_i S_i \quad \text{for all } i \text{ and } j$$

$$\boldsymbol{\lambda}, \mathbf{U}, \mathbf{X}, \mathbf{S} \geq \mathbf{0}$$

Because $\mathbf{D}^T = \mathbf{D}$, the transpose of the first set of equations can be written as

$$-2\mathbf{DX} + \mathbf{A}^T\boldsymbol{\lambda} - \mathbf{U} = \mathbf{C}^T$$

Hence, the necessary conditions may be combined as

$$\begin{pmatrix} -2\mathbf{D} & \mathbf{A}^T & -\mathbf{I} & \mathbf{0} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix}\begin{pmatrix} \mathbf{X} \\ \boldsymbol{\lambda} \\ \mathbf{U} \\ \mathbf{S} \end{pmatrix} = \begin{pmatrix} \mathbf{C}^T \\ \mathbf{b} \end{pmatrix}$$

$$\mu_j x_j = 0 = \lambda_i S_i, \quad \text{for all } i \text{ and } j$$

$$\boldsymbol{\lambda}, \mathbf{U}, \mathbf{X}, \mathbf{S} \geq \mathbf{0}$$

Except for the conditions $\mu_j x_j = 0 = \lambda_i S_i$, the remaining equations are linear functions in $\mathbf{X}$, $\boldsymbol{\lambda}$, $\mathbf{U}$, and $\mathbf{S}$. The problem is thus equivalent to solving a set of linear equations, under the additional conditions $\mu_j x_j = 0 = \lambda_i S_i$. Because z is strictly concave and the solution space is convex, the *feasible* solution satisfying all these conditions must be unique and optimum.

The solution of the system is obtained by using phase I of the two-phase method (Section 3.4.2). The only restriction is to satisfy the conditions $\lambda_i S_i = 0 = \mu_j x_j$. This means that λ_i and S_i cannot be positive simultaneously. Similarly, μ_j and x_j cannot be positive simultaneously. This is the same idea of the *restricted basis* used in Section 21.2.1.

Phase I will render all the artificial variables equal to zero only if the problem has a feasible space.

Example 21.2-3

Consider the problem

$$\text{Maximize } z = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

subject to

$$x_1 + 2x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

This problem can be put in matrix form as follows:

$$\text{Maximize } z = (4, 6)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (x_1, x_2)\begin{pmatrix} -2 & -1 \\ -1 & -2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

subject to

$$(1, 2)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq 2$$

$$x_1, x_2 \geq 0$$

The Kuhn-Tucker conditions are given as

$$\begin{pmatrix} 4 & 2 & 1 & -1 & 0 & 0 \\ 2 & 4 & 2 & 0 & -1 & 0 \\ 1 & 2 & 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \lambda_1 \\ \mu_1 \\ \mu_2 \\ s_1 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \\ 2 \end{pmatrix}$$

The initial tableau for phase I is obtained by introducing the artificial variables R_1 and R_2. Thus

Basic	x_1	x_2	λ_1	μ_1	μ_2	R_1	R_2	s_1	Solution
r	6	6	3	-1	-1	0	0	0	10
R_1	4	2	1	-1	0	1	0	0	4
R_2	2	4	2	0	-1	0	1	0	6
s_1	1	2	0	0	0	0	0	1	2

Iteration 1. Because $\mu_1 = 0$, the most promising entering variable x_1 can be made basic with R_1 as the leaving variable. This yields the following tableau:

Basic	x_1	x_2	λ_1	μ_1	μ_2	R_1	R_2	s_1	Solution
r	0	3	$\frac{3}{2}$	$\frac{1}{2}$	-1	$-\frac{3}{2}$	0	0	4
x_1	1	$\frac{1}{2}$	$\frac{1}{4}$	$-\frac{1}{4}$	0	$\frac{1}{4}$	0	0	1
R_2	0	3	$\frac{3}{2}$	$\frac{1}{2}$	-1	$-\frac{1}{2}$	1	0	4
s_1	0	$\frac{3}{2}$	$-\frac{1}{4}$	$\frac{1}{4}$	0	$-\frac{1}{4}$	0	1	1

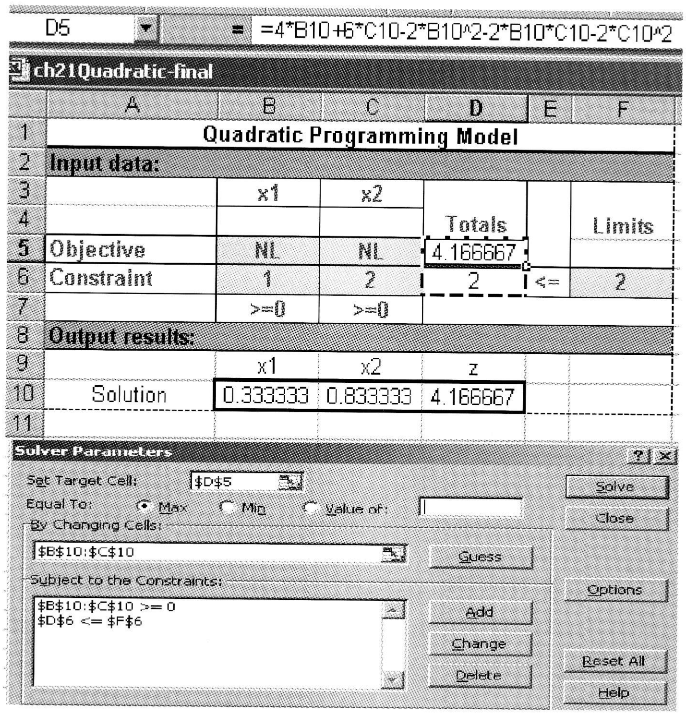
Iteration 2. The most promising variable x_2 can be made basic because $\mu_2 = 0$. This gives

Basic	x_1	x_2	λ_1	μ_1	μ_2	R_1	R_2	s_1	Solution
r	0	0	2	0	-1	-1	0	-2	2
x_1	1	0	$\frac{1}{3}$	$-\frac{1}{3}$	0	$\frac{1}{3}$	0	$-\frac{1}{3}$	$\frac{2}{3}$
R_1	0	0	2	0	-1	0	1	-2	2
x_2	0	1	$-\frac{1}{6}$	$\frac{1}{6}$	0	$-\frac{1}{6}$	0	$\frac{2}{3}$	$\frac{2}{3}$

Iteration 3. Because $s_1 = 0$, λ_1 can be introduced into the solution. This yields

Basic	x_1	x_2	λ_1	μ_1	μ_2	R_1	R_2	s_1	Solution
r	0	0	0	0	0	-1	-1	0	0
x_1	1	0	0	$-\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{3}$	$-\frac{1}{6}$	0	$\frac{1}{3}$
λ_1	0	0	1	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	-1	1
x_2	0	1	0	$\frac{1}{6}$	$-\frac{1}{12}$	$-\frac{1}{6}$	$\frac{1}{12}$	$\frac{1}{2}$	$\frac{5}{6}$

The last tableau gives the optimal solution for phase I. Because $r = 0$, the solution, $x_1 = \frac{1}{3}$, $x_2 = \frac{5}{6}$, is feasible. The optimal value of z is computed from the original problem and is equal to 4.16.



D5 = =4*B10+6*C10-2*B10^2-2*B10*C10-2*C10^2

ch21Quadratic-final

	A	B	C	D	E	F
1	Quadratic Programming Model					
2	Input data:					
3		x1	x2			
4				Totals		Limits
5	Objective	NL	NL	4.166667		
6	Constraint	1	2	2	<=	2
7		>=0	>=0			
8	Output results:					
9		x1	x2	z		
10	Solution	0.333333	0.833333	4.166667		
11						

Solver Parameters

Set Target Cell: $D$5

Equal To: ● Max ○ Min ○ Value of:

By Changing Cells: $B$10:$C$10

Subject to the Constraints:

$B$10:$C$10 >= 0
$D$6 <= $F$6

FIGURE 21.5

Excel solution of the quadratic programming problem of Example 21.2-3

Excel Solver can be used to solve the quadratic programming problem. Figure 21.5 provides the solution for Example 21.2-3 (see file ch21SolverQuadraticProgramming.xls). The data are entered in a manner similar to the one used in linear programming (see Section 2.4.2). The main difference is the way the nonlinear function is entered. Specifically, in Example 21.2-3, the nonlinear objective function

$$z = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

is entered in target cell D5 as

```
=4*B10+6*C10-2*B10^2-2*B10*C10-2*C10^2
```

Here, the changing cells B10 and C10 represent x_1 and x_2. Notice that cells B5:C5 are not used at all in the model. For readability, we entered the symbol NL to indicate that the associated constraint is nonlinear. Also, you can specify the nonnegativity of the variables either in the Options dialogue box or by adding explicit nonnegativity constraints.

PROBLEM SET 21.2B

1. Consider the problem

$$\text{Maximize } z = 6x_1 + 3x_2 - 4x_1x_2 - 2x_1^2 - 3x_2^2$$

subject to

$$x_1 + x_2 \leq 1$$

$$2x_1 + 3x_2 \leq 4$$

$$x_1, x_2 \geq 0$$

Show that z is strictly concave and then solve the problem using the quadratic programming algorithm.

2. Consider the problem:

$$\text{Minimize } z = 2x_1^2 + 2x_2^2 + 3x_3^2 + 2x_1x_2 + 2x_2x_3 + x_1 - 3x_2 - 5x_3$$

subject to

$$x_1 + x_2 + x_3 \geq 1$$

$$3x_1 + 2x_2 + x_3 \leq 6$$

$$x_1, x_2, x_3 \geq 0$$

Show that z is strictly convex and then solve by the quadratic programming algorithm.

21.2.3 Geometric Programming

Geometric programming deals with problems in which the objective and the constraint functions are of the following type:

$$z = f(\mathbf{X}) = \sum_{j=1}^{N} U_j$$

where

$$U_j = c_j \prod_{i=1}^{n} x_i^{a_{ij}}, \quad j = 1, 2, \ldots, N$$

It is assumed that all $c_j > 0$, and that N is finite. The exponents a_{ij} are unrestricted in sign. The function $f(\mathbf{X})$ takes the form of a polynomial except that the exponents a_{ij} may be negative. For this reason, and because all $c_j > 0$, $f(\mathbf{X})$ is called a **posynomial.**

This section will present the unconstrained case of geometric programming. The treatment of the constrained problem is beyond the scope of this chapter. Detailed treatment of the subject is given in Beightler and associates (1979, Chap. 6).

Consider the *minimization* of the posynomial function $f(\mathbf{X})$. This problem will be referred to as the *primal.* The variables x_i are assumed *strictly positive* so that the region $x_i \leq 0$ is infeasible. It will be shown later that the requirement $x_i > 0$ plays an essential part in the derivation of the results.

The first partial derivative of z must vanish at a minimum point. Thus,

$$\frac{\partial z}{\partial x_k} = \sum_{j=1}^{N} \frac{\partial U_j}{\partial x_k} = \sum_{j=1}^{n} c_j a_{kj} (x_k)^{a_{kj}-1} \prod_{i \neq k} x_i^{a_{ij}} = 0, \quad k = 1, 2, \ldots, n$$

Because each $x_k > 0$ by assumption,

$$\frac{\partial z}{\partial x_k} = 0 = \frac{1}{x_k} \sum_{j=1}^{N} a_{kj} U_j, \quad k = 1, 2, \ldots, n$$

Let z^* be the minimum value of z. It follows that $z^* > 0$ because z is posynomial and each $x_k > 0$. Define

$$y_j = \frac{U_j^*}{z^*}$$

Thus $y_j > 0$ and $\sum_j^n y_j = 1$. The value of y_j represents the relative contribution of U_j to the optimal value of the objective function z^*. The *necessary* conditions can now be written as

$$\sum_{j=1}^{N} a_{kj} y_j = 0, \quad k = 1, 2, \ldots, n$$

$$\sum_{j=1}^{N} y_j = 1, \qquad y_j > 0 \text{ for all } j$$

These are known as the **orthogonality** and **normality conditions** and will yield a unique solution for y_j if $n + 1 = N$ and all the equations are independent. The problem becomes more complex when $N > n + 1$ because the values of y_j are no longer unique. It is shown later that, even in this case, optimum y_j is unique.

Given y_j^*, the values of z^* and x_i^* can be determined as follows:

$$z^* = (z^*)^{\sum_{j=1}^{N} y_j^*}$$

Because $z^* = \frac{U_j^*}{y_j^*}$, it follows that

$$\begin{aligned} z^* &= \left(\frac{U_1^*}{y_1^*}\right)^{y_1^*}\left(\frac{U_2^*}{y_2^*}\right)^{y_2^*} \cdots \left(\frac{U_N^*}{y_N^*}\right)^{y_N^*} \\ &= \left\{\prod_{j=1}^{N}\left(\frac{c_j}{y_j^*}\right)^{y_j^*}\right\}\left\{\prod_{j=1}^{N}\left(\prod_{i=1}^{n}(x_i)^{a_{ij}}\right)^{y_j^*}\right\} \\ &= \left\{\prod_{j=1}^{N}\left(\frac{c_j}{y_j^*}\right)^{y_j^*}\right\}\left\{\prod_{i=1}^{n}(x_i^*)^{\sum_{j=1}^{N} a_{ij} y_j^*}\right\} \\ &= \prod_{j=1}^{N}\left(\frac{c_j}{y_j^*}\right)^{y_j^*} \end{aligned}$$

This step is justified because $\sum_{j=1}^{N} a_{ij} y_j = 0$. The value of z^* is known once all y_j have been determined. Now, given y_j^* and z^*, $U_j^* = y_j^* z^*$ can be determined. Solution of the following equations then yields x_i^*.

$$U_j^* = c_j \prod_{i=1}^{n} (x_i^*)^{a_{ij}}, \quad j = 1, 2, \ldots, N$$

The procedure shows that the solution to the original posynomial z can be transformed into the solution of a set of linear equations in y_j. These equations are the necessary conditions for a minimum. It can be shown that these conditions are also sufficient. The proof is given in Beightler and associates (1979, p. 333).

The variables y_j actually define the dual variables associated with the z-primal problem. To see this relationship, consider the primal problem in the form

$$z = \sum_{j=1}^{N} y_j \left(\frac{U_j}{y_j} \right)$$

Now define the dual function

$$w = \prod_{j=1}^{N} \left(\frac{U_j}{y_j} \right)^{y_j}$$

Because $\sum_{j=1}^{N} y_j = 1$ and $y_j > 0$, we have

$$w \leq z$$

This result is based on **Cauchy's arithmetic-geometric inequality**, which states that

$$\sum_{j=1}^{N} w_j z_j \geq \prod_{j=1}^{N} z_j^{w_j}$$

$$w_j > 0, \quad \sum_{j=1}^{N} w_j = 1$$

An immediate consequence of the inequality $w \leq z$ is the following relationship:

$$w^* = \max_{y_j} w = \min_{x_i} z = z^*$$

Example 21.2-4

In this example a problem is considered in which $N = n + 1$ so that the solution to the orthogonality and normality conditions is unique. The next example illustrates the case where $N > n + 1$.

Consider the problem

$$\text{Minimize } z = 7x_1x_2^{-1} + 3x_2x_3^{-2} + 5x_1^{-3}x_2x_3 + x_1x_2x_3$$

This function may be written as

$$\text{Minimize } z = 7x_1^1x_2^{-1}x_3^0 + 3x_1^0x_2^1x_3^{-2} + 5x_1^{-3}x_2^1x_3^1 + x_1^1x_2^1x_3^1$$

Thus,

$$(c_1, c_2, c_3, c_4) = (7, 3, 5, 1)$$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -3 & 1 \\ -1 & 1 & 1 & 1 \\ 0 & -2 & 1 & 1 \end{pmatrix}$$

The orthogonality and normality conditions are thus given by

$$\begin{pmatrix} 1 & 0 & -3 & 1 \\ -1 & 1 & 1 & 1 \\ 0 & -2 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

This yields the unique solution

$$y_1^* = \tfrac{1}{2},\ y_2^* = \tfrac{1}{6},\ y_3^* = \tfrac{5}{24},\ y_4^* = \tfrac{1}{8}$$

Thus,

$$z^* = \left(\frac{7}{\frac{1}{2}}\right)^{\frac{1}{2}}\left(\frac{3}{\frac{1}{6}}\right)^{\frac{1}{6}}\left(\frac{5}{\frac{5}{24}}\right)^{\frac{5}{24}}\left(\frac{1}{\frac{3}{24}}\right)^{\frac{3}{24}} = 15.23$$

From the equation $U_j^* = y_j^* z^*$ it follows that

$$\begin{aligned}
7x_1x_2^{-1} &= U_1 = \tfrac{1}{2}(15.23) = 7.615\\
3x_2x_3^{-2} &= U_2 = \tfrac{1}{6}(15.23) = 2.538\\
5x_1^{-3}x_2x_3 &= U_3 = \tfrac{5}{24}(15.23) = 3.173\\
x_1x_2x_3 &= U_4 = \tfrac{1}{8}(15.23) = 1.904
\end{aligned}$$

The solution of these equations is

$$x_1^* = 1.316,\ x_2^* = 1.21,\ x_3^* = 1.2$$

which is the optimal solution of the problem.

Example 21.2-5

Consider the problem

$$\text{Minimize } z = 5x_1x_2^{-1} + 2x_1^{-1}x_2 + 5x_1 + x_2^{-1}$$

The orthogonality and normality conditions are given by

$$\begin{pmatrix} 1 & -1 & 1 & 0\\ -1 & 1 & 0 & -1\\ 1 & 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} y_1\\ y_2\\ y_3\\ y_4\end{pmatrix} = \begin{pmatrix}0\\0\\1\end{pmatrix}$$

Because $N > n + 1$, these equations do not yield y_j directly. Solving for y_1, y_2, and y_3 in terms of y_4 we get

$$\begin{pmatrix} 1 & -1 & 1\\ -1 & 1 & 0\\ 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} y_1\\ y_2\\ y_3\end{pmatrix} = \begin{pmatrix}0\\ y_4\\ 1 - y_4\end{pmatrix}$$

or

$$\begin{aligned}
y_1 &= .5(1 - 3y_4)\\
y_2 &= .5(1 - y_4)\\
y_3 &= y_4
\end{aligned}$$

The associated dual problem is

$$\text{Maximize } w = \left(\frac{5}{.5(1 - 3y_4)}\right)^{.5(1-3y_4)}\left(\frac{2}{.5(1 - y_4)}\right)^{.5(1-y_4)}\left(\frac{5}{y_4}\right)^{y_4}\left(\frac{1}{y_4}\right)^{y_4}$$

Maximization of w is equivalent to maximization of ln w. The latter is easier to manipulate, however. Thus,

$$\ln w = .5(1 - 3y_4)\{\ln 10 - \ln(1 - 3y_4)\} + .5(1 - y_4)\{\ln 4 - \ln(1 - y_4)\} + y_4(\ln 5 - 2\ln y_4)$$

The value of y_4 maximizing ln w must be unique (because the primal problem has a unique minimum). Hence,

$$\frac{\partial \ln w}{\partial y_4} = (-\tfrac{3}{2}\ln 10 - \tfrac{1}{2}\ln 4 + \ln 5) + \tfrac{3}{2}\ln(1 - 3y_4) + \tfrac{1}{2}\ln(1 - y_4) - 2\ln y_4 = 0$$

This gives, after simplification,

$$-\ln\left(\frac{2 \times 10^{\frac{3}{2}}}{5}\right) + \ln\left(\frac{(1 - 3y_4)^{\frac{3}{2}}(1 - y_4)^{\frac{1}{2}}}{y_4^2}\right) = 0$$

or

$$\frac{\sqrt{(1 - 3y_4)^3(1 - y_4)}}{y_4^2} = 12.6$$

which yields $y_4^* \approx .16$. Hence, $y_3^* = .16$, $y_2^* = .42$, and $y_1^* = .26$.

The value of z^* is obtained from

$$z^* = w^* = \left(\tfrac{5}{.26}\right)^{.26}\left(\tfrac{2}{.42}\right)^{.42}\left(\tfrac{5}{.16}\right)^{.16} \approx 9.661$$

Hence,

$$U_3 = .16(9.661) = 1.546 = 5x_1$$
$$U_4 = .16(9.661) = .1546 = x_2^{-1}$$

The equations yield $x_1^* = .309$ and $x_2^* = .647$.

PROBLEM SET 21.2C

1. Solve the following problem by geometric programming.

$$\text{Minimize } z = 2x_1^{-1}x_2^2 + x_1^4x_2^{-2} + 4x_1^2$$
$$x_1, x_2 > 0$$

2. Solve the following problem by geometric programming.

$$\text{Minimize } z = 5x_1x_2^{-1}x_3^2 + x_1^{-2}x_3^{-1} + 10x_2^3 + 2x_1^{-1}x_2x_3^{-3}$$
$$x_1, x_2, x_3 > 0$$

3. Solve the following problem by geometric programming.

$$\text{Minimize } z = 2x_1^2x_2^{-3} + 8x_1^{-3}x_2 + 3x_1x_2$$
$$x_1, x_2 > 0$$

4. Solve the following problem by geometric programming.

$$\text{Minimize } z = 2x_1^3x_2^{-3} + 4x_1^{-2}x_2 + x_1x_2$$
$$x_1, x_2 > 0$$

21.2.4 Stochastic Programming

Stochastic programming deals with situations where some or all parameters of the problem are random variables. Such cases seem typical of real-life problems, where it may be difficult to determine the values of the parameters with certainty.

