# SECTION V

# METHODS FOR DECISION MAKING

**A. Probabilistic Models and Statistics**
**B. Economic Evaluation**
**C. Computer Simulation**
**D. Optimization**

*Handbook of Industrial Engineering: Technology and Operations Management, Third Edition.*
Edited by Gavriel Salvendy

# V.A

# Probabilistic Models and Statistics

*Handbook of Industrial Engineering: Technology and Operations Management, Third Edition.*
Edited by Gavriel Salvendy Copyright © 2001 John Wiley & Sons, Inc.

# CHAPTER 83
# Stochastic Models

COLM A. O'CINNEIDE
Purdue University

*Handbook of Industrial Engineering: Technology and Operations Management, Third Edition.*
Edited by Gavriel Salvendy

# 1. INTRODUCTION

## 1.1. Overview

Many systems of interest to industrial engineers involve randomness or unpredictability; for example, in the arrival of jobs requiring processing, or in machine breakdowns. In attempting to understand these systems for the purpose of design or control, a mathematical model is needed. Often, randomness and uncertainty must be captured explicitly in the model in order to represent the system reasonably faithfully. Such a model is called a *stochastic model*. The value of these models is that they enable us to predict the performance of a new system or the effect of a change in an existing system.

This chapter is a tutorial covering those aspects of the theory of stochastic models that are of greatest importance for applications in the manufacturing and service industries. After some preliminaries in this section, we cover four families of stochastic models. *Point processes,* treated in Section 2, are collections of random times, representing, for example, arrival times of jobs at a service facility. *Markov chains* are treated in Section 3. These offer both mathematical tractability and the flexibility to model a broad range of systems. *Queueing systems,* treated in Sections 4 and 5, capture the phenomenon of waiting. These are the central focus of this chapter. Indeed, the primary reason for treating point processes and Markov chains is to provide a foundation from which to build and analyze useful queueing models. Queueing networks, treated in Sections 6 and 7, are models of two or more interacting queues.

This chapter deals only with the simplest and most basic stochastic models and their mathematical analysis. Among the many excellent sources for the introductory material discussed here are the books by Ross (1997) and Wolff (1989), who treat the subject fairly mathematically, and Hall (1991), who focuses more on practical considerations. Ross (1996) is more advanced. For a comprehensive treatment of stochastic models of manufacturing systems, see Buzacott and Shanthikumar (1993). Larson and Odoni (1981) and Hall (1991) cover a variety of examples of stochastic modeling for service industries. Other chapters of the Handbook with a substantial stochastic modeling component are Chapters 60, 72, and 94.

## 1.2. Simulation vs. Mathematical Analysis

One of the primary applications of stochastic models is in discrete-event simulation of engineering systems that are subject to randomness. The first step in this methodology is to develop a stochastic model of the system in question. Simulation is used to analyze the model because models of real systems are usually too complex for direct mathematical analysis. Simulation is treated thoroughly in Chapters 93–96 of the Handbook.

Only relatively simple stochastic models may be examined in any detail using mathematical analysis, whereas almost any stochastic model may be analyzed using simulation. Because of this, it is necessary to state clearly what benefits are derived from mathematical analysis of stochastic models. These are as follows.

1. The mathematics yields insights and general principles, such as Little's law (see Section 5.1), which cannot be readily discovered from simulations.
2. The mathematics can be used to validate simulation results. This may be especially important in designing a new system, when one cannot validate the simulation by comparison with observations on an existing system. The goal of validating simulations motivates much of the interest in stochastic models; see Subsection 5.4 and Section 7 below.
3. The mathematics can sometimes show explicitly how performance is affected by system parameters and configuration choices. This is true for certain special families of models, including many of those treated in Sections 4 to 6 below. To get similar information through simulation may require a large amount of computation.
4. The mathematics may help us to understand a system that is fundamentally hard to simulate. Examples of this include models involving queues in heavy traffic (Subsection 5.4) or problems in which a very small probability must be estimated. Long simulation runs may be needed to get any accuracy in such situations, whereas various limit theorems in probability theory often give very accurate results.