The idea of stochastic programming is to convert the probabilistic problem into an equivalent deterministic situation. This section deals with **chance-constrained programming**, defined as

$$\text{Maximize } z = \sum_{j=1}^{n} c_j x_j$$

subject to

$$P\left\{\sum_{j=1}^{n} a_{ij}x_j \leq b_i\right\} \geq 1 - \alpha_i, \quad i = 1, 2, \ldots, m; \ x_j \geq 0, \quad \text{for all } j$$

The name "chance-constrained" follows from the fact that each constraint is realized with a minimum probability of $1 - \alpha_i$, $0 < \alpha_i < 1$. It is assumed that all a_{ij} and b_i are random variables. Three cases are considered:

1. Only a_{ij} is random for all i and j.
2. Only b_i is random for all i.
3. Both a_{ij} and b_i are random for all i and j.

In all three cases, it is assumed that the parameters are normally distributed with known means and variances.

Case 1. Each a_{ij} is normally distributed with mean $E\{a_{ij}\}$, variance $\text{var}\{a_{ij}\}$, and $\text{cov}\{a_{ij}, a_{i'j'}\}$ of a_{ij} and $a_{i'j'}$.

Consider the ith constraint

$$P\left\{\sum_{j=1}^{n} a_{ij}x_j \leq b_i\right\} \geq 1 - \alpha_i$$

and define

$$h_i = \sum_{j=1}^{n} a_{ij}x_j$$

Then h_i is normally distributed with

$$E\{h_i\} = \sum_{j=1}^{n} E\{a_{ij}\}x_j$$

$$\text{var}\{h_i\} = \mathbf{X}^T\mathbf{D}_i\mathbf{X}$$

where

$$\mathbf{X} = (x_1, \ldots, x_n)^T$$

$$\mathbf{D}_i = i\text{th covariance matrix}$$

$$= \begin{pmatrix} var\{a_{i1}\} & \ldots & \text{cov}\{a_{i1}, a_{in}\} \\ \vdots & \vdots & \vdots \\ \text{cov}\{a_{in}, a_{i1}\} & \ldots & \text{var}\{a_{in}\} \end{pmatrix}$$

Now

$$P\{h_i \leq b_i\} = P\left\{\frac{h_i - E\{h_i\}}{\sqrt{\text{var}\{h_i\}}} \leq \frac{b_i - E\{h_i\}}{\sqrt{\text{var}\{h_i\}}}\right\} \geq 1 - \alpha_i$$

where $\frac{h_i - E\{h_i\}}{\sqrt{\text{var}\{h_i\}}}$ is standard normal with mean zero and variance one. This means that

$$P\{h_i \leq b_i\} = F\left(\frac{b_i - E\{h_i\}}{\sqrt{\text{var}\{h_i\}}}\right)$$

where F represents the CDF of the standard normal distribution.

Let K_{α_i} be the standard normal value such that

$$F(K_{\alpha_i}) = 1 - \alpha_i$$

Then the statement $P\{h_i \leq b_i\} \geq 1 - \alpha_i$ is realized if and only if

$$\frac{b_i - E\{h_i\}}{\sqrt{\text{var}\{h_i\}}} \geq K_{\alpha_i}$$

This yields the following nonlinear deterministic constraint:

$$\sum_{j=1}^{n} E\{a_{ij}\}x_j + K_{\alpha_i}\sqrt{\mathbf{X}^T\mathbf{D}_i\mathbf{X}} \leq b_i$$

For the special case where the normal distributions are independent,

$$\text{cov}\{a_{ij}, a_{i'j'}\} = 0$$

and the last constraint reduces to

$$\sum_{j=1}^{n} E\{a_{ij}\}x_j + K_{\alpha_i}\sqrt{\sum_{j=1}^{n} \text{var}\{a_{ij}\}x_j^2} \leq b_i$$

This constraint can be put in the separable programming form (Section 21.2.1) by using the substitution

$$y_i = \sqrt{\sum_{j=1}^{n} \text{var}\{a_{ij}\}x_j^2}, \text{ for all } i$$

Thus, the original constraint is equivalent to

$$\sum_{j=1}^{n} E\{a_{ij}\}x_j + K_{\alpha_i}y_i \leq b_i$$

and

$$\sum_{j=1}^{n} \text{var}\{a_{ij}\}x_j^2 - y_i^2 = 0$$

Case 2. Only b_i is normal with mean $E\{b_i\}$ and variance $\text{var}\{b_i\}$. The analysis is similar to that of case 1. Consider the stochastic constraint

$$P\left\{b_i \geq \sum_{j=1}^{n} a_{ij}x_j\right\} \geq \alpha_i$$

As in case 1,

$$P\left\{\frac{b_i - E\{b_i\}}{\sqrt{\text{var}\{b_i\}}} \geq \frac{\sum_{j=1}^{n} a_{ij}x_j - E\{b_i\}}{\sqrt{\text{var}\{b_i\}}}\right\} \geq \alpha_i$$

This can hold only if

$$\frac{\sum_{j=1}^{n} a_{ij}x_j - E\{b_i\}}{\sqrt{\text{var}\{b_i\}}} \leq K_{\alpha_i}$$

Thus, the stochastic constraint is equivalent to the deterministic linear constraint

$$\sum_{j=1}^{n} a_{ij}x_j \leq E\{b_i\} + K_{\alpha_i}\sqrt{\text{var}\{b_i\}}$$

Case 3. In this case all a_{ij} and b_i are normal random variables. Consider the constraint

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i$$

This may be written

$$\sum_{j=1}^{n} a_{ij}x_j - b_i \leq 0$$

Because all a_{ij} and b_i are normal, $\sum_{j=1}^{n} a_{ij}x_j - b_i$ is also normal. This shows that the chance constraint reduces to the situation in case 1 and is treated in a similar manner.

Example 21.2-6

Consider the chance-constrained problem

$$\text{Maximize } z = 5x_1 + 6x_2 + 3x_3$$

subject to

$$P\{a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \leq 8\} \geq .95$$

$$P\{5x_1 + x_2 + 6x_3 \leq b_2\} \geq .10$$

$$x_1, x_2, x_3 \geq 0$$

Assume that the a_{1j}'s are *independent* normally distributed random variables with the following means and variances:

$$E\{a_{11}\} = 1,\ E\{a_{12}\} = 3,\ E\{a_{13}\} = 9$$

$$\text{var}\{a_{11}\} = 25,\ \text{var}\{a_{12}\} = 16,\ \text{var}\{a_{13}\} = 4$$

The parameter b_2 is normally distributed with mean 7 and variance 9.

From standard normal tables in Appendix D,

$$K_{\alpha_1} = K_{.05} \approx 1.645,\quad K_{\alpha_2} = K_{.10} \approx 1.285$$

The two constraints are converted deterministically to

$$x_1 + 3x_2 + 9x_3 + 1.645\sqrt{25x_1^2 + 16x_2^2 + 4x_3^2} \le 8$$

$$5x_1 + x_2 + 6x_3 \le 7 + 1.285(3) = 10.855$$

If we let

$$y^2 = 25x_1^2 + 16x_2^2 + 4x_3^2$$

the problem becomes

$$\text{Maximize } z = 5x_1 + 6x_2 + 3x_3$$

subject to

$$x_1 + 3x_2 + 9x_3 + 1.645y \le 8$$

$$25x_1^2 + 16x_2^2 + 4x_3^2 - y^2 = 0$$

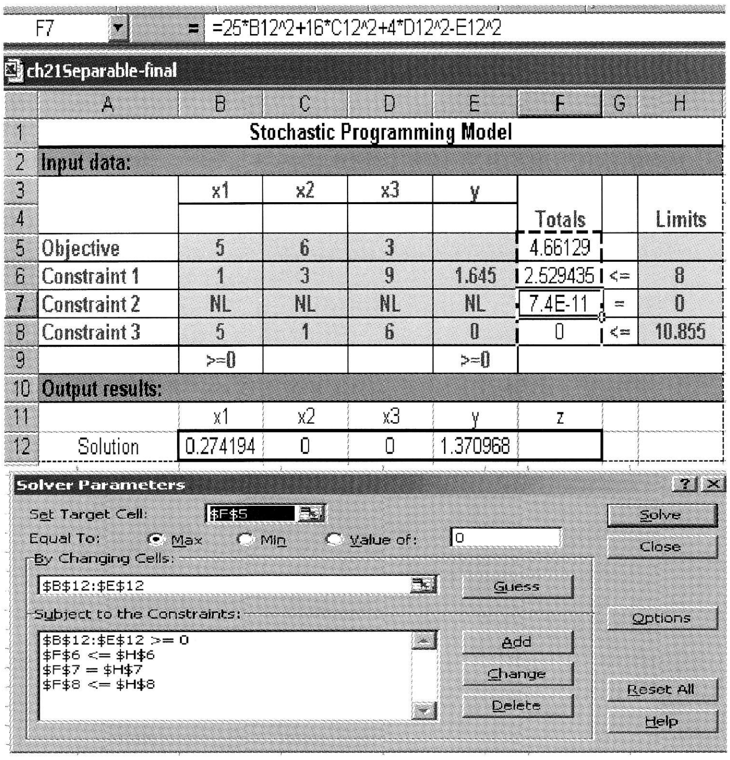
$$5x_1 + x_2 + 6x_3 \le 10.855$$

$$x_1, x_2, x_3, y \ge 0$$

which can be solved by separable programming.

Excel optimum solution of this nonlinear problem is given in Figure 21.6 (file ch21SolverStochasticProgramming.xls). Only the left-hand side of constraint 2 is nonlinear and is entered in cell F7 as

```
=25*B12^2+16*c12^2+4*D12^2−E12^2
```



F7 = =25*B12^2+16*C12^2+4*D12^2-E12^2

ch21Separable-final

	A	B	C	D	E	F	G	H
1	Stochastic Programming Model							
2	Input data:							
3		x1	x2	x3	y			
4						Totals		Limits
5	Objective	5	6	3		4.66129		
6	Constraint 1	1	3	9	1.645	2.529435	<=	8
7	Constraint 2	NL	NL	NL	NL	7.4E-11	=	0
8	Constraint 3	5	1	6	0	0	<=	10.855
9		>=0			>=0			
10	Output results:							
11		x1	x2	x3	y	z		
12	Solution	0.274194	0	0	1.370968			

Solver Parameters

Set Target Cell: $F$5

Equal To: ● Max ○ Min ○ Value of: 0

By Changing Cells: $B$12:$E$12

Subject to the Constraints:

```
$B$12:$E$12 >= 0
$F$6 <= $H$6
$F$7 = $H$7
$F$8 <= $H$8
```

FIGURE 21.6

Excel solution of the stochastic programming problem of Example 21.2-6

PROBLEM SET 21.2D

1. Convert the following stochastic problem into an equivalent deterministic model.

$$\text{Maximize } z = x_1 + 2x_2 + 5x_3$$

subject to

$$P\{a_1x_1 + 3x_2 + a_3x_3 \leq 10\} \geq 0.9$$
$$P\{7x_1 + 5x_2 + x_3 \leq b_2\} \geq 0.1$$
$$x_1, x_2, x_3 \geq 0$$

Assume that a_1 and a_3 are independent and normally distributed random variables with means $E\{a_1\} = 2$ and $E\{a_3\} = 5$ and variances $\text{var}\{a_1\} = 9$ and $\text{var}\{a_3\} = 16$. Assume further that b_2 is normally distributed with mean 15 and variance 25.

2. Consider the following stochastic programming model:

$$\text{Maximize } z = x_1 + x_2^2 + x_3$$

subject to

$$P\{x_1^2 + a_2x_2^3 + a_3\sqrt{x_3} \leq 10\} \geq 0.9$$
$$x_1, x_2, x_3 \geq 0$$

The parameters a_2 and a_3 are independent and normally distributed random variables with means 5 and 2, and variance 16 and 25, respectively. Convert the problem into the (deterministic) separable programming form.

21.2.5 Linear Combinations Method

This method deals with the following problem in which all constraints are linear:

$$\text{Maximize } z = f(\mathbf{X})$$

subject to

$$\mathbf{AX} \leq \mathbf{b}, \mathbf{X} \geq \mathbf{0}$$

The procedure is based on the steepest ascent (gradient) method (Section 21.1.2). However, the direction specified by the gradient vector may not yield a feasible solution for the constrained problem. Also, the gradient vector will not necessarily be null at the optimum (constrained) point. The steepest ascent method thus must be modified to handle the constrained case.

Let $\mathbf{X}^k$ be the *feasible* trial point at iteration k. The objective function $f(\mathbf{X})$ can be expanded in the neighborhood of $\mathbf{X}^k$ using Taylor's series. This gives

$$f(\mathbf{X}) \approx f(\mathbf{X}^k) + \nabla f(\mathbf{X}^k)(\mathbf{X} - \mathbf{X}^k) = (f(\mathbf{X}^k) - \nabla f(\mathbf{X}^k)\mathbf{X}^k) + \nabla f(\mathbf{X}^k)\mathbf{X}$$

The procedure calls for determining a feasible point $\mathbf{X} = \mathbf{X}^*$ such that $f(\mathbf{X})$ is maximized subject to the (linear) constraints of the problem. Because $f(\mathbf{X}^k) - \nabla f(\mathbf{X}^k)\mathbf{X}^k$ is a constant, the problem for determining $\mathbf{X}^*$ reduces to solving the following linear program:

$$\text{Maximize } w_k(\mathbf{X}) = \nabla f(\mathbf{X}^k)\mathbf{X}$$

subject to

$$\mathbf{AX} \leq \mathbf{b}, \mathbf{X} \geq \mathbf{0}$$

Given w_k is constructed from the gradient of $f(\mathbf{X})$ at $\mathbf{X}^k$, an improved solution point can be secured if and only if $w_k(\mathbf{X}^*) > w_k(\mathbf{X}^k)$. From Taylor's expansion, the condition does not guarantee that $f(\mathbf{X}^*) > f(\mathbf{X}^k)$ unless $\mathbf{X}^*$ is in the neighborhood of $\mathbf{X}^k$. However, given $w_k(\mathbf{X}^*) > w_k(\mathbf{X}^k)$, there must exist a point $\mathbf{X}^{k+1}$ on the line segment $(\mathbf{X}^k, \mathbf{X}^*)$ such that $f(\mathbf{X}^{k+1}) > f(\mathbf{X}^k)$. The objective is to determine $\mathbf{X}^{k+1}$. Define

$$\mathbf{X}^{k+1} = (1 - r)\mathbf{X}^k + r\mathbf{X}^* = \mathbf{X}^k + r(\mathbf{X}^* - \mathbf{X}^k), \quad 0 < r \leq 1$$

This means that $\mathbf{X}^{k+1}$ is a **linear combination** of $\mathbf{X}^k$ and $\mathbf{X}^*$. Because $\mathbf{X}^k$ and $\mathbf{X}^*$ are two feasible points in a *convex* solution space, $\mathbf{X}^{k+1}$ is also feasible. By comparison with the steepest ascent method (Section 21.1.2), the parameter r represents the step size.

The point $\mathbf{X}^{k+1}$ is determined such that $f(\mathbf{X})$ is maximized. Because $\mathbf{X}^{k+1}$ is a function of r only, $\mathbf{X}^{k+1}$ is determined by maximizing

$$h(r) = f(\mathbf{X}^k + r(\mathbf{X}^* - \mathbf{X}^k))$$

The procedure is repeated until, at the kth iteration, $w_k(\mathbf{X}^*) \leq w_k(\mathbf{X}^k)$. At this point, no further improvements are possible, and the process terminates with $\mathbf{X}^k$ as the best solution point.

The linear programming problems generated at the successive iterations differ only in the coefficients of the objective function. Sensitivity analysis procedures presented in Section 4.5 thus may be used to carry out calculations efficiently.

Example 21.2-7

Consider the quadratic programming of Example 21.2–3.

$$\text{Maximize } f(\mathbf{X}) = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

subject to

$$x_1 + 2x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

Let the initial trial point be $\mathbf{X}^0 = (\frac{1}{2}, \frac{1}{2})$, which is feasible. Now

$$\nabla f(\mathbf{X}) = (4 - 4x_1 - 2x_2, 6 - 2x_1 - 4x_2)$$

Iteration 1.

$$\nabla f(\mathbf{X}^0) = (1, 3)$$

The associated linear program maximizes $w_1 = x_1 + 3x_2$ subject to the constraints of the original problem. This gives the optimal solution $\mathbf{X}^* = (0, 1)$. The values of w_1 at $\mathbf{X}^0$ and $\mathbf{X}^*$ equal 2 and 3, respectively. Hence, a new trial point is determined as

$$\mathbf{X}^1 = (\tfrac{1}{2}, \tfrac{1}{2}) + r[(0, 1) - (\tfrac{1}{2}, \tfrac{1}{2})] = (\tfrac{1 - r}{2}, \tfrac{1 + r}{2})$$

The maximization of

$$h(r) = f(\tfrac{1 - r}{2}, \tfrac{1 + r}{2})$$

yields $r^1 = 1$. Thus $\mathbf{X}^1 = (0, 1)$ with $f(\mathbf{X}^1) = 4$.

21.2.6

Iteration 2.

$$\nabla f(\mathbf{X}^1) = (2, 2)$$

The objective function of the new linear programming problem is $w_2 = 2x_1 + 2x_2$. The optimum solution to this problem yields $\mathbf{X}^* = (2, 0)$. Because the values of w_2 at $\mathbf{X}^1$ and $\mathbf{X}^*$ are 2 and 4, a new trial point must be determined. Thus

$$\mathbf{X}^2 = (0, 1) + r[(2, 0) - (0, 1)] = (2r, 1 - r)$$

The maximization of

$$h(r) = f(2r, 1 - r)$$

yields $r^2 = \frac{1}{6}$. Thus $\mathbf{X}^2 = (\frac{1}{3}, \frac{5}{6})$ with $f(\mathbf{X}^2) \approx 4.16$.

Iteration 3.

$$\nabla f(\mathbf{X}^2) = (1, 2)$$

The corresponding objective function is $w_3 = x_1 + 2x_2$. The optimum solution of this problem yields the alternative solutions $\mathbf{X}^* = (0, 1)$ and $\mathbf{X}^* = (2, 0)$. The value of w_3 for both points equals its value at $\mathbf{X}^2$. Consequently, no further improvements are possible. The *approximate* optimum solution is $\mathbf{X}^2 = (\frac{1}{3}, \frac{5}{6})$ with $f(\mathbf{X}^2) \approx 4.16$. This happens to be the exact optimum.

PROBLEM SET 21.2E

1. Solve the following problem by the linear combinations method.

$$\text{Minimize } f(\mathbf{X}) = x_1^3 + x_2^3 - 3x_1x_2$$

subject to

$$\begin{aligned} 3x_1 + x_2 &\le 3 \\ 5x_1 - 3x_2 &\le 5 \\ x_1, x_2 &\ge 0 \end{aligned}$$

21.2.6 SUMT Algorithm

In this section, a more general gradient method is presented. It is assumed that the objective function $f(\mathbf{X})$ is concave and each constraint function $g_i(\mathbf{X})$ is convex. Moreover, the solution space must have an interior. This rules out both implicit and explicit use of *equality* constraints.

The SUMT (Sequential Unconstrained Maximization Technique) algorithm is based on transforming the constrained problem into an equivalent *un*constrained problem. The procedure is more or less similar to the use of the Lagrange multipliers method. The transformed problem can then be solved using the steepest ascent method (Section 21.1.2).

To clarify the concept, consider the new function

$$p(\mathbf{X}, t) = f(\mathbf{X}) + t\left(\sum_{i=1}^{m}\frac{1}{g_i(\mathbf{X})} - \sum_{j=1}^{n}\frac{1}{x_j}\right)$$

where t is a nonnegative parameter. The second summation sign accounts for the nonnegativity constraints, which must be put in the form $-x_j \leq 0$ to be consistent with the original constraints. Because $g_i(\mathbf{X})$ is convex, $\frac{1}{g_i(\mathbf{X})}$ is concave. This means that $p(\mathbf{X}, t)$ is concave in $\mathbf{X}$. Consequently, $p(\mathbf{X}, t)$ possesses a unique maximum. Optimization of the original constrained problem is equivalent to optimization of $p(\mathbf{X}, t)$.

The algorithm is initiated by arbitrarily selecting an initial *nonnegative* value for t. An initial point $\mathbf{X}_0$ is selected as the first trial solution. This point must be an interior point—that is, it must not lie on the boundaries of the solution space. Given the value of t, the steepest ascent method is used to determine the corresponding optimal solution (maximum) of $p(\mathbf{X}, t)$.

The new solution point will always be an interior point because if the solution point is close to the boundaries, at least one of the functions $\frac{1}{g_i(\mathbf{X})}$ or $-\frac{1}{x_i}$ will acquire a very large negative value. Because the objective is to maximize $p(\mathbf{X}, t)$, such solution points are automatically excluded. The main result is that successive solution points will always be interior points. Consequently, the problem can always be treated as an unconstrained case.

Once the optimum solution corresponding to a given value of t is obtained, a new value of t is generated and the optimization process (using the steepest ascent method) is repeated. If t' is the current value of t, the next value, t'', must be selected such that $0 < t'' < t'$.

The SUMT algorithm ends when, for two successive values of t, the corresponding *optimum* values of $\mathbf{X}$ obtained by maximizing $p(\mathbf{X}, t)$ are approximately the same. At this point further trials will produce little improvement.

Actual implementation of SUMT involves more details than have been presented here. Specifically, the selection of an initial value of t is an important factor that can affect the speed of convergence. Further, the determination of an initial interior point may require special techniques. These details can be found in Fiacco and McCormick (1968).

SELECTED REFERENCES

Bazaraa, M., H. Sherall, and C. Shetty, *Nonlinear Programming, Theory and Algorithms,* 2nd ed., Wiley, New York, 1993.

Beightler, C., D. Phillips, and D. Wilde, *Foundations of Optimization,* 2nd ed., Prentice Hall, Upper Saddle River, NJ, 1979.

Fiacco, A., and G. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques,* Wiley, New York, 1968.

Rardin, D., *Optimization in Operations Research,* Prentice Hall, Upper Saddle River, NJ, 1998.