## 1.3. Preliminaries on Probability

Here we briefly review some of the fundamental concepts in probability used in later sections. To define probabilities, we begin with an *experiment,* which may be thought of as some process leading to one of a collection of *outcomes*. This collection of outcomes is called the *sample space*. An *event* is a subcollection of outcomes. We say that an event *occurs* if the outcome that results from the experiment is in the event. Events are precisely the objects to which probabilities are assigned. The probability of an event $E$ is denoted by $P(E)$.

The *conditional probability* of an event $A$ given an event $B$ is the probability that they both occur divided by the probability that $B$ occurs:

$$P(A|B) = \frac{P(A \text{ and } B)}{P(B)} \tag{1}$$

This is also called simply the probability of $A$ given $B$. One of the most important concepts in probability is the formal notion of independence or ‘‘unrelatedness’’—that one event has no bearing on another. Formally, two events $A$ and $B$ are said to be *independent* if either of the following holds.

$$P(A|B) = P(A) \text{ or } P(A \text{ and } B) = P(A)P(B) \tag{2}$$

The first of these may be expressed by saying that the knowledge that $B$ will occur has no effect on the chance that $A$ will occur.

A *random variable* $X$ is a quantity whose value depends on the outcome of an experiment. The *distribution* of a random variable is any specification of all probabilities associated with that random variable. One such specification is the *distribution function* $F$ of $X$, which is defined by

$$F(x) = P(X \leq x), -\infty < x < \infty \tag{3}$$

If $X$ is *discrete*, which is to say its possible values may be written as a list $x_1, x_2, x_3, \ldots$, we may specify its distribution through its *probability mass function,* $f$, defined by

$$f(x) = P(X = x), -\infty < x < \infty \tag{4}$$

Note that $f(x) = 0$ unless $x$ is one of the possible values $x_i$ of $X$. To say that $X$ is a *continuous random variable* is to say that there is a function $f$ for which

$$P(X \leq x) = \int_{-\infty}^{x} f(u)\, du, -\infty < x < \infty \tag{5}$$

In this situation, $f$ is called the *probability density function* of $X$, and $f$ is the derivative $F'$ of the distribution function $F$ of $X$.

The *expected value* $E(X)$ of a random variable $X$ is its weighted average value, in which each possible value of $X$ is weighted by its probability. This has the interpretation of being a central value, sometimes also a typical value, of $X$. It may be written in the following ways, depending on whether the random variable is discrete or continuous.

$$E(X) = \sum_x xf(x) \text{ in the discrete case, and}$$

$$E(X) = \int_{-\infty}^{\infty} xf(x)\, dx \text{ in the continuous case} \tag{6}$$

Sometimes it is convenient to use the Steiltjes integral notation to combine these into the single formula

$$E(X) = \int_{-\infty}^{\infty} x\, dF(x) \tag{7}$$

The *variance* of a random variable is its *expected squared deviation* from its mean, or

$$\text{Var}(X) = E[(X - E(X))^2] = \int_{-\infty}^{\infty} (x - E(X))^2 f(x)\, dx \tag{8}$$

the integral form being valid in the case of a continuous random variable only. The *standard deviation* SD($X$) is the square root of the variance. This has the interpretation of being a ‘‘typical distance’’ between the value of the random variable and its mean. Because of this simple interpretation, the standard deviation may be considered a natural measure of variability of a random variable. In the context of stochastic models, another measure of variability is often more convenient to work with than the standard deviation. This is the *coefficient of variation,* or CV, defined as the ratio CV($X$) = SD($X$)/$E$($X$) of the standard deviation to the mean. The CV may be interpreted as a typical deviation of a random variable from its mean, expressed relative to that mean. The squared coefficient of variation, or SCV, is often used below also.