APPENDIX A

Review of Vectors and Matrices

A.1 VECTORS

A.1.1 Definition of a Vector

Let $p_1, p_2, \ldots, p_n$ be any n real numbers and $\mathbf{P}$ an ordered set of these real numbers—that is,

$$\mathbf{P} = (p_1, p_2, \ldots, p_n)$$

Then $\mathbf{P}$ is an n-vector (or simply a vector). The ith component of $\mathbf{P}$ is p_i. For example, $\mathbf{P} = (2, 4)$ is a two-dimensional vector with $p_1 = 2$ and $p_2 = 4$.

A.1.2 Addition (Subtraction) of Vectors

Consider the n-vectors

$$\mathbf{P} = (p_1, p_2, \ldots, p_n)$$
$$\mathbf{Q} = (q_1, q_2, \ldots, q_n)$$
$$\mathbf{R} = (r_1, r_2, \ldots, r_n)$$

For $\mathbf{R} = \mathbf{P} \pm \mathbf{Q}$, component i is computed as $r_i = p_i \pm q_i$.

In general, given the vectors $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{S}$,

$$\mathbf{P} + \mathbf{Q} = \mathbf{Q} + \mathbf{P} \qquad \text{(Commutative law)}$$
$$(\mathbf{P} + \mathbf{Q}) + \mathbf{S} = \mathbf{P} + (\mathbf{Q} + \mathbf{S}) \qquad \text{(Associative law)}$$
$$\mathbf{P} + (-\mathbf{P}) = \mathbf{0} \qquad \text{(Zero or null vector)}$$

A.1.3 Multiplication of Vectors by Scalars

Given a vector **P** and a scalar (constant) quantity θ, the new vector

$$\mathbf{Q} = \theta\mathbf{P} = (\theta p_1, \theta p_2, \dots, \theta p_n)$$

is the scalar product of **P** and θ. In general, given the vectors **P** and **S** and the scalars θ and γ,

$$\theta(\mathbf{P} + \mathbf{S}) = \theta\mathbf{P} + \theta\mathbf{S}$$

$$\theta(\gamma\mathbf{P}) = (\theta\gamma)\mathbf{P}$$

A.1.4 Linearly Independent Vectors

The vectors $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n$ are *linearly independent* if, and only if

$$\sum_{j=1}^{n} \theta_j \mathbf{P}_j = \mathbf{0} \Rightarrow \theta_j = 0,\ j = 1, 2, \dots, n$$

If, for some $\theta_j \neq 0$,

$$\sum_{j=1}^{n} \theta_j \mathbf{P}_j = \mathbf{0}$$

then the vectors are *linearly dependent*. For example, the vectors

$$\mathbf{P}_1 = (1, 2),\ \mathbf{P}_2 = (2, 4)$$

are linearly dependent because for $\theta_1 = 2$ and $\theta_2 = -1$,

$$\theta_1\mathbf{P}_1 + \theta_2\mathbf{P}_2 = \mathbf{0}$$

A.2 MATRICES

A.2.1 Definition of a Matrix

A matrix is a rectangular array of elements. The element a_{ij} of the matrix **A** occupies the ith row and jth column of the array. A matrix with m rows and n columns is said to be of size (or order) $m \times n$. For example, the following matrix is of size (4×3).

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{pmatrix} = \|a_{ij}\|_{4\times 3}$$

A.2.2 Types of Matrices

1. A **square matrix** has $m = n$.
2. An **identity matrix** is a square matrix in which the main diagonal elements are 1 and the off-diagonal elements are zero. For example, a (3×3) identity matrix is given by

$$\mathbf{I}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

3. A **row vector** is a matrix with one row and n columns.
4. A **column vector** is a matrix with m rows and one column.
5. The matrix $\mathbf{A}^T$ is the **transpose** of $\mathbf{A}$ if the element a_{ij} in $\mathbf{A}$ equals the element a_{ji} in $\mathbf{A}^T$ for all i and j. For example,

$$\mathbf{A} = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix} \Rightarrow \mathbf{A}^T = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

6. A matrix $\mathbf{B} = \mathbf{0}$ is a **zero matrix** if every element of $\mathbf{B}$ is zero.
7. Two matrices $\mathbf{A} = \|a_{ij}\|$ and $\mathbf{B} = \|b_{ij}\|$ are equal if, and only if, they have the same size and $a_{ij} = b_{ij}$ for all i and j.

A.2.3 Matrix Arithmetic Operations

In matrices only addition (subtraction) and multiplication are defined. The division, though not defined, is replaced by inversion (see Section A.2.6).

Addition (Subtraction) of Matrices. Two matrices $\mathbf{A} = \|a_{ij}\|$ and $\mathbf{B} = \|b_{ij}\|$ can be added if they are of the same size ($m \times n$). The sum $\mathbf{D} = \mathbf{A} + \mathbf{B}$ is obtained by adding the corresponding elements. Thus,

$$\|d_{ij}\|_{m\times n} = \|a_{ij} + b_{ij}\|_{m\times n}$$

If the matrices **A**, **B**, and **C** have the same size, then

$$\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$$

$$\mathbf{A} \pm (\mathbf{B} \pm \mathbf{C}) = (\mathbf{A} \pm \mathbf{B}) \pm \mathbf{C}$$

$$(\mathbf{A} \pm \mathbf{B})^T = \mathbf{A}^T \pm \mathbf{B}^T$$

Product of Matrices. The product $\mathbf{D} = \mathbf{AB}$ of two matrices, $\mathbf{A} = \|a_{ij}\|$ and $\mathbf{B} = \|b_{ij}\|$, is defined if, and only if, the number of columns of **A** equals the number of rows of **B**. If **A** is of size ($m \times r$) and **B** is of size ($r \times n$), then **D** must be of size $m \times n$, where m and n are arbitrary positive integer values. In this case, the elements of **D** are computed as

$$d_{ij} = \sum_{k=1}^{r} a_{ik} b_{kj}, \quad \text{for all } i \text{ and } j$$

For example, given

$$\mathbf{A} = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 5 & 7 & 9 \\ 6 & 8 & 0 \end{pmatrix}$$

we have

$$\mathbf{D} = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}\begin{pmatrix} 5 & 7 & 9 \\ 6 & 8 & 0 \end{pmatrix} = \begin{pmatrix} (1 \times 5 + 3 \times 6) & (1 \times 7 + 3 \times 8) & (1 \times 9 + 3 \times 0) \\ (2 \times 5 + 4 \times 6) & (2 \times 7 + 4 \times 8) & (2 \times 9 + 4 \times 0) \end{pmatrix}$$

$$= \begin{pmatrix} 23 & 31 & 9 \\ 34 & 46 & 18 \end{pmatrix}$$

In general, $\mathbf{AB} \neq \mathbf{BA}$ even if **BA** is defined.

The following general properties apply to matrix multiplication:

$$\mathbf{I}_m\mathbf{A} = \mathbf{A}\mathbf{I}_n = \mathbf{A},\ \mathbf{I}_m \text{ and } \mathbf{I}_n \text{ are identity matrices}$$

$$(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$$

$$\mathbf{C}(\mathbf{A} + \mathbf{B}) = \mathbf{CA} + \mathbf{CB}$$

$$(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$$

$$\alpha(\mathbf{AB}) = (\alpha\mathbf{A})\mathbf{B} = \mathbf{A}(\alpha\mathbf{B}),\ \alpha \text{ is a scalar}$$

Multiplication of Partitioned Matrices. Let $\mathbf{A}$ be an $(m \times r)$-matrix and $\mathbf{B}$ an $(r \times n)$-matrix. Assume that $\mathbf{A}$ and $\mathbf{B}$ are partitioned as follows:

$$\mathbf{A} = \left(\begin{array}{c|c|c} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \hline \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} \end{array}\right),\ \mathbf{B} = \left(\begin{array}{c|c} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \hline \mathbf{B}_{21} & \mathbf{B}_{22} \\ \hline \mathbf{B}_{31} & \mathbf{B}_{32} \end{array}\right)$$

The partitioning assumes that the number of columns of $\mathbf{A}_{ij}$ is equal to the number of rows of $\mathbf{B}_{ij}$ for all i and j. Then

$$\mathbf{A} \times \mathbf{B} = \left(\begin{array}{c|c} \mathbf{A}_{11}\mathbf{B}_{11} + \mathbf{A}_{12}\mathbf{B}_{21} + \mathbf{A}_{13}\mathbf{B}_{31} & \mathbf{A}_{11}\mathbf{B}_{12} + \mathbf{A}_{12}\mathbf{B}_{22} + \mathbf{A}_{13}\mathbf{B}_{32} \\ \hline \mathbf{A}_{21}\mathbf{B}_{11} + \mathbf{A}_{22}\mathbf{B}_{21} + \mathbf{A}_{23}\mathbf{B}_{31} & \mathbf{A}_{21}\mathbf{B}_{12} + \mathbf{A}_{22}\mathbf{B}_{22} + \mathbf{A}_{23}\mathbf{B}_{32} \end{array}\right)$$

For example,

$$\left(\begin{array}{c|cc} 1 & 2 & 3 \\ \hline 1 & 0 & 5 \\ 2 & 5 & 6 \end{array}\right)\left(\begin{array}{c} 4 \\ \hline 1 \\ 8 \end{array}\right) = \left(\begin{array}{c} (1)(4) + (2 \quad 3)\begin{pmatrix} 1 \\ 8 \end{pmatrix} \\ \hline \begin{pmatrix} 1 \\ 2 \end{pmatrix}(4) + \begin{pmatrix} 0 & 5 \\ 5 & 6 \end{pmatrix}\begin{pmatrix} 1 \\ 8 \end{pmatrix} \end{array}\right) = \left(\begin{array}{c} 4 + 2 + 24 \\ \hline \begin{pmatrix} 4 \\ 8 \end{pmatrix} + \begin{pmatrix} 40 \\ 53 \end{pmatrix} \end{array}\right) = \begin{pmatrix} 30 \\ 44 \\ 61 \end{pmatrix}$$

A.2.4 Determinant of a Square Matrix

Consider the n-square matrix

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

Next, define the product

$$P_{j_1 j_2 \dots j_n} = a_{1j_1} a_{2j_2} \dots a_{nj_n}$$

such that each column and each row of $\mathbf{A}$ is represented exactly once among the subscripts of $j_1, j_2, \dots$, and j_n. Next, define

$$\in_{j_1 j_2 \dots j_n} = \begin{cases} 1, & j_1 j_2 \dots j_n \text{ even permutation} \\ 0, & j_1 j_2 \dots j_n \text{ odd permutation} \end{cases}$$

Let ρ represent the summation over all $n!$ permutations; then the determinant of $\mathbf{A}$, det $\mathbf{A}$ or $|\mathbf{A}|$, is computed as

$$\sum_{\rho} \in_{j_1 j_2 \dots j_n} P_{j_1 j_2 \dots j_n}$$

As an illustration, consider

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Then

$$|\mathbf{A}| = a_{11}(a_{22}\,a_{33} - a_{23}\,a_{32}) - a_{12}(a_{21}\,a_{33} - a_{31}\,a_{23}) + a_{13}(a_{21}\,a_{32} - a_{22}\,a_{31})$$

The properties of a determinant are:

1. The value of a determinant is zero if every element of a row or a column is zero.
2. $|\mathbf{A}| = |\mathbf{A}^T|$.
3. If **B** is obtained from **A** by interchanging any two rows or any two columns, then $|\mathbf{B}| = -|\mathbf{A}|$.
4. If two rows (or two columns) of **A** are multiples of one another, then $|\mathbf{A}| = 0$.
5. The value of $|\mathbf{A}|$ remains the same if scalar α times a column (row) vector is added to another column (row) vector.
6. If every element of a column or a row of a determinant is multiplied by a scalar α, the value of the determinant is multiplied by α.
7. If **A** and **B** are two n-square matrices, then

$$|\mathbf{AB}| = |\mathbf{A}||\mathbf{B}|$$

Definition of the Minor of a Determinant. The minor M_{ij} of the element a_{ij} in the determinant $|\mathbf{A}|$ is obtained from the matrix **A** by striking out the ith row and jth column of **A**. For example, for

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

$$M_{11} = \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}, \quad M_{22} = \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix}, \quad \ldots$$

Definition of the Adjoint Matrix. Let $A_{ij} = (-1)^{i+j}M_{ij}$ be defined as the **cofactor** of the element a_{ij} of the square matrix **A**. Then, the adjoint matrix of **A** is the transpose of $\|A_{ij}\|$ and is defined as:

$$\text{adj}\,\mathbf{A} = \|A_{ij}\|^T = \begin{pmatrix} A_{11} & A_{21} & \ldots & A_{n1} \\ A_{12} & A_{22} & \ldots & A_{n2} \\ \vdots & \vdots & \vdots & \vdots \\ A_{1n} & A_{2n} & \ldots & A_{nn} \end{pmatrix}$$

For example, if

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 2 \\ 3 & 3 & 4 \end{pmatrix}$$

then, $A_{11} = (-1)^2(3 \times 4 - 2 \times 3) = 6$, $A_{12} = (-1)^3(2 \times 4 - 3 \times 2) = -2$, …, or

$$\text{adj } \mathbf{A} = \begin{pmatrix} 6 & 1 & -5 \\ -2 & -5 & 4 \\ -3 & 3 & -1 \end{pmatrix}$$

A.2.5 Nonsingular Matrix

A matrix is of a rank r if the largest *square* array in the matrix having a nonzero determinant is of size r. A *square* matrix with a nonzero determinant is called a **full-rank** or **nonsingular** matrix. For example, consider

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 5 & 7 \end{pmatrix}$$

A is a **singular** matrix because

$$|\mathbf{A}| = 1 \times (21 - 20) - 2 \times (14 - 12) + 3 \times (10 - 9) = 0$$

But **A** has a rank $r = 2$ because

$$\begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} = -1 \neq 0$$

A.2.6 Inverse of a Nonsingular Matrix

If **B** and **C** are two n-square matrices such that $\mathbf{BC} = \mathbf{CB} = \mathbf{I}$, then **B** is called the inverse of **C** and **C** the inverse of **B**. The common notation for the inverse is $\mathbf{B}^{-1}$ and $\mathbf{C}^{-1}$.

Theorem *If* $\mathbf{BC} = \mathbf{I}$ *and* **B** *is* **nonsingular**, *then* $\mathbf{C} = \mathbf{B}^{-1}$, *which means that the inverse is unique.*

Proof. By assumption,

$$\mathbf{BC} = \mathbf{I}$$

then

$$\mathbf{B}^{-1}\mathbf{BC} = \mathbf{B}^{-1}\mathbf{I}$$

or

$$\mathbf{IC} = \mathbf{B}^{-1}$$

or

$$\mathbf{C} = \mathbf{B}^{-1}$$

Two important results can be proved for nonsingular matrices.

1. If **A** and **B** are nonsingular n-square matrices, then $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$
2. If **A** is nonsingular, then $\mathbf{AB} = \mathbf{AC}$ implies that $\mathbf{B} = \mathbf{C}$.

Matrix inversion is used to solve n linearly independent equations. Consider

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

where x_i represents the unknowns and a_{ij} and b_i are constants. These n equations can be written in matrix form as

$$\mathbf{AX} = \mathbf{b}$$

Because the equations are independent, **A** must be nonsingular. Thus

$$\mathbf{A}^{-1}\mathbf{AX} = \mathbf{A}^{-1}\mathbf{b}$$

or

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{b}$$

A.2.7 Methods of Computing the Inverse of Matrix[1]

Adjoint Matrix Method. Given **A**, a nonsingular matrix of size n,

$$\mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|}\text{adj }\mathbf{A} = \frac{1}{|\mathbf{A}|}\begin{pmatrix} A_{11} & A_{21} & \dots & A_{n1} \\ A_{12} & A_{22} & \dots & A_{n2} \\ \vdots & \vdots & \vdots & \vdots \\ A_{1n} & A_{2n} & \dots & A_{nn} \end{pmatrix}$$

For example, for

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 2 \\ 3 & 3 & 4 \end{pmatrix}$$

$$\text{adj }\mathbf{A} = \begin{pmatrix} 6 & 1 & -5 \\ -2 & -5 & 4 \\ -3 & 3 & -1 \end{pmatrix}, \ |\mathbf{A}| = -7$$

Hence

$$\mathbf{A}^{-1} = \frac{1}{-7}\begin{pmatrix} 6 & 1 & -5 \\ -2 & -5 & 4 \\ -3 & 3 & -1 \end{pmatrix} = \begin{pmatrix} -\frac{6}{7} & -\frac{1}{7} & \frac{5}{7} \\ \frac{2}{7} & \frac{5}{7} & -\frac{4}{7} \\ \frac{3}{7} & -\frac{3}{7} & \frac{1}{7} \end{pmatrix}$$

Row Operations (Gauss-Jordan) Method. Consider the partitioned matrix $(\mathbf{A}|\mathbf{I})$, where **A** is nonsingular. Premultiplying by $\mathbf{A}^{-1}$, we obtain

$$(\mathbf{A}^{-1}\mathbf{A}|\mathbf{A}^{-1}\mathbf{I}) = (\mathbf{I}|\mathbf{A}^{-1})$$

[1] TORA's inverse module is based on LU decomposition method. See Press and associates (1986).

Thus, applying a specific sequence of row transformations, **A** is changed to **I** and **I** is changed to $\mathbf{A}^{-1}$. To illustrate the procedure, consider the system of equations:

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 2 \\ 3 & 3 & 4 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix}$$

The solution of **X** and the inverse of basis matrix can be obtained directly by considering

$$\mathbf{A}^{-1}(\mathbf{A}|\mathbf{I}|\mathbf{b}) = (\mathbf{I}|\mathbf{A}^{-1}|\mathbf{A}^{-1}\mathbf{b})$$

The following iterations detail the transformation operation:

Iteration 0

$$\left(\begin{array}{ccc|ccc|c} 1 & 2 & 3 & 1 & 0 & 0 & 3 \\ 2 & 3 & 2 & 0 & 1 & 0 & 4 \\ 3 & 3 & 4 & 0 & 0 & 1 & 5 \end{array}\right)$$

Iteration 1

$$\left(\begin{array}{ccc|ccc|c} 1 & 2 & 3 & 1 & 0 & 0 & 3 \\ 0 & -1 & -4 & -2 & 1 & 0 & -2 \\ 0 & -3 & -5 & -3 & 0 & 1 & -4 \end{array}\right)$$

Iteration 2

$$\left(\begin{array}{ccc|ccc|c} 1 & 0 & -5 & -3 & 2 & 0 & -1 \\ 0 & 1 & 4 & 2 & -1 & 0 & 2 \\ 0 & 0 & 7 & 3 & -3 & 1 & 2 \end{array}\right)$$

Iteration 3

$$\left(\begin{array}{ccc|ccc|c} 1 & 0 & 0 & -\frac{6}{7} & -\frac{1}{7} & \frac{5}{7} & \frac{3}{7} \\ 0 & 1 & 0 & \frac{2}{7} & \frac{5}{7} & -\frac{4}{7} & \frac{6}{7} \\ 0 & 0 & 1 & \frac{3}{7} & -\frac{3}{7} & \frac{1}{7} & \frac{2}{7} \end{array}\right)$$

This gives $x_1 = \frac{3}{7}$, $x_2 = \frac{6}{7}$, and $x_3 = \frac{2}{7}$. The inverse of **A** is given by the right-hand-side matrix, which is the same as obtained by the method of adjoint matrix.

Product Form of the Inverse. Suppose that two nonsingular matrices, **B** and $\mathbf{B}_{\text{next}}$, differ exactly in one column. Further, assume that $\mathbf{B}^{-1}$ is given. Then the inverse $\mathbf{B}_{\text{next}}^{-1}$ can be computed using the formula

$$\mathbf{B}_{\text{next}}^{-1} = \mathbf{E}\mathbf{B}^{-1}$$

The matrix **E** is computed in the following manner. If the column vector $\mathbf{P}_j$ in **B** is replaced with the column vector $\mathbf{P}_r$ to produce $\mathbf{B}_{\text{next}}$, then **E** is constructed as an m-identity matrix with its rth column replaced by

$$\xi = \frac{1}{(\mathbf{B}^{-1}\mathbf{P}_j)_r}\begin{pmatrix} -(\mathbf{B}^{-1}\mathbf{P}_j)_1 \\ -(\mathbf{B}^{-1}\mathbf{P}_j)_2 \\ \vdots \\ +1 \\ \vdots \\ -(\mathbf{B}^{-1}\mathbf{P}_j)_m \end{pmatrix}_{\leftarrow r\text{th place}}, \quad (\mathbf{B}^{-1}\mathbf{P}_j)_r \neq 0$$

If $(\mathbf{B}^{-1}\mathbf{P}_j)_r = 0$, then $\mathbf{B}_{\text{next}}^{-1}$ does not exist.

To prove the validity of the formula $\mathbf{B}_{\text{next}}^{-1}$, define $\mathbf{F}$ as an m-identity matrix whose rth column is replaced by $\mathbf{B}^{-1}\mathbf{P}_j$—that is,

$$\mathbf{F} = (\mathbf{e}_1, \mathbf{e}_{r-1}, \mathbf{B}^{-1}\mathbf{P}_j, \mathbf{e}_{r+1}, \dots, \mathbf{e}_m)$$

Because $\mathbf{B}_{\text{next}}$ differs from $\mathbf{B}$ only in that its rth column is replaced with $\mathbf{P}_j$, then

$$\mathbf{B}_{\text{next}} = \mathbf{BF}$$

Thus,

$$\mathbf{B}_{\text{next}}^{-1} = (\mathbf{BF})^{-1} = \mathbf{F}^{-1}\mathbf{B}^{-1}$$

The formula follows by setting $\mathbf{E} = \mathbf{F}^{-1}$

The product form can be used to invert any nonsingular matrix, $\mathbf{B}$. Start with $\mathbf{B}_0 = \mathbf{I} = \mathbf{B}_0^{-1}$. Next, construct $\mathbf{B}_1$ as an identity matrix with its first column replaced with the first column in $\mathbf{B}$. Then

$$\mathbf{B}_1^{-1} = \mathbf{E}_1\mathbf{B}_0^{-1} = \mathbf{E}_1\mathbf{I} = \mathbf{E}_1$$

In general, if we construct $\mathbf{B}_i$ as an identity matrix with its first i columns replaced with the first i columns of $\mathbf{B}$, then

$$\mathbf{B}_i^{-1} = \mathbf{E}_i\mathbf{B}_{i-1}^{-1} = \mathbf{E}_i\mathbf{E}_{i-1}\mathbf{B}_{i-2}^{-1} = \dots = \mathbf{E}_i\mathbf{E}_{i-1}\dots\mathbf{E}_1$$

This means that for the original matrix $\mathbf{B}$,

$$\mathbf{B}^{-1} = \mathbf{E}_m\mathbf{E}_{m-1}\dots\mathbf{E}_1$$

The following example illustrates the application of the product form of the inverse. Consider

$$\mathbf{B} = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 4 & 0 & 1 \end{pmatrix}$$

Iteration 0

$$\mathbf{B}_0 = \mathbf{B}_0^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Iteration 1

$$\mathbf{B}_1 = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 4 & 0 & 1 \end{pmatrix}$$

$$\mathbf{B}_0^{-1}\mathbf{P}_1 = \mathbf{P}_1 = \begin{pmatrix} \mathbf{2} \\ 0 \\ 4 \end{pmatrix} \begin{matrix} \leftarrow r=1 \\ \\ \\ \end{matrix}$$

$$\mathbf{E}_1 = \begin{pmatrix} +\frac{1}{2} & 0 & 0 \\ -\frac{0}{2} & 1 & 0 \\ -\frac{4}{2} & 0 & 1 \end{pmatrix}$$

$$\mathbf{B}_1^{-1} = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}$$

Iteration 2

$$\mathbf{B}_2 = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 4 & 0 & 1 \end{pmatrix} = \mathbf{B}$$

$$\mathbf{B}_1^{-1}\mathbf{P}_2 = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \mathbf{2} \\ -2 \end{pmatrix} \begin{matrix} \\ \leftarrow r=2 \\ \\ \end{matrix}$$

$$\mathbf{E}_2 = \begin{pmatrix} 1 & -(\frac{1}{2})/2 & 0 \\ 0 & +1/2 & 0 \\ 0 & -(-2)/2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -\frac{1}{4} & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

$$\mathbf{B}^{-1} = \mathbf{B}_2^{-1} = \mathbf{E}_2\mathbf{B}_1^{-1} = \begin{pmatrix} 1 & -\frac{1}{4} & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{4} & 0 \\ 0 & \frac{1}{2} & 0 \\ -2 & 1 & 1 \end{pmatrix}$$

Partitioned Matrix Method. Suppose that the two n-nonsingular matrices $\mathbf{A}$ and $\mathbf{B}$ are partitioned as follows:

$$\mathbf{A} = \left(\begin{array}{c|c} \mathbf{A}_{11} & \mathbf{A}_{12} \\ (p \times p) & (p \times q) \\ \hline \mathbf{A}_{21} & \mathbf{A}_{22} \\ (q \times p) & (q \times q) \end{array}\right), \ \mathbf{A}_{11} \text{ nonsingular}$$

$$\mathbf{B} = \left(\begin{array}{c|c} \mathbf{B}_{11} & \mathbf{B}_{12} \\ (p \times p) & (p \times q) \\ \hline \mathbf{B}_{21} & \mathbf{B}_{22} \\ (q \times p) & (q \times q) \end{array}\right)$$

If $\mathbf{B}$ is the inverse of $\mathbf{A}$, then from $\mathbf{AB} = \mathbf{I}_n$, we have

$$\mathbf{A}_{11}\mathbf{B}_{11} + \mathbf{A}_{12}\mathbf{B}_{21} = \mathbf{I}_p$$
$$\mathbf{A}_{11}\mathbf{B}_{12} + \mathbf{A}_{12}\mathbf{B}_{22} = \mathbf{0}$$

Also, from $\mathbf{BA} = \mathbf{I}_n$, we get

$$\mathbf{B}_{21}\mathbf{A}_{11} + \mathbf{B}_{22}\mathbf{A}_{21} = \mathbf{0}$$
$$\mathbf{B}_{21}\mathbf{A}_{12} + \mathbf{B}_{22}\mathbf{A}_{22} = \mathbf{I}_q$$

Because $\mathbf{A}_{11}$ is nonsingular, $\mathbf{A}_{11}^{-1}$ exists. Solving for $\mathbf{B}_{11}$, $\mathbf{B}_{12}$, $\mathbf{B}_{21}$, and $\mathbf{B}_{22}$, we get

$$\begin{aligned}
\mathbf{B}_{11} &= \mathbf{A}_{11}^{-1} + (\mathbf{A}_{11}^{-1}\mathbf{A}_{12})\mathbf{D}^{-1}(\mathbf{A}_{21}\mathbf{A}_{11}^{-1})\\
\mathbf{B}_{12} &= -(\mathbf{A}_{11}^{-1}\mathbf{A}_{12})\mathbf{D}^{-1}\\
\mathbf{B}_{21} &= -\mathbf{D}^{-1}(\mathbf{A}_{21}\mathbf{A}_{11}^{-1})\\
\mathbf{B}_{22} &= \mathbf{D}^{-1}
\end{aligned}$$

where

$$\mathbf{D} = \mathbf{A}_{22} - \mathbf{A}_{21}(\mathbf{A}_{11}^{-1}\mathbf{A}_{12})$$

To illustrate the use of these formulas, partition the matrix

$$\mathbf{A} = \left(\begin{array}{c|cc} 1 & 2 & 3 \\ \hline 2 & 3 & 2 \\ 3 & 3 & 4 \end{array}\right)$$

such that

$$\mathbf{A}_{11} = (1),\ \mathbf{A}_{12} = (2, 3),\ \mathbf{A}_{21} = \begin{pmatrix} 2 \\ 3 \end{pmatrix},\ \mathbf{A}_{22} = \begin{pmatrix} 3 & 2 \\ 3 & 4 \end{pmatrix}$$

In this case, $A_{11}^{-1} = 1$ and

$$\mathbf{D} = \begin{pmatrix} 3 & 2 \\ 3 & 4 \end{pmatrix} - \begin{pmatrix} 2 \\ 3 \end{pmatrix}(1)(2, 3) = \begin{pmatrix} -1 & -4 \\ -3 & -5 \end{pmatrix}$$
$$\mathbf{D}^{-1} = -\tfrac{1}{7}\begin{pmatrix} -5 & 4 \\ 3 & -1 \end{pmatrix} = \begin{pmatrix} \frac{5}{7} & -\frac{4}{7} \\ -\frac{3}{7} & \frac{1}{7} \end{pmatrix}$$

Thus,

$$\mathbf{B}_{11} = (-\tfrac{6}{7}),\ \mathbf{B}_{12} = (-\tfrac{1}{7}\quad \tfrac{5}{7})$$
$$\mathbf{B}_{12} = \begin{pmatrix} \frac{2}{7} \\ \frac{3}{7} \end{pmatrix},\ \mathbf{B}_{22} = \begin{pmatrix} \frac{5}{7} & -\frac{4}{7} \\ -\frac{3}{7} & \frac{1}{7} \end{pmatrix}$$

which directly give $\mathbf{B} = \mathbf{A}^{-1}$

A.3 QUADRATIC FORMS

Given

$$\mathbf{X} = (x_1, x_2, \ldots, x_n)^T$$

and

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

the function

$$Q(\mathbf{X}) = \mathbf{X}^T\mathbf{A}\mathbf{X} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_ix_j$$

is called a quadratic form. The matrix $\mathbf{A}$ can always be assumed symmetric because each element of every pair of coefficients a_{ij} and $a_{ji}(i \neq j)$ can be replaced by $\frac{(a_{ij} + a_{ji})}{2}$ without changing $Q(\mathbf{X})$.

As an illustration, the quadratic form

$$Q(\mathbf{X}) = (x_1, x_2, x_3)\begin{pmatrix} 1 & 0 & 1 \\ 2 & 7 & 6 \\ 3 & 0 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

with unsymmetric $\mathbf{A}$ is the same as

$$Q(\mathbf{X}) = (x_1, x_2, x_3)\begin{pmatrix} 1 & 1 & 2 \\ 1 & 7 & 3 \\ 2 & 3 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

with symmetric $\mathbf{A}$. We will assume henceforth that $\mathbf{A}$ is always symmetric.

The quadratic form is said to be

1. *Positive-definite* if $Q(\mathbf{X}) > 0$ for all $\mathbf{X} \neq \mathbf{0}$.
2. *Positive-semidefinite* if $Q(\mathbf{X}) \geq 0$ for all $\mathbf{X}$, and there exists $\mathbf{X} \neq \mathbf{0}$ such that $Q(\mathbf{X}) = 0$.
3. *Negative-definite* if $-Q(\mathbf{X})$ is positive-definite.
4. *Negative-semidefinite* if $-Q(\mathbf{X})$ is positive-semidefinite.
5. *Indefinite* in all other cases.

It can be proved that the necessary and sufficient conditions for the realization of the preceding cases are

1. $Q(\mathbf{X})$ is positive-definite (semidefinite) if the values of the principal minor determinants of $\mathbf{A}$ are positive (nonnegative).[2] In this case, $\mathbf{A}$ is said to be positive-definite (semidefinite).

[2] The kth *principal minor* determinant of $\mathbf{A}_{n\times n}$ is defined by

$$\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & & \vdots \\ a_{k1} & a_{k2} & & a_{kk} \end{vmatrix}, \quad k = 1, 2, \ldots, n$$

2. $Q(\mathbf{X})$ is negative-definite if the value of kth principal minor determinants of $\mathbf{A}$ has the sign of $(-1)^k$, $k = 1, 2, \ldots, n$. In this case, $\mathbf{A}$ is called negative-definite.
3. $Q(\mathbf{X})$ is a negative-semidefinite if the kth principal minor determinant of $\mathbf{A}$ is either zero or has the sign of $(-1)^k$, $k = 1, 2, \ldots, n$.

A.4 CONVEX AND CONCAVE FUNCTIONS

A function $f(\mathbf{X})$ is said to be strictly convex if, for any two distinct points $\mathbf{X}_1$ and $\mathbf{X}_2$,

$$f(\lambda\mathbf{X}_1 + (1-\lambda)\mathbf{X}_2) < \lambda f(\mathbf{X}_1) + (1-\lambda)f(\mathbf{X}_2)$$

where $0 < \lambda < 1$. Conversely, a function $f(\mathbf{X})$ is strictly concave if $-f(\mathbf{X})$ is strictly convex.

A special case of the convex (concave) function is the quadratic form (see Section A.3)

$$f(\mathbf{X}) = \mathbf{CX} + \mathbf{X}^T\mathbf{AX}$$

where $\mathbf{C}$ is a constant vector and $\mathbf{A}$ is a symmetric matrix. It can be proved that $f(\mathbf{X})$ is strictly convex if $\mathbf{A}$ is positive-definite and $f(\mathbf{X})$ is strictly concave if $\mathbf{A}$ is negative-definite.

SELECTED REFERENCES

Hadley, G., *Matrix Algebra*, Addison-Wesley, Reading, MA, 1961.

Hohn, F., *Elementary Matrix Algebra*, 2nd ed., Macmillan, New York, 1964.

Press, W., B. Flannery, A. Teukolsky, and W. Vetterling, *Numerical Recipes: The Art of Scientific Computing,* Cambridge University Press, Cambridge, England, 1986.

PROBLEMS

A-1. Show that the following vectors are linearly dependent.

(a) $\begin{pmatrix}1\\-2\\3\end{pmatrix}\begin{pmatrix}-2\\4\\-2\end{pmatrix}\begin{pmatrix}1\\-2\\-1\end{pmatrix}$

(b) $\begin{pmatrix}2\\-3\\4\\5\end{pmatrix}\begin{pmatrix}4\\-6\\8\\10\end{pmatrix}$

A-2. Given

$$\mathbf{A} = \begin{pmatrix}1 & 4 & 9\\2 & 5 & -8\\3 & 7 & 2\end{pmatrix}, \mathbf{B} = \begin{pmatrix}7 & -1 & 2\\9 & 4 & 8\\3 & 6 & 10\end{pmatrix}$$

find

(a) $\mathbf{A} + 7\mathbf{B}$

(b) $2\mathbf{A} - 3\mathbf{B}$

(c) $(\mathbf{A} + 7\mathbf{B})^T$

A-3. In Problem A-2, show that $\mathbf{AB} \neq \mathbf{BA}$

A-4. Consider the partitioned matrices

$$\mathbf{A} = \left(\begin{array}{c|cc} 1 & 5 & 7 \\ 2 & -6 & 9 \\ \hline 3 & 7 & 2 \\ 4 & 9 & 1 \end{array}\right), \quad \mathbf{B} = \left(\begin{array}{cc|cc} 2 & 3 & -4 & 5 \\ \hline 1 & 2 & 6 & 7 \\ 3 & 1 & 0 & 9 \end{array}\right)$$

Find **AB** using partitioned matrix manipulation.

A-5. In Problem A-2, find $\mathbf{A}^{-1}$ and $\mathbf{B}^{-1}$ using the following:

(a) Adjoint matrix method

(b) Row operations method

(c) Product form of the inverse

(d) Partitioned matrix method

A-6. Consider

$$\mathbf{B} = \begin{pmatrix} 2 & 1 & 2 \\ 0 & 2 & 1 \\ 4 & 0 & 5 \end{pmatrix}, \quad \mathbf{B}^{-1} = \begin{pmatrix} \frac{5}{4} & -\frac{5}{8} & -\frac{3}{8} \\ \frac{1}{2} & \frac{1}{4} & -\frac{1}{4} \\ -1 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Suppose that the third vector $\mathbf{P}_3$ is replaced with the $\mathbf{V}_3 = \mathbf{P}_1 + 2\mathbf{P}_2$. This means that the resulting matrix is singular. Show how the product form of the inverse discovers the singularity of the matrix.

A-7. Use the product form of the inverse to verify whether each of the following equations has a unique solution, no solution, or infinity of solutions.

(a) $x_1 + 2x_2 = 3$

$x_1 + 4x_2 = 2$

(b) $x_1 + 2x_2 = 5$

$-x_1 - 2x_2 = -5$

(c) $x_1 + x_3 = 5$

$4x_1 + x_2 + 3x_3 = 8$

$x_1 + 3x_2 - 2x_3 = 3$

A-8. Verify the formulas given in Section A.2.7 for obtaining the inverse of a partitioned matrix.

A-9. Find the inverse of

$$\mathbf{A} = \begin{pmatrix} 1 & \mathbf{G} \\ \mathbf{H} & \mathbf{B} \end{pmatrix}, \ \mathbf{B} \text{ nonsingular}$$

A-10. Show that the following quadratic form is negative-definite.

$$Q(x_1, x_2) = 6x_1 + 3x_2 - 4x_1x_2 - 2x_1^2 - 3x_2^2$$

A-11. Show that the following quadratic form is positive-definite.

$$Q(x_1, x_2, x_3) = 2x_1^2 + 2x_2^2 + 3x_3^2 + 2x_1x_2 + 2x_2x_3$$

A-12. Show that the function $f(x) = e^x$ is strictly convex over all real values of x.

A-13. Show that the quadratic function

$$f(x_1, x_2, x_3) = 5x_1^2 + 5x_2^2 + 4x_3^2 + 4x_1x_2 + 2x_2x_3$$

is strictly convex.

A-14. In Problem A-13, show that $-f(x_1, x_2, x_3)$ is strictly concave.

APPENDIX B

TORA Primer

The TORA Optimization System is a Windows®-based software designed for use with many of the techniques presented in this book. An important feature of the system is that it can be used to solve problems in a tutorial or automated mode. The tutorial mode is particularly useful because it allows concentrating on the main concepts of the algorithms while relieving you of the burden of the tedious computations that generally characterize OR algorithms.

TORA is totally self-contained, in the sense that all the instructions needed to drive the software are represented by menus, command buttons, check boxes, and the like. It requires no user manual. Nevertheless, a summary of the basic features of the system will be given in this Appendix.

TORA is automated for screen display settings of 800 × 600 and 1024 × 768 pixels. The second setting is recommended because it produces a more proportionate layout of the screen.

B.1 MAIN MENU

Figure B.1 shows the Main Menu screen. A selection from this menu will lead to a new screen for selecting the input mode of the problem.

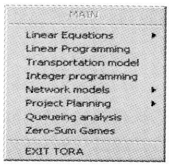


FIGURE B.1
Main menu screen

B.2 INPUT MODE AND FORMAT

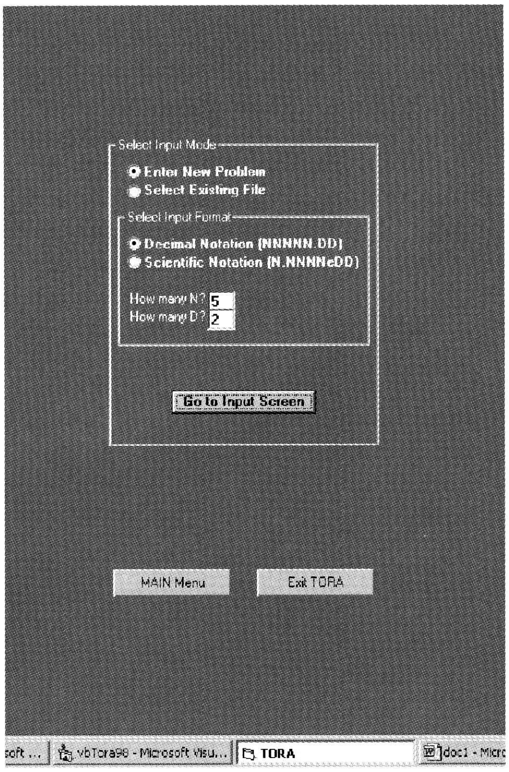
The input mode screen (Figure B.2) does two things:

1. It allows you to enter a new set of data for the problem (default) or read the data from an existing file that was originally created by TORA.
2. It allows you to select the format (decimal or scientific) as well as control the desired level of accuracy in inputting the data.

The decimal format (default) is represented by the code NNNNN.DD, whereas the scientific format is represented as .NNNNNeDD. The default values for the number of N's and the number of D's are 5 and 2, respectively. These values can be changed to any desired (reasonable) values.

FIGURE B.2

Input mode screen



Select Input Mode

● Enter New Problem

○ Select Existing File

Select Input Format

● Decimal Notation (NNNNN.DD)

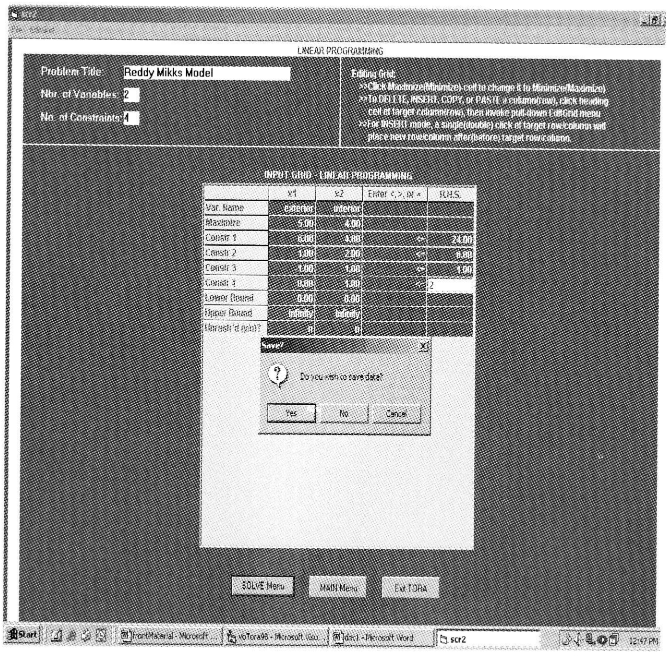
○ Scientific Notation (N.NNNNeDD)

How many N? 5

How many D? 2

B.3 INPUT DATA SCREEN

Inputting the appropriate size of the problem (top left of the input screen) automatically exposes the input data grid (Figure B.3). The grid entries are designed to match the data of the selected model (e.g., linear programming or transportation model). Regardless of the module used, the input grid is edited very much like a spreadsheet.



LINEAR PROGRAMMING

Problem Title: Reddy Mikks Model

Nbr. of Variables: 2

No. of Constraints: 4

Editing Grid:
>>Click Maximize(Minimize) cell to change it to Minimize(Maximize)
>>To DELETE, INSERT, COPY, or PASTE a column(row), click heading cell of target column(row), then invoke pull-down EditGrid menu
>>For INSERT mode, a single(double) click of target row/column will place new row/column after(before) target row/column.

INPUT GRID - LINEAR PROGRAMMING

	x1	x2	Enter <, >, or =	R.H.S.
Var. Name	exterior	interior		
Maximize	5.00	4.00		
Constr 1	6.00	4.00	<=	24.00
Constr 2	1.00	2.00	<=	6.00
Constr 3	-1.00	1.00	<=	1.00
Constr 4	0.00	1.00	<=	2
Lower Bound	0.00	0.00		
Upper Bound	infinity	infinity		
Unrestr'd (y/n)?	n	n		

Save?

Do you wish to save data?

FIGURE B.3
Input data screen

The design of the grid allows *inserting* or *deleting* a column or a row as well as *copying* and *pasting* the contents of a row or a column. To achieve this, first click the heading of the target column or row; then use EditGrid Menu to effect the desired result. The menu uses the suggestive key combinations CTRL+I, CTRL+D, CTRL+C, and CTRL+P for insert, delete, copy, and paste. Any of these operations may be undone using CTRL+U.

Once all data have been entered, press `Solve Menu` and follow instructions to save the data in a file, if desired.

B.4 SOLVE/MODIFY MENU

The Solve/Modify menu (Figure B.4) provides options for solving the selected problem. An important feature of TORA is that it allows solving the problem either automatically or in a tutorial (user-guided) mode. All these options are generated in a logical manner using submenus.

The Modify entry allows you to go back to the Input Data screen to make changes in the original data of the problem.

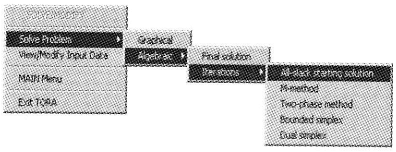
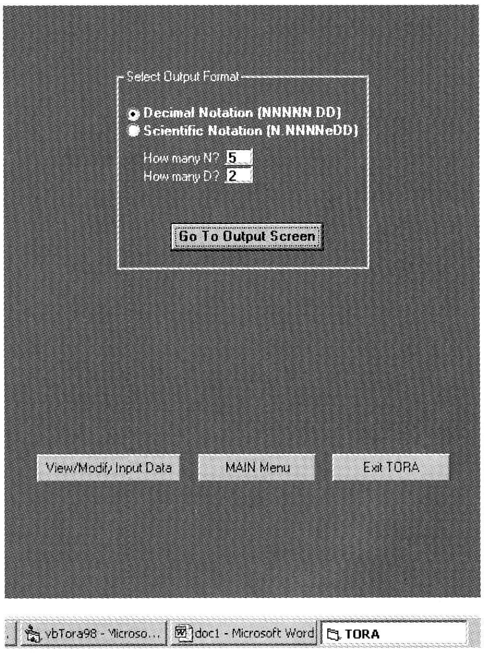


FIGURE B.4
Solve/modify screen

B.5 OUTPUT FORMAT

The Output Format screen (Figure B.5) controls the accuracy of the output results. The details for the output format are the same as in the input format (Section B.2).



Select Output Format

● Decimal Notation (NNNNN.DD)

○ Scientific Notation (N.NNNNeDD)

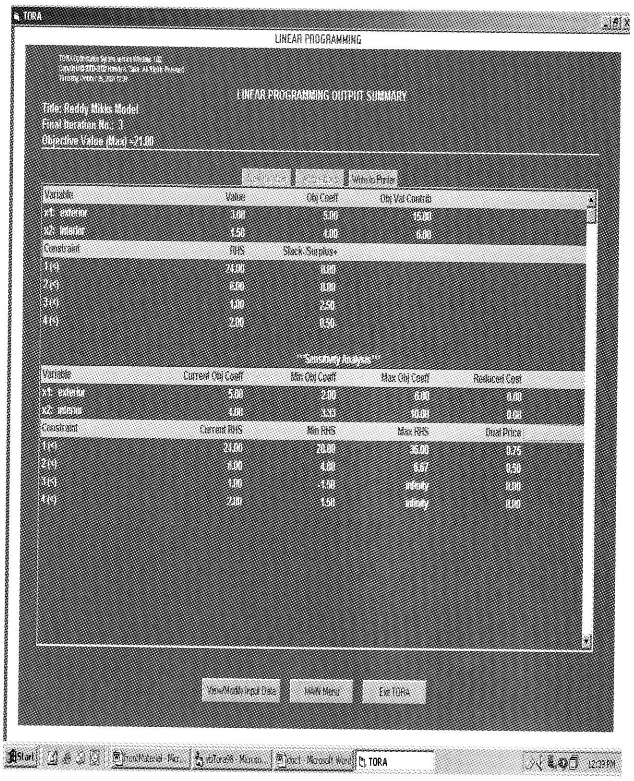
How many N? 5

How many D? 2

FIGURE B.5
Output format screen

B.6 OUTPUT RESULTS

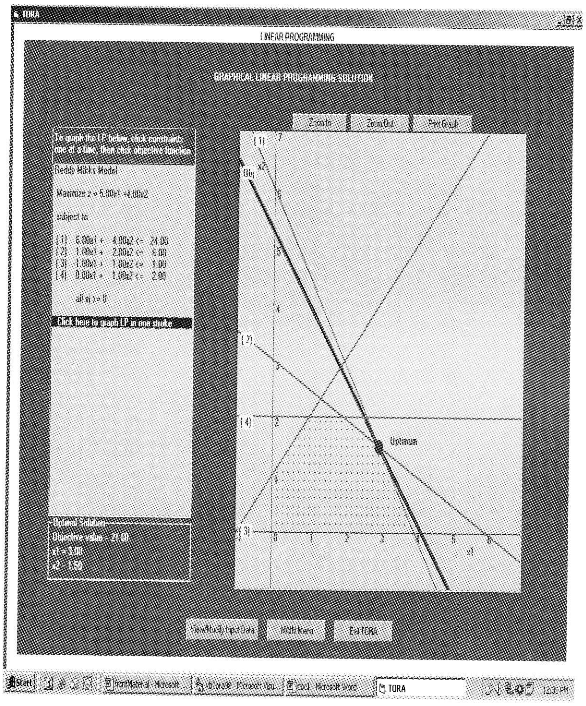
The output screen provides the results either in text format or graphically depending on the type of problem being solved (Figures B.6 and B.7). Both text and graphical results can be printed using a command button `Write to Printer`.



```
TORA
                         LINEAR PROGRAMMING

TORA Optimization System, version Windows 1.02
Copyright © 2000-2002 Hamdy A. Taha. All Rights Reserved
Thursday, October 24, 2001 12:39

              LINEAR PROGRAMMING OUTPUT SUMMARY
Title: Reddy Mikks Model
Final Iteration No.: 3
Objective Value (Max) =21.00

Variable                  Value        Obj Coeff     Obj Val Contrib
x1: exterior               3.00           5.00           15.00
x2: interior               1.50           4.00            6.00
Constraint                  RHS    Slack-/Surplus+
1 (<)                     24.00           0.00
2 (<)                      6.00           0.00
3 (<)                      1.00           2.50-
4 (<)                      2.00           0.50-

                        ***Sensitivity Analysis***
Variable         Current Obj Coeff   Min Obj Coeff   Max Obj Coeff   Reduced Cost
x1: exterior                  5.00            2.00            6.00           0.00
x2: interior                  4.00            3.33           10.00           0.00
Constraint             Current RHS         Min RHS         Max RHS     Dual Price
1 (<)                        24.00           20.00           36.00           0.75
2 (<)                         6.00            4.00            6.67           0.50
3 (<)                         1.00           -1.50        infinity           0.00
4 (<)                         2.00            1.50        infinity           0.00
```

FIGURE B.6

Text output screen



```
TORA
                         LINEAR PROGRAMMING
            GRAPHICAL LINEAR PROGRAMMING SOLUTION

To graph the LP below, click constraints
one at a time, then click objective function
Reddy Mikks Model

 Maximize z = 5.00x1 +4.00x2

 subject to

 (1)   6.00x1 +   4.00x2 <=  24.00
 (2)   1.00x1 +   2.00x2 <=   6.00
 (3)  -1.00x1 +   1.00x2 <=   1.00
 (4)   0.00x1 +   1.00x2 <=   2.00

       all xj >= 0

Click here to graph LP in one stroke

Optimal Solution
Objective value = 21.00
x1 = 3.00
x2 = 1.50
```

FIGURE B.7

Graphical output screen

APPENDIX C

Statistical Tables

TABLE C.1 Normal Distribution Function

$$F(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-(\frac{1}{2})t^2}\,dt$$

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.6	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.7	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.8	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
2.9	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
3.0	0.9987	0.9987	0.9987	0.9988	0.9988	0.9989	0.9989	0.9989	0.9990	0.9990
3.1	0.9990	0.9991	0.9991	0.9991	0.9992	0.9992	0.9992	0.9992	0.9993	0.9993
3.2	0.9993	0.9993	0.9994	0.9994	0.9994	0.9994	0.9994	0.9995	0.9995	0.9995
3.3	0.9995	0.9995	0.9995	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9997
3.4	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9998
3.5	0.9998									
4.0	0.99997									
5.0	0.9999997									
6.0	0.999999999									

Source: MILLER, I., and J. FREUND, *Probability and Statistics for Engineers,* Prentice Hall, Upper Saddle River, NJ, 1985.

TABLE C.2 $t_{\alpha,\nu}$ (student t) values*

ν	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.025$	$\alpha = 0.01$	$\alpha = 0.005$	ν
1	3.078	6.314	12.706	31.821	63.657	1
2	1.886	2.920	4.303	6.965	9.925	2
3	1.638	2.353	3.182	4.541	5.841	3
4	1.533	2.132	2.776	3.747	4.604	4
5	1.476	2.015	2.571	3.365	4.032	5
6	1.440	1.943	2.447	3.143	3.707	6
7	1.415	1.895	2.365	2.998	3.499	7
8	1.397	1.860	2.306	2.896	3.355	8
9	1.383	1.833	2.262	2.821	3.250	9
10	1.372	1.812	2.228	2.764	3.169	10
11	1.363	1.796	2.201	2.718	3.106	11
12	1.356	1.782	2.179	2.681	3.055	12
13	1.350	1.771	2.160	2.650	3.012	13
14	1.345	1.761	2.145	2.624	2.977	14
15	1.341	1.753	2.131	2.602	2.947	15
16	1.337	1.746	2.120	2.583	2.921	16
17	1.333	1.740	2.110	2.567	2.898	17
18	1.330	1.734	2.101	2.552	2.878	18
19	1.328	1.729	2.093	2.539	2.861	19
20	1.325	1.725	2.086	2.528	2.845	20
21	1.323	1.721	2.080	2.518	2.831	21
22	1.321	1.717	2.074	2.508	2.819	22
23	1.319	1.714	2.069	2.500	2.807	23
24	1.318	1.711	2.064	2.492	2.797	24
25	1.316	1.708	2.060	2.485	2.787	25
26	1.315	1.706	2.056	2.479	2.779	26
27	1.314	1.703	2.052	2.473	2.771	27
28	1.313	1.701	2.048	2.467	2.763	28
29	1.311	1.699	2.045	2.462	2.756	29
Inf.	1.282	1.645	1.960	2.326	2.576	inf.

*Abridged by permission of Macmillan Publishing Co., Inc., from *Statistical Methods for Research Workers,* 14th ed., by R. A. Fisher. Copyright © 1970 University of Adelaide.

TABLE C.3

ν	α =
1	0.0
2	0.0
3	0.0
4	0.2
5	0.4
6	0.6
7	0.9
8	1.3
9	1.7
10	2.1
11	2.6
12	3.0
13	3.5
14	4.0
15	4.6
16	5.1
17	5.6
18	6.2
19	6.8
20	7.4
21	8.0
22	8.6
23	9.2
24	9.8
25	10.5
26	11.1
27	11.8
28	12.4
29	13.1
30	13.7

*This table i

TABLE C.3 $\chi^2_{\alpha,\nu}$ (chi-square) values*

ν	$\alpha = 0.995$	$\alpha = 0.99$	$\alpha = 0.975$	$\alpha = 0.95$	$\alpha = 0.05$	$\alpha = 0.025$	$\alpha = 0.01$	$\alpha = 0.005$	ν
1	0.0000393	0.000157	0.000982	0.00393	3.841	5.024	6.635	7.879	1
2	0.0100	0.0201	0.0506	0.103	5.991	7.378	9.210	10.597	2
3	0.0717	0.115	0.216	0.352	7.815	9.348	11.345	12.838	3
4	0.207	0.297	0.484	0.711	9.488	11.143	13.277	14.860	4
5	0.412	0.554	0.831	1.145	11.070	12.832	15.056	16.750	5
6	0.676	0.872	1.237	1.635	12.592	14.449	16.812	18.548	6
7	0.989	1.239	1.690	2.167	14.067	16.013	18.475	20.278	7
8	1.344	1.646	2.180	2.733	15.507	17.535	20.090	21.955	8
9	1.735	2.088	2.700	3.325	16.919	19.023	21.666	23.589	9
10	2.156	2.558	3.247	3.940	18.307	20.483	23.209	25.188	10
11	2.603	3.053	3.816	4.575	19.675	21.920	24.725	26.757	11
12	3.074	3.571	4.404	5.226	21.026	23.337	26.217	28.300	12
13	3.565	4.107	5.009	5.892	22.362	24.736	27.688	29.819	13
14	4.075	4.660	5.629	6.571	23.685	26.119	29.141	31.319	14
15	4.601	5.229	6.262	7.261	24.996	27.488	30.578	32.801	15
16	5.142	5.812	6.908	7.962	26.296	28.845	32.000	34.267	16
17	5.697	6.408	7.564	8.672	27.587	30.191	33.409	35.718	17
18	6.265	7.015	8.231	9.390	28.869	31.526	34.805	37.156	18
19	6.844	7.633	8.907	10.117	30.144	32.852	36.191	38.582	19
20	7.434	8.260	9.591	10.851	31.410	34.170	37.566	39.997	20
21	8.034	8.897	10.283	11.591	32.671	35.479	38.932	41.401	21
22	8.643	9.542	10.982	12.338	33.924	36.781	40.289	42.796	22
23	9.260	10.196	11.689	13.091	35.172	38.076	41.638	44.181	23
24	9.886	10.856	12.401	13.484	36.415	39.364	42.980	45.558	24
25	10.520	11.524	13.120	14.611	37.652	40.646	44.314	46.928	25
26	11.160	12.198	13.844	15.379	38.885	41.923	45.642	48.290	26
27	11.808	12.879	14.573	16.151	40.113	43.194	46.963	49.645	27
28	12.461	13.565	15.308	16.928	41.337	44.461	48.278	50.993	28
29	13.121	14.256	16.047	17.708	42.557	45.772	49.588	52.336	29
30	13.787	14.953	16.791	18.493	43.773	46.979	50.892	53.672	30

*This table is based on Table 8 of *Biometrika Tables for Statisticians,* Vol. 1, by permission of the *Biometrika* trustees.

APPENDIX D

Partial Answers to Selected Problems

CHAPTER 1

Set 1.1a

4. (c) 17 minutes

5. (a) Jim's alternatives: Throw curve or fast ball.
Joe's alternatives: Prepare for curve or fast ball.

(b) Joe wants to increase his batting average.
Jim wants to reduce Joe's batting average.

CHAPTER 2

Set 2.1a

1. (a) $-x_1 + x_2 \geq 1$

(c) $x_1 - x_2 \leq 0$

(e) $.5x_1 - .5x_2 \geq 0$

3. (a) 4 tons/day

Set 2.2a

1. (a and e) See Figure D.1.

2. (a and d) See Figure D.2.

5. Let

$$x_1 = \text{play hours per day}$$

$$x_2 = \text{work hours per day}$$

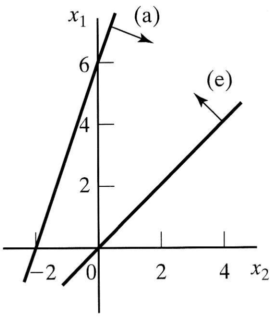


FIGURE D.1

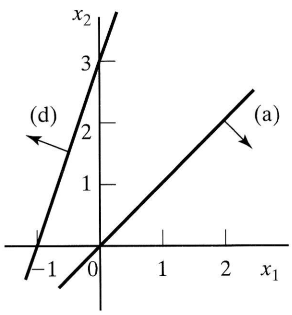


FIGURE D.2

Maximize $z = 2x_1 + x_2$ subject to

$$x_1 + x_2 \le 10,\ x_1 - x_2 \le 0$$

$$x_1 \le 4,\ x_1, x_2 \ge 0$$

Optimum: $(x_1, x_2) = (4, 6)$, $z = 14$

Set 2.2b

2. Optimum solution: $x_1 = 450$ lb, $x_2 = 350$ lb, $z = \$450$

5. Let

$$x_1 = \text{thousand bbl/day from Iran}$$

$$x_2 = \text{thousand bbl/day from Dubai}$$

Minimize $z = x_1 + x_2$ subject to

$$-.6x_1 + .4x_2 \le 0,\ .2x_1 + .1x_2 \ge 14$$

$$.25x_1 + .6x_2 \ge 30,\ .1x_1 + .15x_2 \ge 10$$

$$.15x_1 + .1x_2 \ge 8,\ x_1, x_2 \ge 0$$

Optimum: $x_1 = 55$, $x_2 = 30$, $z = 85$

Set 2.3a

1. (b) $\frac{-1}{1} \le \frac{c_2}{c_1} \le \frac{2}{3}$, $c_1 \neq 0$. See Figure D.3

3. Let

$$x_1 = \text{sold A1 cans per day}$$

$$x_2 = \text{sold B\&K cans per day}$$

Maximize $z = 5x_1 + 7x_2$ subject to

$$x_1 + x_2 \le 500,\ 2x_1 - x_2 \le 0,\ x_1 \ge 100$$

$$x_1, x_2 \ge 0$$

(a) $x_1 = 100$, $x_2 = 400$, $z = \$33$

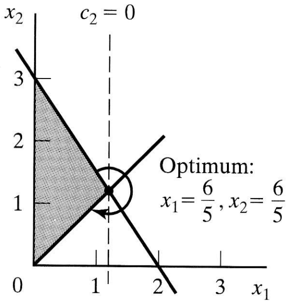


FIGURE D.3

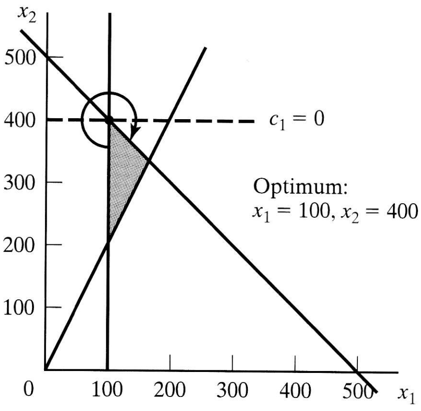


FIGURE D.4

(b) View $x_1 \geq 100$ as $\lim_{\delta \to 0}(x_1 - \delta x_2) \geq 100$. Hence, $\lim_{\delta \to 0} \frac{1}{-\delta} \leq \frac{c_1}{c_2} \leq \frac{1}{1}$, or, $-\infty \leq \frac{c_1}{c_2} \leq 1$. See Figure D.4.

7. Let

$$x_1 = \text{cases of juice per day}$$

$$x_2 = \text{cases of paste per day}$$

Maximize $z = 18x_1 + 9x_2$ subject to

$$24x_1 + 8x_2 \leq 60000$$

$$x_1 \leq 2000, \; x_2 \leq 6000$$

$$x_1, x_2 \geq 0$$

(a) $x_1 = 500, x_2 = 6000, z = \$63000$

(b) $0 \leq \frac{c_1}{c_2} \leq 3$, $c_2 \neq 0$, see Figure D.5.

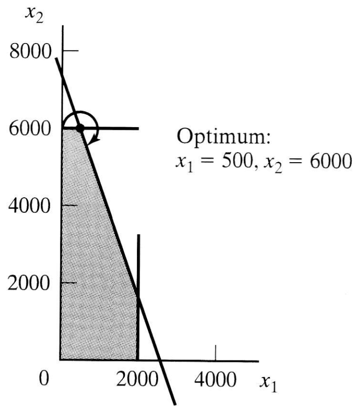


FIGURE D.5

Set 2.3b

1. Let

$$x_1 = \text{number of type 1 hats per day}$$

$$x_2 = \text{number of type 2 hats per day}$$

Maximize $z = 8x_1 + 5x_2$ subject to

$$2x_1 + x_2 \leq 400$$

$$x_1 \leq 150,\ x_2 \leq 200$$

$$x_1,\ x_2 \geq 0$$

(a) See Figure D.6. $x_1 = 100$, $x_2 = 200$, $z =$ \$1800 at point B

(b) \$4 per type 2 hat in the range (200,500)

(c) \$0 worth per unit in the range $(100, \infty)$, hence change to 120 has no effect

(d) \$1 worth per unit in the range (100, 400)

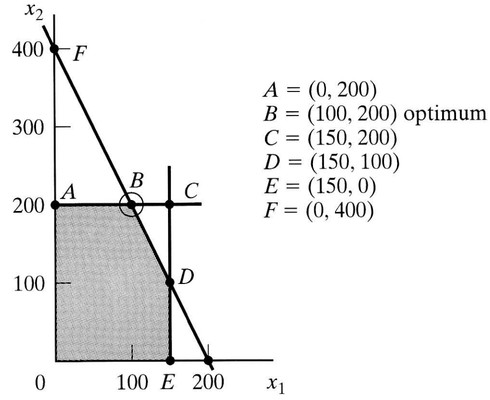


FIGURE D.6

4. Let

$$x_1 = \text{radio minutes}$$

$$x_2 = \text{TV minutes}$$

Maximize $z = x_1 + 25x_2$ subject to

$$15x_1 + 300x_2 \leq 10{,}000$$

$$-x_1 + 2x_2 \leq 0,\ x_1 \leq 400$$

$$x_1,\ x_2 \geq 0$$

(a) $x_1 = 60.61$, $x_2 = 30.3$, $z = 818.18$

(b) 0 per minute in the range $(60.61, \infty)$

(c) .082 worth per budget \$ in the range (0, 66000)

8. Let

$$x_1 = \text{units of solution } A$$

$$x_2 = \text{units of solution } B$$

Maximize $z = 8x_1 + 10x_2$ subject to

$$.5x_1 + .5x_2 \leq 150,\ .6x_1 + .4x_2 \leq 145$$

$$30 \leq x_1 \leq 150,\ 40 \leq x_2 \leq 200$$

$$x_1, x_2 \geq 0$$

(a) $x_1 = 100$, $x_2 = 200$, $z =$ \$2800

(b) Worth per unit of raw material I = \$16 in the range (115, 154.17)
Worth per unit of raw material II = \$0 in the range (140, ∞)

Set 2.4a

1. (a) One additional lb of feedstuff costs 55 cents.
(b) Total cost = \$495 per day.
(c) Current solution remains optimal.

3. (b) LINGO model:

```
MODEL:
    TITLE Diet model;
SETS:
    Constr: Rhs;
    Var: C, X;
    ConsVar(Constr,Var): Aij;
ENDSETS
    MIN=@SUMVar(j):C(j)*X(j) );      !obj function;
    @FOR(                              !constraints;
        Constr(i):@SUM( Var(j): Aij(i,j)*X(j) )>=Rhs(i)
        );
DATA:
     Constr=MinDemand Protein Fiber;
     Rhs=800 0 0;
     Var=X1 X2;
         C = .3 .9;          !obj func
       Aij = 1.  1.          !constr 1
           -.21 .30          !constr 2
            .03-.01;         !constr 3
ENDDATA
END
```

Set 2.5a

1. (a) Write the first two constraints as $x_1 + x_2 + x_3 + (x_4 + x_5) \leq 12$ and $(x_4 + x_5) \geq 4.8$.
(b) New $z = .936$ million dollars.

4. (a) 1150L ft^2
(b) (3,0,0), (1,1,0), and (1,0,1) with respective 0, 3, and 1 trim loss per foot.
(c) Number of standard 20′-rolls decreased by 30.

6. (a) Let

$$x_1 = \text{tons of brown sugar per week}$$

$$x_2 = \text{tons of white sugar per week}$$

$$x_3 = \text{tons of powdered sugar per week}$$

$$x_4 = \text{tons of molasses per week}$$

Maximize $z = 150x_1 + 200x_2 + 230x_3 + 35x_4$ subject to

$$.76x_1 + .95x_2 + x_3 \leq 912$$

$$x_1 \geq 25,\ x_2 \geq 25,\ x_3 \geq 25,\ 0 \leq x_4 \leq 400$$

Optimum solution: $x_1 = 25$, $x_2 = 25$, $x_3 = 869.25$, $x_4 = 400$, $z = \$222,677.50$.

(b) Worth per unit of syrup = \$55.94 valid in the range $(187.15, \infty)$

9. (a) Let

$$x_i = \text{dollars invested in project } i,\ i = 1, 2, 3, 4$$

$$y_j = \text{dollars invested in bank in year } j,\ j = 1, 2, 3, 4$$

Maximize $z = y_5$ subject to

$$x_1 + x_2 + x_4 + y_1 \leq 10{,}000$$

$$.5x_1 + .6x_2 - x_3 + .4x_4 + 1.065y_1 - y_2 = 0$$

$$.3x_1 + .2x_2 + .8x_3 + .6x_4 + 1.065y_2 - y_3 = 0$$

$$1.8x_1 + 1.5x_2 + 1.9x_3 + 1.8x_4 + 1.065y_3 - y_4 = 0$$

$$1.2x_1 + 1.3x_2 + .8x_3 + .95x_4 + 1.065y_4 - y_5 = 0$$

$$x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4 \geq 0$$

Optimum solution: $x_1 = 0$, $x_2 = \$10{,}000$, $x_3 = \$6000$, $x_4 = 0$, $y_1 = 0$, $y_2 = 0$, $y_3 = \$6800$, $y_4 = \$33{,}642$, $z = \$53{,}628.73$ at the start of year 5.

(b) 5.36%

(c) Total return reduced by 1000 × .373 = \$3730

12. (a) Let

$$x_j = \text{number of units of model } j,\ j = 1, 2, 3$$

Maximize $z = 30x_1 + 20x_2 + 50x_3$ subject to

$$2x_1 + 3x_2 + 5x_3 \leq 4000$$

$$4x_1 + 2x_2 + 7x_3 \leq 6000$$

$$x_1 + .5x_2 + .33x_3 \leq 1500$$

$$2x_1 - 3x_2 = 0$$

$$5x_2 - 2x_3 = 0$$

$$x_1 \geq 200,\ x_2 \geq 200,\ x_3 \geq 150$$

$$x_1, x_2, x_3 \geq 0$$

Optimum solution: $x_1 = 324.32$, $x_2 = 216.22$, $x_3 = 540.54$, $z = \$41{,}081.08$.

(b) Not advisable because dual price = \$10.27 per lb

(c) Not advisable because dual price = \$0 per lb

15. (a) Let

$$x_{iA} = \text{amount invested in year } i \text{ using plan } A,\ i = 1, 2, 3$$

$$x_{iB} = \text{amount invested in year } i \text{ using plan } B,\ i = 1, 2, 3$$

Maximize $z = 3x_{2B} + 1.7x_{3A}$ subject to

$$x_{1A} + x_{1B} \leq 100$$

$$-1.7x_{1A} + x_{2A} + x_{2B} \leq 0$$

$$-3x_{1B} - 1.7x_{2A} + x_{3A} \leq 0$$

$$x_{iA}, x_{iB} \geq 0,\ i = 1, 2, 3$$

Optimum solution: Invest \$100,000 in plan A in year 1 and \$170,000 in plan B in year 2. Accumulation = \$510,000.

(b) Yes, each additional dollar of investment is worth \$5.10 at the end of 3 years.

CHAPTER 3

Set 3.1a

1. 2 tons/day and 1 ton/day for raw materials $M1$ and $M2$.

4. Let x_{ij} = units of product i produced on machine j, $i = 1, 2$; $j = 1, 2$

Maximize $z = 10(x_{11} + x_{12}) + 15(x_{21} + x_{22})$ subject to

$$x_{11} + x_{21} - x_{12} - x_{22} + s_1 = 5$$

$$-x_{11} - x_{21} + x_{12} + x_{22} + s_2 = 5$$

$$x_{11} + x_{21} + s_3 = 200$$

$$x_{12} + x_{22} + s_4 = 250$$

$$x_{ij} \geq 0, \text{ for all } i \text{ and } j,\ s_i \geq 0,\ i = 1, 2, 3, 4$$

Set 3.1b

2. Let x_j = units of product j, $j = 1, 2, 3$.

Maximize $z = 2x_1 + 5x_2 + 3x_3 - 15x_4^- - 10x_5^-$ subject to

$$2x_1 + x_2 + 2x_3 + x_4^+ - x_4^- = 80$$

$$x_1 + x_2 + 2x_3 + x_5^+ - x_5^- = 65$$

$$x_1, x_2, x_3, x_4^+, x_4^-, x_5^+, x_5^- \geq 0$$

Optimum solution: $x_1 = 0$ units, $x_2 = 65$ units, all others = 0, $z = \$325$.

Set 3.2a

2. (c) $x_1 = \frac{6}{7}$, $x_2 = \frac{12}{7}$, $z = \frac{48}{7}$.
(e) $(x_1 = 0, x_2 = 3)$ and $(x_1 = 6, x_2 = 0)$.

4. Infeasible basic solutions are:

$$(x_1, x_2) = (\tfrac{26}{3}, -\tfrac{4}{3}), (x_1, x_3) = (8, -2)$$
$$(x_1, x_4) = (6, -4), (x_2, x_3) = (16, -26)$$
$$(x_2, x_4) = (3, -13), (x_3, x_4) = (6, -16)$$

Set 3.3a

3. (a) All pairs but (A, B) because associated corner points are not adjacent.
(b) (i) Legitimate. (ii) Not legitimate (C and I not adjacent). (iii) Not legitimate (A is revisited).

5. (a) x_3 enters at value 1, $z = 3$.

Set 3.3b

3.

Basic variable	x_1	x_2	x_3	x_4
Value	1.5	1	0	.8
Leaving variable	x_7	x_7	x_8	x_5

6. (b) x_2, x_5, and x_6 can increase value of z. If x_2 enters, $\Delta z = +20$. If x_5 enters, $\Delta z = 0$. If x_6 enters, $\Delta z = \infty$.

9. Next best value of $z = 20$.

Set 3.4a

3. (a) Minimize $z = (8M - 4)x_1 + (6M - 1)x_2 - Ms_2 - Ms_3 = 10M$
(b) Minimize $z = (3M - 4)x_1 + (M - 1)x_2 = 3M$

6. The starting tableau is

Basic	x_1	x_2	x_3	x_4	Solution
z	−1	−12	0	0	−8
x_3	1	1	1	0	4
x_4	1	4	0	1	8

Set 3.4b

1. Always minimize the sum of artificials because it represents the amount of infeasibility in the problem.

7. Any nonbasic variable having nonzero objective coefficients at end of Phase I cannot become positive in Phase II because it will mean that the optimal objective value in Phase I will be positive—that is, it leads to an infeasible Phase I solution.

Set 3.5a

1. (a) $A \rightarrow B \rightarrow C \rightarrow D$.
(b) 1 at A, 1 at B, $C_2^3 = 3$ at C, and 1 at D.

Set 3.5b

1. Alternative basic optima: $(0, 0, \frac{10}{3})$, $(1, 0, 3)$, $(0, 5, 0)$. Nonbasic alternative optima: $(\alpha_2, 5\alpha_3, \frac{10}{3}\alpha_1 + 3\alpha_2)$, $\alpha_1 + \alpha_2 + \alpha_3 = 1$, $0 \leq \alpha_i \leq 1$, $i = 1, 2, 3$.

Set 3.5c

2. (a) Solution space is unbounded in the direction of x_2.
(b) Objective value is unbounded because a unit increase in x_2 increases z by 10.

Set 3.5d

1. The most that can be produced is 275 units.

CHAPTER 4

Set 4.1a

2. Let y_1, y_2, and y_3 be the dual variables.
Maximize $w = 3y_1 + 5y_2 + 4y_3$ subject to

$$y_1 + 2y_2 + 3y_3 \leq 15,\ 2y_1 - 4y_2 + y_3 \leq 12$$

$$y_1 \geq 0,\ y_2 \leq 0,\ y_3 \text{ unrestricted}$$

4. (c) Let y_1 and y_2 be the dual variables.
Minimize $z = 5y_1 + 6y_2$ subject to

$$2y_1 + 3y_2 = 1,\ y_1 - y_2 = 1$$

$$y_1,\ y_2 \text{ unrestricted}$$

5. Dual constraint associated with the artificial variables is $y_2 \geq -M$, which is the same as y_2 being unrestricted.

Set 4.2a

1. (a) $\mathbf{AV}_1$ is undefined.
(e) $\mathbf{V}_2\mathbf{A} = (-14 \quad -32)$

Set 4.2b

2. (a) Inverse $= \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ -\frac{4}{3} & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{pmatrix}$

Set 4.2c

2. Let y_1 and y_2 be the dual variables.
Minimize $w = 30y_1 + 40y_2$ subject to

$$y_1 + y_2 \geq 5,\ 5y_1 - 5y_2 \geq 2,\ 2y_1 - 6y_2 \geq 3$$

$$y_1 \geq -M,\ y_2 \geq 0$$

Solution: $y_1 = 5,\ y_2 = 0,\ w = 150$.

4. (a) Let y_1 and y_2 be the dual variables.
Minimize $w = 3y_1 + 4y_2$ subject to

$$y_1 + 2y_2 \geq 1,\ 2y_1 - y_2 \geq 5,\ y_1 \geq 3$$

$$y_2 \text{ unrestricted}$$

(b) Solution: $y_1 = 3,\ y_2 = -1,\ w = 5$

Set 4.2d

2. (a) Feasibility: $(x_2, x_4) = (3, 15) \Rightarrow$ feasible
Optimality: Objective coefficients of $(x_1, x_3) = (0, 2) \Rightarrow$ optimal

7. (a) $b_1 = 30,\ b_2 = 40$.
(b) $a = 23,\ b = 5,\ c = -10,\ d = 5,\ e = 0$

Set 4.2e

2. (a) Both primal and dual are infeasible.
(b) Solutions are feasible but not optimal.

Set 4.3a

2. (a) Let (x_1, x_2, x_3, x_4) = daily units of SC320, SC325, SC340, and SC370
Maximize $z = 9.4x_1 + 10.8x_2 + 8.75x_3 + 7.8x_4$ subject to

$$\begin{aligned}
10.5x_1 + 9.3x_2 + 11.6x_3 + 8.2x_4 &\leq 4800 \\
20.4x_1 + 24.6x_2 + 17.7x_3 + 26.5x_4 &\leq 9600 \\
3.2x_1 + 2.5x_2 + 3.6x_3 + 5.5x_4 &\leq 4700 \\
5x_1 + 5x_2 + 5x_3 + 5x_4 &\leq 4500 \\
x_1 \geq 100,\ x_2 \geq 100,\ x_3 \geq 100,\ x_4 &\geq 100
\end{aligned}$$

Optimum solution: $x_1 = 100,\ x_2 = 100,\ x_3 = 138.42,\ x_4 = 100,\ z = 4011.16$.

(b) Only soldering capacity can be increased because it has a positive dual price (=.4944).

(c) Dual prices are negative or zero. Hence, lower bounds represent disadvantages.

Set 4.3b

2. New fire truck toy is profitable because its reduced cost $= y_1 + 3y_3 - 4 = -2$.

Set 4.4a

1. (a) No, because point E is feasible and the dual simplex must stay infeasible until the last iteration where it becomes feasible.

4. (c) Add the artificial constraint $x_1 \le M$. Problem has no feasible solution.

Set 4.5a

4. Let Q be the weekly feed. Optimum solution: Limestone = .028Q, corn = .649Q, and soybean meal = .323Q. Cost = .81221Q.

Set 4.5b

1. (a) $-20 \le D_2 \le 400$, $D_3 \ge -20$.

5. (a) Scarce: resistor and capacitor resource, abundant: chips resource.

(b) Worth per unit of resistor, capacitor, and chips is \$1.25, \$.25, and \$0.

(g) Increase in profit = \$250. Additional cost = \$200. Net profit = \$50.

8. (b) Solution $x_1 = x_2 = 2 + \frac{\Delta}{3}$ is feasible for all $\Delta > 0$. For $0 < \Delta < 3$, $r_1 + r_2 = \frac{\Delta}{3} < 1 \Rightarrow$ feasibility confirmed. For $3 \le \Delta \le 6$, $r_1 + r_2 = \frac{\Delta}{3} \ge 1 \Rightarrow$ feasibility not confirmed. For $\Delta > 6$, the change falls outside the ranges for D_1 and D_2.

Set 4.5c

1. (a) Additional constraint, $4x_1 + x_2 + 2x_3 \le 570$, is redundant.

Set 4.5d

2. (a) Current solution remains optimal.

(c) New solution: $x_1 = 2$, $x_2 = 2$, $x_3 = 4$, $z = 14$.

Set 4.5e

2. (b) Optimum remains the same.

(d) Optimum changes: $x_1 = 10$, $x_2 = 102.5$, $x_3 = 215$, $z = 665$.

6. (b) Smallest unit profit for product 1 = \$6.

(c) New solution: $x_1 = 0$, $x_2 = 165$, $x_3 = 10$, $z = 4105$.

9. (a) $1.25 - .25d_1 + .5d_2 \ge 0$, $.25 + .75d_1 - .5d_2 \ge 0$.

Set 4.5f

1. 42.86%.

3. (a) Fire engines are not profitable.

CHAPTER 5

Set 5.1a

4. Assign a very high cost, M, to the route from Detroit to dummy destination.

6. (a and b) Use $M = 10{,}000$. Solution is shown in bold. Total cost = \$49,710.

	1	2	3	Supply
Plant 1	600	700	400 **25**	**25**
Plant 2	320 **23**	300 **17**	350	**40**
Plant 3	500	480 **25**	450 **5**	**30**
Excess Plant 4	1000 **13**	1000	*M*	**13**
Demand	**36**	**42**	**30**	

(c) City 1 excess cost = \$13,000.

9. Solution (in million gallons) is shown in bold. Area 2 will be 2 million gallons short. Total cost = \$304,000.

	A1	A2	A3	Supply
Refinery 1	12 **4**	18 **2**	*M*	**6**
Refinery 2	30	10 **4**	8 **1**	**5**
Refinery 3	20	25	12 **6**	**6**
Dummy	*M*	50 **2**	50	**2**
Demand	**4**	**8**	**7**	

Set 5.2a

2. Total cost = $804.

		Sharpening service			
Day	*New*	*Overnight*	*2-day*	*3-day*	*Disposal*
Monday	24	0	6	18	0
Tuesday	12	12	0	0	0
Wednesday	2	14	0	0	0
Thursday	0	0	20	0	0
Friday	0	14	0	0	4
Saturday	0	2	0	0	12
Sunday	0	0	0	0	22

5. Total cost = $190,040. Problem has alternative optima.

Period	Capacity	Produced amount	Delivery
1	500	500	400 for (period) 1 and 100 for 2
2	600	600	200 for 2, 220 for 3, and 180 for 4
3	200	200	200 for 3
4	300	200	200 for 4

Set 5.3a

1. (a) Northwest: cost = $42. Least-cost: cost = $37. Vogel: cost = $37.

Set 5.3b

5. (a) Cost = $1475.
(b) $c_{12} \geq 3$, $c_{13} \geq 8$, $c_{23} \geq 13$, $c_{31} \geq 7$.

Set 5.4a

5. Use the code (city, date) to define the rows and columns of the assignment problem. Example: The assignment (D, 3)-(A,7) means leaving Dallas on June 3 and returning from Atlanta June 7 at a cost of $400. Solution is shown in bold. Cost = $1180. Problem has alternative optima.

	(A, 7)	(A, 12)	(A, 21)	(A, 28)
(D, 3)	400	300	300	**280**
(D, 10)	**300**	400	300	300
(D, 17)	300	**300**	400	300
(D, 25)	300	300	**300**	400

6. Optimum assignment: I-*d*, II-*c*, III-*a*, IV-*b*.

Set 5.5a

4. Total cost = \$1550. Optimum solution summarized below. Problem has alternative optima.

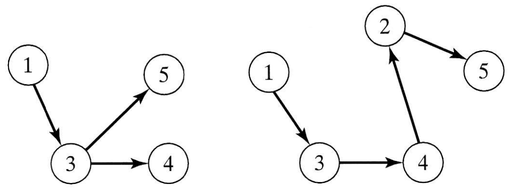
	Store 1	Store 2	Store 3
Factory 1	50	0	0
Factory 2	50	200	50

CHAPTER 6

Set 6.1a

1. (i)(a) 1-3-4-2. (b) 1-5-4-3-1. (c) 1-3-4-5-1. (d) See Figure D.7. (e) See Figure D.7.
4. 1 and 8 must appear in center. Problem has more than one solution. See Figure D.8.

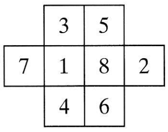
FIGURE D.7



Tree

Spanning tree

FIGURE D.8



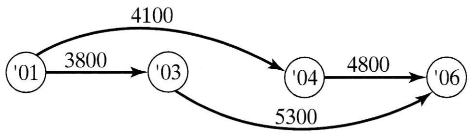
Set 6.2a

2. (a) 1-2-5-6-4-3 or 3-1-2-5-6-4. Total length = 14 miles.
5. High pressure: 1-2-3-4-6. Low pressure: 1-5-7 and 5-9-8. Total length = 53 miles

Set 6.3a

1. Buy new car in 2001 and 2004. Total cost = \$8900. See Figure D.9.
5. For arc (i, v_i) - $(i + 1, v_{i+1})$, define $p(q)$ = value (units of item i). Solution: Select items 1 and 2. Total value = \$80. See Figure D.10.

FIGURE D.9



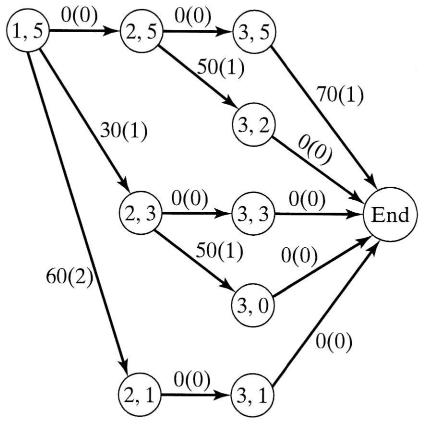


FIGURE D.10

Set 6.3b

1. (c) Delete all nodes but 4, 5, 6, 7, and 8. Shortest distance = 8 associated with routes 4-5-6-8 and 4-6-8.

Set 6.3c

1. (a) 5-4-2-1, distance = 12.
5. See formulation in Figure D.11. Each arc has unit length. Arrows show one-way routes. Example solution: Bob to Joe: Bob-Kay-Rae-Kim-Joe. Largest number of contacts = 4.

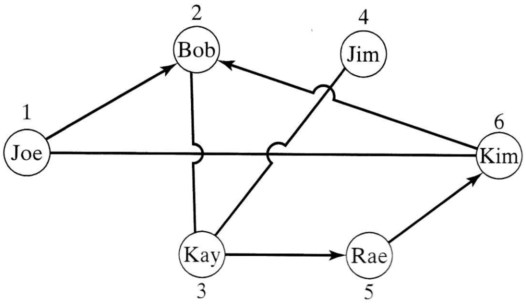


FIGURE D.11

Set 6.3d

1. (a) Formulation 1: Right-hand side of nodes 1 and 5 equations are −1 and 1; all others = 0. Formulation 2: Objective function is minimize $y_5 - y_1$. Optimum solution: 1-3-4-5, distance = 90.

Set 6.4a

1. Cut 1: 1-2, 1-4, 3-4, 3-5, capacity = 60.

Set 6.4b

1. (a) Surplus capacities: arc (2-3) = 40, arc(2-5) = 10, arc(4-3) = 5.
(b) Node 2: 20 units, node 3: 30 units, node 4: 20 units.
(c) No, because there is no surplus capacity out of node 1.

7. Maximum number of chores is 4. Rif-3, Mai-1, Ben-2, Kim-5. Ken has no chore.

Set 6.5a

1. See Figure D.12.

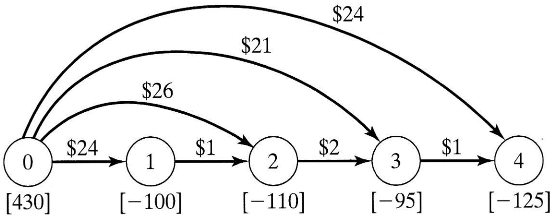


FIGURE D.12

Set 6.5b

1. Case 1: Lower bound is not substituted out.

	x_{12}	x_{13}	x_{24}	x_{32}	x_{34}	
Minimize z	1	5	3	4	6	
Node 1	1	1				= 50
Node 2	−1		1	−1		= −40
Node 3		−1		1	1	= 20
Node 4			−1		−1	= −30
Lower bound	0	30	10	10	0	
Upper bound	∞	40	∞	∞	∞	

Case 2: Lower bound is substituted out.

	x_{12}	x'_{13}	x'_{24}	x'_{32}	x_{34}	
Minimize z	1	5	3	4	6	
Node 1	1	1				= 20
Node 2	−1		1	−1		= −40
Node 3		−1		1	1	= 40
Node 4			−1		−1	= −20
Upper bound	∞	10	∞	∞	∞	

Set 6.5c

1. Optimum cost = \$9895. Produce 210 units in period 1 and 220 units in period 3.
5. Optimal solution: Total student miles = 24,300. Problem has alternative optima.

	Number of students	
	School 1	*School 2*
Minority area 1	0	500
Minority area 2	450	0
Minority area 3	0	300
Nonminority area 1	1000	0
Nonminority area 2	0	1000

Set 6.6a

3. See Figure D.13.

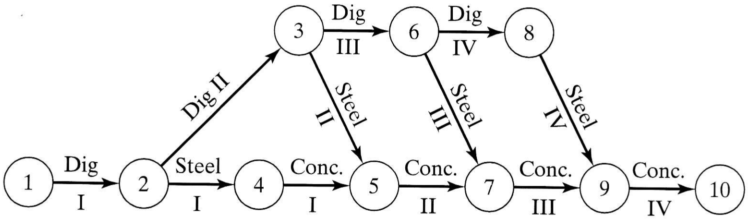


FIGURE D.13

Set 6.6b

1. Critical path: 1-3-4-5-6-7. Project duration = 19.
3. Project duration = 38 days.

Set 6.6c

3. (a) Maximum delay = 10.
5. (a) Critical path: 1-3-6, duration = 45 days.
   (b) Red-flagged activities: A, D, and E.
   (c) Start of C, D, and G will be delayed by 5 days. E will not be affected.
   (d) Minimum equipment = 2 units.

CHAPTER 7

Set 7.1a

2. Points (1, 0) and (0, 2) are in Q, but $\lambda(1,0) + (1 - \lambda)(0, 2) = (\lambda, 2 - 2\lambda)$ does not lie in Q for $0 < \lambda < 1$.

Set 7.1b

2. (b) Unique solution, see Figure D.14.

(d) Infinity of solutions.

(f) No solution.

3. (a) Basis because det $\mathbf{B} = -4$.

(d) Not a basis because a basis must include exactly three independent vectors.

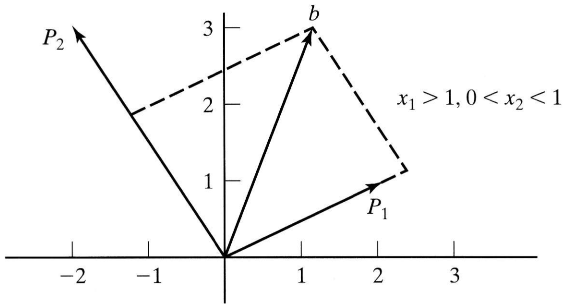


FIGURE D.14

Set 7.1c

1. $\mathbf{X}_B = (x_3, x_4)^T = (2, 1.5)^T$, which is feasible.

4. Optimal $z = 34$.

Maximize $z = 2x_1 + 5x_2$ subject to

$$x_1 \leq 4,\ x_2 \leq 6,\ x_1 + x_2 \leq 8,\ x_1, x_2 \geq 0$$

Set 7.2a

1. (a) $\mathbf{P}_1$ must leave.

(b) $\mathbf{B} = (\mathbf{P}_2, \mathbf{P}_4)$ is a feasible basis.

2. For the basic vector $\mathbf{X}_B$, we have

$$\text{For } \mathbf{X}_B,\ \{z_j - c_j\} = \mathbf{C}_B\mathbf{B}^{-1}\mathbf{B} - \mathbf{C}_B = \mathbf{C}_B\mathbf{I} - \mathbf{C}_B = \mathbf{C}_B - \mathbf{C}_B = \mathbf{0}$$

7. Number of adjacent extreme points is $n - m$ assuming nondegeneracy.

10. In case of degeneracy, number of extreme points is less than the number of basic solutions, else they must be equal.

11. (a) new $x_j = \frac{1}{\alpha}$ old x_j

(b) new $x_j = \frac{\beta}{\alpha}$ old x_j

Set 7.2b

2. (b) $(x_1, x_2, x_3) = (1.5, 2, 0)$, $z = 5$

Set 7.3a

2. $(x_1, x_2, x_3, x_4, x_5, x_6) = (0, 1, .75, 1, 0, 1)$, $z = 22$

Set 7.4a

1. (c) Add the artificial constraint $x_2 \leq M$. Then

$$(x_1, x_2) = \alpha_1(0, 0) + \alpha_2(10, 0) + \alpha_3(20, 10) + \alpha_4(20, M) + \alpha_5(0, M)$$

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 = 1, \alpha_j \geq 0, j = 1, 2, \ldots, 5$$

2. Subproblem 1: $(x_1, x_2) = \alpha_1(0, 0) + \alpha_2(\frac{12}{5}, 0) + \alpha_3(0, 12)$
Subproblem 2: $(x_3, x_4) = \beta_1(5, 0) + \beta_2(50, 0) + \beta_3(0, 10) + \beta_4(0, 5)$
Optimal solution: $\alpha_1 = \alpha_2 = 0, \alpha_3 = 1 \Rightarrow x_1 = 0, x_2 = 12$

$$\beta_1 = .4889, \beta_2 = .5111, \beta_3 = \beta_4 = 0 \Rightarrow x_4 = 28, x_5 = 0$$

6. Because the original problem is minimization, we must maximize each subproblem. Optimal solution: $(x_1, x_2, x_3, x_4) = (\frac{5}{3}, \frac{10}{3}, 0, 20)$, $z = -\frac{245}{3}$

Set 7.5a

2. Maximize $w = \mathbf{Yb}$ subject to $\mathbf{YA} \leq \mathbf{C}, \mathbf{Y} \geq \mathbf{0}$

Set 7.5b

5. Method 1: Given $\mathbf{X}_B = (2, 6, 2)^T$, then $(b_1, b_2, b_3) = (4, 6, 8) \Rightarrow$ dual objective value $= 34$

Method 2: Given $\mathbf{Y} = (0, 3, 2)^T$, then $(c_1, c_2) = (2, 5) \Rightarrow$ primal objective value $= 34$

7. Minimize $w = \mathbf{Yb}$ subject to $\mathbf{YA} = \mathbf{C}$, $\mathbf{Y}$ unrestricted

Set 7.6a

1. $-\frac{2}{7} \leq t \leq 1$

2. (a)

Basic solution	Applicable range of t
$(x_2, x_3, x_6) = (5, 30, 10)$	$0 \leq t \leq \frac{1}{3}$
$(x_2, x_3, x_1) = (\frac{25}{4}, \frac{90}{4}, 5)$	$\frac{1}{3} \leq t \leq \frac{5}{12}$
$(x_2, x_4, x_1) = (\frac{5}{2}, 15, 20)$	$\frac{5}{12} \leq t \leq \infty$

5. $\{z_j - c_j\}_{j=1, 4, 5} = (4 - \frac{3t}{2} - \frac{3t^2}{2}, 1 - t^2, 2 - \frac{t}{2} + \frac{t^2}{2})$. Basis remains optimal for $0 \leq t \leq 1$.

Set 7.6b

1. (a) $t_1 = 10$, $\mathbf{B}_1 = (\mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4)$

2. At $t = 0$, $(x_1, x_2, x_6) = (.4, 1.8, 1)$. It remains basic for $0 \leq t \leq 1.5$. No feasible solution for $t > 1.5$.

CHAPTER 8

Set 8.1a

1. G_5: Minimize s_5^-, $55x_p + 3.5x_f + 5.5x_s - .0675x_g + s_5^+ - s_5^- = 0$

3. Let x_1 = No. of in-state freshmen, x_2 = No. of out-of-state freshmen, x_3 = No. of international freshmen

G_i: Minimize s_i^+, $i = 1, 2, \ldots, 5$, subject to $x_1 + x_2 + x_3 + s_1^+ - s_1^- = 1200$,

$$2x_1 + x_2 - 2x_3 + s_2^+ - s_2^- = 0,\ -.1x_1 - .1x_2 + .9x_3 + s_3^+ - s_3^- = 0,$$

$$.125x_1 - .05x_2 - .556x_3 + s_4^+ - s_4^- = 0,\ -.2x_1 + .8x_2 - .2x_3 + s_5^+ - s_5^- = 0$$

All variables are nonnegative

5. Let x_j = No. of production runs in shift j, $j = 1, 2, 3$.

Minimize $z = s_1^+ + s_1^-$, subject to $-100x_1 + 40x_2 - 80x_3 + s_1^+ - s_1^- = 0$,

$$4 \le x_1 \le 5,\ 10 \le x_2 \le 20,\ 3 \le x_3 \le 5$$

Set 8.2a

1. Objective function: Minimize $z = s_1^+ + s_2^+ + s_3^+ + s_4^- + s_5^-$
Solution: $x_p = .0201$, $x_f = .0457$, $x_s = .0582$, $x_g = 2$ cents, $s_5^- = 1.45$
Gasoline tax is \$1.45 million short of goal.

4. x_1 = lb of limestone/day, x_2 = lb of corn/day, x_3 = lb of soybean meal/day.
Objective function: Minimize $z = s_1^+ + s_2^- + s_3^+ + s_4^+ + s_5^-$
Solution: $x_1 = 166.08$ lb, $x_2 = 2778.56$ lb, $x_3 = 3055.36$ lb, $z = 0$.
All goals are satisfied.

7. x_j = No. of units of product j, $j = 1, 2$.
Assign a relatively high weight to the quota constraints.
Objective function: Minimize $z = 100s_1^+ + 100s_2^+ + s_3^- + s_4^-$
Solution: $x_1 = 80$, $x_2 = 60$, $s_3^- = 100$ minutes, $s_4^- = 120$ minutes.
Production quota can be met with 100 minutes of overtime for machine 1 and 120 minutes of overtime for machine 2.

Set 8.2b

2. G_1 solution: $x_p = .0204$, $x_f = .0457$, $x_s = .0582$, $x_g = 2$. $s_5^- = 1.45$, all others = 0.
Goals G_1, G_2, G_3, and G_4 are satisfied. G_5 is not.
G_5 problem: Same as G_1 plus $s_1^+ = 0$, $s_2^+ = 0$, $s_3^+ = 0$, $s_4^- = 0$.
Solution: Same as G_1 with $s_5^- = 1.45$. This means that G_5 cannot be satisfied.

CHAPTER 9

Set 9.1a

3. x_{ij} = No. of bottles of type i assigned to individual j, where i = 1(full), 2(half full), 3(empty).

Constraints:

$$x_{11} + x_{12} + x_{13} = 7,\ x_{21} + x_{22} + x_{23} = 7,\ x_{31} + x_{32} + x_{33} = 7$$

$$x_{11} + .5x_{21} = 3.5,\ x_{12} + .5x_{22} = 3.5,\ x_{13} + .5x_{23} = 3.5$$

$$x_{11} + x_{21} + x_{31} = 7,\ x_{12} + x_{22} + x_{32} = 7,\ x_{13} + x_{23} + x_{33} = 7$$

$$x_{ij} \geq 0 \text{ and integer for all } i \text{ and } j$$

Solution: Use a dummy objective function.

	No. bottles assigned to individual		
Status	*1*	*2*	*3*
Full	1	3	3
Half full	5	1	1
Empty	1	3	3

6. y = Original sum of money. x_j = Amount taken on night j, j = 1, 2, 3.
x_4 = Amount given to each mariner by first officer.
Minimize $z = y$ subject to $3x_1 - y = 2$, $x_1 + 3x_2 - y = 2$, $x_1 + x_2 + 3x_3 - y = 2$, $y - x_1 - x_2 - x_3 - 3x_4 = 1$. All variables are nonnegative integers.
Solution: $y = 79 + 81n$, $n = 0, 1, 2, \ldots$

8. Side 1: 5, 6, and 8 (27 minutes). Side 2: 1, 2, 3, 4, and 7 (28 minutes). Problem has alternative optima.

Set 9.1b

1. Let x_j = No. widgets produced on machine j, j = 1, 2, 3. y_j = 1 if machine j is used and 0 otherwise. Minimize $z = 2x_1 + 10x_2 + 5x_3 + 300y_1 + 100y_2 + 200y_3$ subject to

$$x_1 + x_2 + x_3 \geq 2000,\ x_1 - 600y_1 \leq 0,\ x_2 - 800y_2 \leq 0,\ x_3 - 1200y_3 \leq 0,$$

$$x_1, x_2, x_3 \geq 500 \text{ and integer, } y_1, y_2, y_3 = (0, 1).$$

Solution: $x_1 = 600$, $x_2 = 500$, $x_3 = 900$, z = \$11,300.

2. Solution: Site 1 is assigned to targets 1 and 2, and site 2 is assigned to targets 3 and 4. $z = 18$.

Set 9.1c

1. Let $x_j = 1$ if route j is selected and 0 otherwise, $j = 1, 2, \ldots, 6$.
Minimize $z = 80x_1 + 50x_2 + 70x_3 + 52x_4 + 60x_5 + 44x_6$ subject to

$$x_1 + x_3 + x_5 + x_6 \geq 1,\ x_1 + x_3 + x_4 + x_5 \geq 1,\ x_1 + x_2 + x_4 + x_6 \geq 1,$$

$$x_1 + x_2 + x_5 \geq 1,\ x_2 + x_3 + x_4 + x_6 \geq 1,\ x_j = (0, 1),\ \text{for all } j.$$

Solution: $x_5 = x_6 = 1$. Select routes (1, 4, 2) and (1, 3, 5), $z = 104$. Customer 1 should be skipped in one of the two routes.

2. Solution: Committee is formed of individuals *a*, *d*, and *f*. Problem has alternative optima.

Set 9.1d

1. (a) Let x_{ij} = Integer amount assigned to square (i, j).
Constraints:

$$\sum_{j=1}^{3} x_{ij} = 15,\ i = 1, 2, 3,\ \sum_{i=1}^{3} x_{ij} = 15,\ j = 1, 2, 3$$

$$x_{11} + x_{22} + x_{33} = 15,\ x_{31} + x_{22} + x_{13} = 15,$$

$$(x_{11} \geq x_{12} + 1 \text{ or } x_{11} \leq x_{12} - 1),\ (x_{11} \geq x_{13} + 1 \text{ or } x_{11} \leq x_{13} - 1),$$

$$(x_{12} \geq x_{13} + 1 \text{ or } x_{12} \leq x_{13} - 1),\ (x_{11} \geq x_{21} + 1 \text{ or } x_{11} \leq x_{21} - 1),$$

$$(x_{11} \geq x_{31} + 1 \text{ or } x_{11} \leq x_{31} - 1),\ (x_{21} \geq x_{31} + 1 \text{ or } x_{21} \leq x_{31} - 1),$$

$$x_{ij} = 1, 2, \ldots, 9 \text{ for all } i \text{ and } j$$

Solution: Problem has more than two alternative solutions.

9	1	5
1	5	9
5	9	1

or

6	4	5
4	5	6
5	6	4

3. Solution: Produce 26 units of product 1, 3 of product 2, and none of product 3, and use location 2.

Set 9.2a[1]

2. (a) $z = 23,\ x_1 = 3,\ x_2 = 2.$
(e) $z = 37,\ x_1 = 6,\ x_2 = 1.$

3. (a) $z = 7.25,\ x_1 = 1.75,\ x_2 = 1.$
(e) $z = 37,\ (x_1 = 4.6,\ x_2 = 2)$ or $(x_1 = 6,\ x_2 = 1)$

Set 9.2b

1. (a) 9 subproblems.
(b) 25,739 subproblems.

[1] Use TORA integer programming module to generate the B&B tree.

3. Equivalent 0-1 ILP:

$$\text{Maximize } z = 18y_{11} + 36y_{12} + 14y_{21} + 28y_{22} + 8y_{31} + 16y_{32} + 32y_{33}$$

$$\text{subject to } 15y_{11} + 30y_{12} + 12y_{21} + 24y_{22} + 7y_{31} + 14y_{32} + 28y_{33} \leq 43$$

All variables are binary.

Solution: $z = 50$, $y_{12} = 1$, $y_{21} = 1$, all others $= 0$. Equivalently, $x_1 = 2$, $x_2 = 1$.

Set 9.2c

1. (a) Legitimate cut because it passes through an integer point and does not eliminate any feasible integer point. You can verify this result by plotting the cut on the LP solution space.

6. (b) Optimum integer solution: $(x_1, x_2, x_3) = (5, 2, 2)$, $z = 23$.
Rounded solution: $(x_1, x_2, x_3) = (5, 3, 3)$, which is infeasible.

Set 9.3a

1. The table below gives the number of distinct employees who enter/leave the manager's office when we switch from project i to project j. The objective is to find a "tour" through all projects that will minimize the total traffic.

				Project j			
		1	2	3	4	5	6
	1	–	4	4	6	6	5
	2	4	–	6	4	6	3
	3	4	6	–	4	8	7
Project i	4	6	4	4	–	6	5
	5	6	6	8	6	–	5
	6	5	3	7	5	5	–

CHAPTER 10

Set 10.1a

1. Solution: Shortest distance = 21 miles. Route: 1-3-5-7.

Set 10.2a

3. Solution: Shortest distance = 17. Route: 1-2-3-5-7.

Set 10.3a

2. (a) Solution: Value = 120. $(m_1, m_2, m_3) = (0, 0, 3)$ or $(0, 2, 2)$ or $(0, 4, 1)$ or $(0, 6, 0)$.

5. Solution: Total points = 250. Select 2 courses from I, 3 from II, 4 from III, and 1 from IV.

7. Let $x_j = 1$ if application j is accepted, and 0 otherwise. Equivalent knapsack model is

$$\text{Maximize } z = 78x_1 + 64x_2 + 68x_3 + 62x_4 + 85x_5 \text{ subject to}$$

$$7x_1 + 4x_2 + 6x_3 + 5x_4 + 8x_5 \leq 23,\ x_j = (0, 1),\ j = 1, 2, \ldots, 5.$$

Solution: Accept all but the first application. Value = 279.

Set 10.3b

1. (a) Solution: Hire 6 in week 1, fire 1 in week 2, fire 2 in week 3, hire 3 in week 4, and hire 2 for week 5.

3. Solution: Rent 7 cars in week 1, return 3 in week 2, rent 4 in week 3, and no action for week 4.

Set 10.3c

2. Decisions for next 4 years: Keep, Keep, Replace, Keep. Total cost = $458.

Set 10.3d

3. (a) Let x_i and y_i be no. sheep kept and sold at the end of period i and define $z_i = x_i + y_i$.

$$f_n(z_n) = \max_{y_n = z_n}\{p_n y_n\}$$

$$f_i(z_i) = \max_{y_i \leq z_i}\{p_i y_i + f_{i+1}(2z_i - 2y_i)\},\ i = 1, 2, \ldots, n - 1$$

CHAPTER 11

Set 11.2a

2. (a) Total cost per week = $51.50

(b) Order 239.05 lb whenever inventory level is zero. Total cost per week = $50.20

4. (a) Choose policy 1 because its cost per day is $2.17 as opposed to $2.50 for policy 2.

(b) Optimal policy: Order 100 units whenever the level drops to 10 units. Cost per day = $2.00.

Set 11.2b

2. Optimal policy: Order 500 units whenever level drops to 130 units. Cost per day = $258.50.

3. Optimal policy: Order 150 units whenever level drops to 0. Cost per day = $479.17.

Set 11.2c

1. Excel solution: $(y_1, y_2, y_3, y_4, y_5) = (4.41, 6.87, 4.12, 7.2, 5.8)$.

4. $L(y_1, y_2, y_3, y_4, \lambda) = \sum_{i=1}^{4} \frac{K_i D_i}{y_i} + \frac{h_i y_i}{2} - \lambda\left(\sum_{i=1}^{4} \frac{365D_i}{y_i} - 150\right)$

$$\text{Formula: } y_i = \sqrt{\frac{2(K_i - 365\lambda)D_i}{h_i}}$$

Excel solution: $(y_1, y_2, y_3, y_4, \lambda) = (155.3, 118.82, 74.36, 90.10, -.0564)$.

Set 11.3a

1. (a) 500 units required at the start of periods 1, 4, 7, and 10.

Set 11.3b

3. Produce 173 units in period 1, 180 in period 2, 240 in period 3, 110 in period 4, and 203 in period 5.

Set 11.3c

1. (a) No, because inventory should not be held needlessly at end of horizon.

(b) (i) $0 \le z_1 \le 5$, $1 \le z_2 \le 5$, $0 \le z_3 \le 4$; $x_1 = 4$, $1 \le x_2 \le 6$, $0 \le x_3 \le 4$

2. (a) $z_1 = 7$, $z_2 = 0$, $z_3 = 6$, $z_4 = 0$. Total cost $=$ \$33.

Set 11.3d

1. Use initial inventory to satisfy demand for period 1 and 4 units in period 2, thus reducing demand for the four periods to 0, 22, 90, and 67, respectively.

Optimal solution: Order 112 units in period 2 and 67 units in period 4. Total cost $=$ \$632.

Set 11.3e

1. Solution: Produce 210 units in period 1, 255 in period 4, 210 in period 7, and 165 in period 10. Total cost $=$ \$1930.

CHAPTER 12

Set 12.1a

1. (a) .15 and .25, respectively.

(b) .571

(c) .821.

2. $n \ge 23$.

3. $n > 253$.

Set 12.1b

4. Let p = probability Liz will win. Then, probability John will win is $3p$, which equals the probability Jim will win. Probability Ann will win is $6p$. Thus, $p + 3p + 3p + 3p + 6p = 1$.

Set 12.1c

3. (a) $\frac{3}{8}$.

(b) $\frac{3}{5}$.

7. .9545.

Set 12.2a

2. $K = 20$.

3. $P\{\text{Demand} \geq 1100\} = .3$.

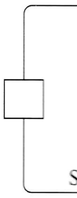
Set 12.3a

3. (a) $P\{x \geq 50\} = \frac{2}{3}$.

(b) Expected no. unsold copies = 2.67.

Set 12.3b

1. Mean = 3.667, variance = 1.556.



Set 12.3c

1. (a) $P\{x_1 = 1, 2, 3\} = P\{x_2 = 1, 2, 3\} = (.4, .2, .4)$.

(b) No.

Set 12.4a

1. $(\frac{1}{2})^{10}$.

3. .0547.

Set 12.4b

1. .8646.

3. (a) $P_0 \cong 0$.

(b) $P_{n\geq 3} \cong 1$.

Set 12.4c

1. $\lambda = 12$ arrivals/min. $P(t \leq 5 \text{ Sec}) = .63$.

Set 12.4d

2. .0014.

CHAPTER 14

Set 14.1a

1. Weights for A, B, and C = (.44214, .25184, .30602).

Set 14.1b

2. $(w_S, w_J, w_M) = (.331, .292, .377)$. Select Maisa.
4. $(w_P, w_H) = (.502, .498)$. Select P.

Set 14.2a

2. (a) See Figure D.15.
(b) EV(corn) = – \$8250, EV(soybean) = \$250. Select soybean.
6. (a) See Figure D.16.
(b) EV(game) = – \$.025. Do not play the game.

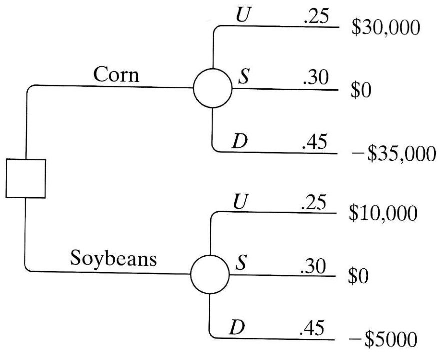


FIGURE D.15

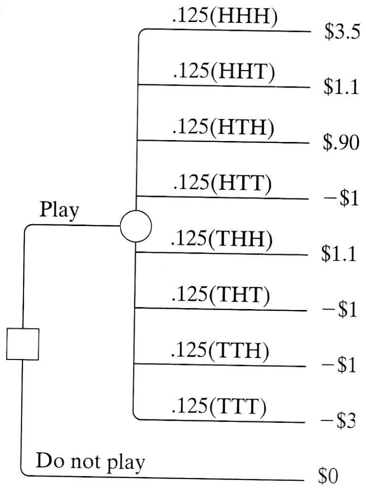


FIGURE D.16

Set 14.2b

2. Let z be the event of having one defective item in a sample of size 5. Answer: $P\{A|z\} = .6097$, $P\{B|z\} = .3903$.
4. (a) Expected revenue if you publish = \$196,000.
Expected revenue if you use publisher = \$163,000.
(b) If survey predicts success, publish it yourself, else use publisher.
7. Ship lot to B if both items are bad; else, ship lot to A.

Set 14.2c

1. (a) Expected value = \$5, hence there is no advantage.
(b) For $0 \le x < 10$, $U(x) = 0$, and for $x = 10$, $U(x) = 100$.
(c) Play the game.

2. Lottery: $U(x) = 100 - 100p$.

Set 14.3a

1. (a) All methods: Study all night (action a_1).
(b) All methods: Select actions a_2 or a_3.

Set 14.4a

1. (a) Saddle-point solution at (2,3). Value of game = 4.

3. (a) $2 < v < 4$.

Set 14.4b

1. Each player should mix strategies 50-50. Value of game = 0.

2. Robin's payoff matrix:

	100%*A*	50%*A*-50%*B*	100%*B*
A	−100	−50	0
B	0	−30	−100

Strategy for Police: Mix 100%*A* and 100%*B* with probability .5 each. Strategy for Robin: Mix A and B 50-50. Value of game = \$50 (expected fine paid by Robin).

Set 14.4c

1. (a) Payoff matrix for the searching team:

	AB	AC	AD	BC	BD	CD
AB	1	0	0	0	0	−1
AC	0	1	0	0	−1	0
AD	0	0	1	−1	0	0
BC	0	0	−1	1	0	0
BD	0	−1	0	0	1	0
CD	−1	0	0	0	0	1

Optimal strategy for both teams: Mix AB and CD 50-50. Value of the game = 0.

3. (a) Payoff matrix for Colonel Blotto:

	3-0	2-1	1-2	0-3
2-0	−1	−1	0	0
1-1	0	−1	−1	0
0-2	0	0	−1	−1

Optimal strategy for Blotto: Blotto mixes (2-0) and (0-2) 50-50, and the enemy mixes (2-1) and (0-3) 50-50. Value of the game = −.5, and Blotto loses.

CHAPTER 15

Set 15.1a

2. Solution: Day 1: Accept if offer is high. Day 2: Accept if offer is medium or high. Day 3: Accept any offer.

Set 15.2a

1. Solution: Year 1: Invest $10,000. Year 2: Invest all. Year 3: Do not invest. Year 4: Invest all. Expected accumulation = $35,520.

4. Allocate 2 bikes to center 1, 3 to center 2, and 3 to center 3.

Set 15.3a

3. Solution: First game: Bet $1. Second game: Bet $1. Third game: Bet $1 or none. Maximum probability = .109375.

CHAPTER 16

Set 16.1a

1. (a) Order 1000 units whenever inventory level drops to 537 units.

Set 16.1b

2. Solution: $y^* = 317.82$ gallons, $R^* = 46.82$ gallons.

3. Solution: $y^* = 316.85$ gallons, $R^* = 58.73$ gallons. In Example 16.1-2, $y^* = 319.44$ gallons, $R^* = 93.61$ gallons. Order quantity remains about the same as in Example 16.1-2, but R^* is smaller because the demand pdf has a smaller variance.

Set 16.2a

3. $19 \leq p \leq 35.7$

6. 39 coats.

Set 16.2b

1. Order $8 - x$ if $x < 3.528$, else do not order.

Set 16.3a

2. Order $4.61 - x$ if $x < 4.61$, else do not order.

CHAPTER 17

Set 17.1a

1. (a) Efficiency = 71%.
(b) The two requirements cannot be satisfied simultaneously.

Set 17.2a

1.

Situation	Customer	Server
a	Plane	Runway
b	Passenger	Taxi

Set 17.3a

1. (b) (i) $\lambda = 6$ arrivals per hour, average interarrival time $= \frac{1}{6}$ hour.
(c) (i) $\mu = 5$ services per hour, average service time $= .2$ hour.

3. (a) $f(t) = 20e^{-20t}$, $t > 0$.
(b) $P\{t > \frac{15}{60}\} = .00674$.

7. Jim's payoff is 2 cents with probability $P\{t \leq 1\} = .4866$ and -2 cents with probability $P\{t \geq 1\} = .5134$. In 8 hours, Jim pays Ann = 17.15 cents.

10. (a) $P\{t \leq 4 \text{ minutes}\} = .4866$.
(b) Expected discount percentage = 6.208.

Set 17.4a

1. $P_{n \geq 5}(1 \text{ hour}) = .55951$.

4. (a) $P_2(t = 7) = .24167$.

6. (a) Combined $\lambda = \frac{1}{10} + \frac{1}{7}$, $P_2(t = 5) = .219$.

Set 17.4b

2. (a) $\mu t = 9$, $p_0(t = 3) = .00532$.
(c) $\mu t = 3$, $p_{n \leq 17}(t = 1) = .9502$.

5. $\mu t = 4$, $p_0(4) = .37116$.

8. (a) Average order size $= 25 - 7.11 = 17.89$ items.

(b) $\mu t = 12$, $p_0(t = 4) = .00069$.

Set 17.5a

3. (a) $p_{n \geq 3} = .4445$.

(b) $p_{n \leq 2} = .5555$.

6. (a) $p_j = .2$, $j = 0, 1, 2, 3, 4$.

(b) Expected number in shop $= 2$ customers.

(c) $p_4 = .2$.

Set 17.6a

1. (a) $L_q = 1p_6 + 2p_7 + 3p_8 = .1917$ car.

(c) $\lambda_{\text{lost}} = .1263$ car per hour.

(d) No. of empty spaces $= c - (L_s - L_q) = c - \sum_{n=0}^{8} np_n + \sum_{n=c+1}^{8} (n - c)p_n$.

Set 17.6b

2. (a) $p_0 = .2$.

(b) Average monthly income $= \$50 \times \mu t = \$375$.

(c) Expected payment $= \$40 \times L_q = \$128$.

5. (a) $p_0 = .4$.

(b) $L_q = .9$ car.

(d) $p_{n \geq 11} = .0036$.

6. (d) No. of spaces is at least 13.

Set 17.6c

1. $P\{\tau > 1\} = .659$.

5. \$37.95 per 12-hour day.

Set 17.6d

1. (a) $p_0 = .3654$.

(b) $W_q = .207$ hour.

(c) Expected number of empty spaces $= 4 - L_q = 3.212$.

(e) $\mu = 10$ will reduce W_s to about 9.6 minutes.

4. (a) $p_8 = .6$.

(c) Probability of finding an empty space cannot exceed .4 regardless of belt capacity. This means that the best utilization of the assembly department is 60%.

7. (a) $1 - p_5 = .962$.
(b) $\lambda_{\text{lost}} = \lambda p_5 = .19$ customer per hour.

Set 17.6e

2. For $c = 2$, $W_q = 3.446$ hour and for $c = 4$, $W_q = 1.681$ hour, an improvement of over 50%.

5. Let K be the number of waiting room spaces. Using TORA,
$p_0 + p_1 + \cdots + p_{K+2} \geq .999$ yields $K \geq 10$.

7. (a) $p_{n\geq 4} = .65772$.
(e) Average number of idle computers = .667 computer.

Set 17.6f

2. (c) Utilization = 81.8%.
(d) $p_2 + p_3 + p_4 = .545$.

4. (a) $p_{40} = .00014$.
(b) $p_{30} + p_{31} + \cdots + p_{39} = .02453$.
(d) Expected number of occupied spaces $= L_s - L_q \cong 20$.
(f) Probability of not finding a parking space $= 1 - p_{n\leq 29} = .02467$. Number of students who cannot park in an 8-hour period is approximately 4.

Set 17.6g

2. (a) Approximately 7 seats.
(b) $p_{n\geq 8} = .2911$.

Set 17.6h

1. (b) Average number of idle repair persons $= 4 - (L_s - L_q) = 2.01$.
(d) $P\{2 \text{ or } 3 \text{ idle servers}\} = p_0 + p_1 = .34492$.

4. (a) $L_s = 1.25$ machines.
(b) $p_0 = .33341$.
(c) $W_s = .25$ hour.

6. $\lambda = 2$ calls per hour per baby, $\mu = .5$ baby per hour, $R = 5$, $K = 5$.
(a) Number of awake babies $= 5 - L_s = 1$ baby.
(b) $P_5 = .32768$.
(c) $p_{n\leq 2} = .05782$.

Set 17.7a

2. (a) $E\{t\} = 14$ minutes and $\text{var}\{t\} = 12$ minutes^2. $L_s = 7.867$ cars.

4. $\lambda = .0625$ prescriptions per minute, $E\{t\} = 15$ minutes, $\text{var}\{t\} = 9.33$ minutes^2.
(a) $p_0 = .0625$.

(b) $L_q = 7.3$ prescriptions.
(c) $W_s = 132.17$ minutes.

Set 17.9a

2. Use (M/M/1):(GD/10/10). Cost per hour is \$431.50 for repair person 1 and \$386.50 for repair person 2.

4. (b) $\mu = \lambda + \sqrt{\frac{c_2 \lambda}{c_1}}$

(c) Optimum production rate = 2725 pieces per hour.

Set 17.9b

2. (a) Hourly cost per hour is \$86.4 for two repair persons and \$94.80 for three.

(b) Schedule loss per breakdown = \$30 × W_s = \$121.11 for two repair persons and \$94.65 for three.

4. $\lambda = .36125$ per machine per hour, $\mu = 10$ per hour. Model $(M/M/3){:}(GD/20/20)$ yields $W_s = .10118$ hour. Lost revenue per machine per hour $= 25 \times \lambda W_s \times \$2 = \$1.83$ or \$36.55 for all 20 machines. Cost of 3 repair persons is \$60.

Set 17.9c

1. (a) Number of repair persons = 5.

(b) Number of repair persons = 4.

CHAPTER 18

Set 18.1a

4. (a) $P\{H\} = P\{T\} = .5$. If $0 \le R \le .5$, Jim gets \$10.00. If $.5 < R \le 1$, Jan gets \$10.00.

7. Lead time sampling: If $0 \le R \le .5$, $L = 1$ day. If $.5 < R \le 1$, $L = 2$ days. Demand per day sampling: If $0 \le R \le .2$, demand = 0 unit. If $.2 < R \le .9$, demand = 1 unit. If $.9 < R \le 1$, demand = 2 units. Use one R to sample L. If $L = 1$, use another R to sample demand for one day, else if $L = 2$, use one R to generate demand for day 1 and then another R to generate demand for day 2.

Set 18.2a

1. (a) Discrete.

Set 18.3a

4. See Figure D.17.

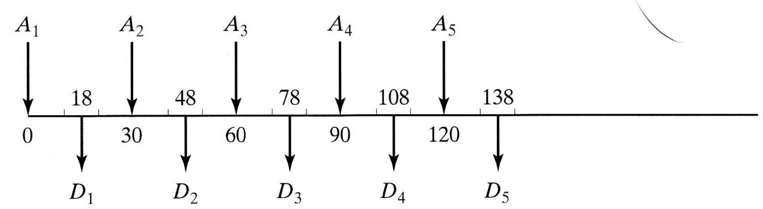


FIGURE D.17

Set 18.3b

1. $t = -\frac{1}{\lambda}\ln(1 - R)$, $\lambda = 4$ customers per hour.

Customer	R	t	Arrival time
1	—	—	0
2	.0589	.015	.015
3	.6733	.280	.295
4	.4799	.163	.458

2. $t = a + (b - a)R$.

4. (a) $0 \le R < .2$: $d = 0$, $.2 \le R < .5$: $d = 1$, $.5 \le R < .9$: $d = 2$, $.9 \le R \le 1$: $d = 3$.

9. If $0 \le R \le p$, then $x = 0$, else $x =$ (largest integer $\le \frac{\ln(1 - R)}{\ln q}$).

Set 18.3c

1. $y = -\frac{1}{5}\ln(.0589 \times .6733 \times .4799 \times .9486) = .803$ hour.

6. $t = x_1 + x_2 + x_3 + x_4$, where $x_i = 10 + 10R_i$, $i = 1, 2, 3, 4$.

Set 18.5a

2. (a) Observation-based.

(b) Time-based.

4. (a) 1.1 barbers.

Set 18.6a

2. Confidence interval: $15.07 \le \mu \le 23.27$.

CHAPTER 19

Set 19.1a

2. Do not fertilize, fertilize when in state 1, fertilize when in state 2, fertilize when in state 3, fertilize when in state 1 or 2, fertilize when in state 1 or 3, fertilize when in state 2 or 3, or fertilize regardless of state.

Set 19.2a

1. Years 1 and 2: Don't advertise if product is successful; otherwise, advertise. Year 3: Don't advertise.
3. If stock level at the start of month is zero, order 2 refrigerators; otherwise, do not order.

Set 19.3a

1. Advertise whenever in state 1.

CHAPTER 20

Set 20.1a

1. (a) No stationary points.
   (b) Minimum at $x = 0$.
   (e) Inflection point at $x = 0$, minimum at $x = .63$, and maximum at $x = -.63$.
4. $(x_1, x_2) = (-1, 1)$ or $(2, 4)$.

Set 20.2a

1. (b) $(\partial x_1, \partial x_2) = (2.83, -2.5)\partial x_2$

Set 20.2b

3. Necessary conditions: $2(x_i - \frac{x_n^2}{x_i}) = 0$, $i = 1, 2, \ldots, n - 1$. Solution is $x_i = \sqrt[n]{C}$, $i = 1, 2, \ldots, n$. $\partial f = 2\delta C^{\frac{2-n}{n}}$.
6. (b) Solution: $z = \frac{375}{74}$, $(x_1, x_2, x_3 x_4) = (-\frac{5}{74}, -\frac{10}{74}, \frac{155}{74}, \frac{60}{74})$, which is a minimum point.

Set 20.2c

2. Minima points: $(x_1, x_2, x_3, \lambda_1, \lambda_2) = (-14.4, 4.56, -1.44, 38.5, -67.3)$ and $(4.4, .44, .44, 10.2, -1.4)$.

CHAPTER 21

Set 21.1a

2. (c) $x = 2.5$.
   (e) $x = 2$.
3. Number of iterations $\approx 1.44 \ln(\frac{b-a}{\Delta - 1})$.

Set 21.1b

1. By Taylor's expansion, $\nabla f(\mathbf{X}) = \nabla f(\mathbf{X}^0) + \mathbf{H}(\mathbf{X} - \mathbf{X}^0)$. The Hessian $\mathbf{H}$ is independent of $\mathbf{X}$ because $f(\mathbf{X})$ is quadratic. Also, the given expansion is exact because higher-order derivatives are zero. Thus, $\nabla f(\mathbf{X}) = \mathbf{0}$ yields $\mathbf{X} = \mathbf{X}^0 - \mathbf{H}^{-1}\nabla f(\mathbf{X}^0)$. Because $\mathbf{X}$ satisfies $\nabla f(\mathbf{X}) = \mathbf{0}$, $\mathbf{X}$ must be optimum regardless of the choice of initial $\mathbf{X}^0$.

Set 21.2a

2. Optimal solution: $x_1 = 0,\ x_2 = 3,\ z = 16$.
4. Let $w_j = x_j + 1,\ j = 1, 2, 3,\ v_1 = w_1 w_2,\ v_2 = w_1 w_3$. Then,

   Maximize $z = v_1 + v_2 - 2w_1 - w_2 + 1$ subject to

$$v_1 + v_2 - 2w_1 - w_2 \le 9,\ \ln v_1 - \ln w_1 - \ln w_2 = 0$$

$$\ln v_2 - \ln w_1 - \ln w_3 = 0, \quad \text{all variables are nonnegative.}$$

Set 21.2b

1. Solution: $x_1 = 1,\ x_2 = 0,\ z = 4$.
2. Solution: $x_1 = 0,\ x_2 = .4,\ x_3 = .7$.

Set 21.2c

3. Solution: $x_1 = 1.39,\ x_2 = 1.13$

Set 21.2d

2. Maximize $z = x_1 + x_2^2 + x_3$ subject to

$$x_1^2 + 5x_2^2 + 2\sqrt{x_3} + 1.28y \le 10$$

$$16x_2^2 + 25x_3 - y^2 = 0, \quad \text{all variables are nonnegative}$$

Index

OPERATIONS RESEARCH

AN INTRODUCTION

SEVENTH EDITION

HAMDY A. TAHA

This seventh edition continues to build on the strength of the first six editions, providing balanced coverage of the theory, applications, and computations of operations research. Complex mathematical concepts are effectively explained by means of carefully designed numerical examples, essentially eliminating the need for the usually obscure formal mathematical proofs. The book includes fully analyzed practical situations and each chapter concludes with summary applications borrowed from published case studies. The role of modern computational tools in enhancing the effectiveness of operations research as a decision-making tool receives considerable attention in this new edition.

New for the Seventh Edition:

- Practically every algorithm in the book is now supported and explained by an appropriate software tool, greatly facilitating the process of explaining concepts that otherwise would be difficult, if not impossible, to demonstrate.
- The powerful Windows®-based TORA software offers new and unique *tutorial* features, ranging from animated graphical LP solution to dynamic construction of CPM time charts and generation of branch-and-bound search trees.
- For the first time, Excel® templates are designed to solve *general* problems in dynamic programming, inventory problems, and the analytic hierarchy approach, simply by changing the input data in the template.
- Excel® Solver is used to solve transportation, network, and linear and nonlinear programming problems.
- Commercial AMPL® and LINGO® packages are used to show how very large linear and integer programs are solved in practice.
- Chapters 1 through 14, 17, and 18 have been streamlined to eliminate ambiguity and redundancy in the presentation of the material.
- New material includes a new introduction to operations research, the generalized simplex method, representation of all network models, including CPM, as linear programs, PERT networks, solution of the traveling salesperson problem, and the golden section method.
- Numerous new problems have been added in many chapters.
- Appendix D provides complete solutions to selected chapter problems.

Software Support:

- Easy-to-use menu-driven TORA optimization system.
- Over 20 general and ready-to-use Excel® spreadsheet templates.
- Several Excel® Solver templates.
- Example applications of the commercial packages AMPL® and LINGO®.

ISBN 0-13-032374-8

90000

9 780130 323743

Pearson Education

Prentice Hall
Upper Saddle River, NJ 07458
www.prenhall.com