We briefly discuss three distributions of importance, the *Poisson distribution,* the *exponential distribution,* and the *normal distribution.* A random variable $X$ has the Poisson distribution with mean $m$ if

$$P(X = x) = e^{-m} \frac{m^x}{x!} \text{ for } x = 0, 1, 2, \ldots \tag{9}$$

This is a discrete distribution, over the non–negative integers. Its expected value and variance are equal:

$$E(X) = \mathrm{Var}(X) = m \tag{10}$$

See Subsection 2.1 for an explanation of how the Poisson distribution arises naturally.

A random variable $Y$ is said to have the exponential distribution with *rate* $\lambda$ if its probability density function is

$$f(y) = \lambda e^{-\lambda y} \text{ for } y \geq 0 \tag{11}$$

The mean and variance are

$$E(Y) = \frac{1}{\lambda} \text{ and } \mathrm{Var}(Y) = \frac{1}{\lambda^2} \tag{12}$$

It follows from this that the CV of the exponential distribution is 1. The exponential distribution has the remarkable *memoryless property,* which states that

$$P(Y > s + t | Y > s) = P(Y > t) \text{ for all } s, t \geq 0 \tag{13}$$

Thinking of $Y$ as a random time, this says that the conditional distribution of the excess of $Y$ over a fixed time $s$, given that $Y$ does indeed exceed $s$, is the same as the unconditional distribution of $Y$. To express this in a more down-to-earth way, suppose that $Y$ is the time at which a train will come and that $Y$ has the exponential distribution. Then, no matter how long we have waited for the train, the distribution of the time we have left to wait is always the same. If this seems counterintuitive, then it may help to consider the memoryless property as it arises in a "discrete-time" setting. If you keep tossing a coin until you first get a head, then as long as you have not yet got a head, it is clear that the distribution of the number of tosses you have left to go is always the same. The number of tails before the first head here has the *geometric distribution*, which is a discrete-time analog of the exponential distribution. See Eq. (24) for its probability mass function.

A random variable $Z$ is said to have the *standard* normal distribution if it has probability density function

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}, \text{ for } -\infty < z < \infty \tag{14}$$

The mean and variance are 0 and 1, respectively. A random variable $X$ is said to have the normal distribution with mean $m$ and variance $\sigma^2$ if $(X - m)/\sigma$ has the standard normal distribution.

To understand how the normal distribution arises, we must introduce another basic concept, that of *independent random variables*. Two random variables, $X$ and $Y$, are said to be independent if

$$P(X \leq x \text{ and } Y \leq y) = P(X \leq x)P(Y \leq y), -\infty \leq x, y \leq \infty \tag{15}$$

This is equivalent to saying that any event defined in terms of $X$ is independent of any event defined in terms of $Y$. For independent random variables $X$ and $Y$, the variance of the sum is the sum of the variances:

$$\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) \tag{16}$$

This simple formula often makes it easy to compute variances. So while it is the standard deviations that are most easily interpreted, typically the variances are the first quantities to be calculated. A significant analogy is that it is the *squared* length of the hypotenuse of a right-angled triangle that is got by adding the *squared* lengths of the other sides—throughout it is squared lengths, not the lengths themselves, that have this simple additive behavior. Often in stochastic models and statistics, collections $X_1$, $X_2$, $X_3$, . . . of independent and identically distributed random variables play a key role. See, for example, Subsection 2.2. For such random variables, with $m = E(X_1)$ and $\sigma^2 = \text{Var}(X_1)$ denoting the common expected value and variance, and with $\overline{X}_n = (X_1 + X_2 + \ldots + X_n)/n$ denoting the mean of the first $n$ of the $X_i$'s, we have

$$E(\overline{X}_n) = m \text{ and } \text{Var}(\overline{X}_n) = \frac{\sigma^2}{n} \tag{17}$$

Thus, $\text{SD}(\overline{X}_n) = \sigma/\sqrt{n}$, and so, while an individual $X_i$ is typically about $\sigma$ in distance from $m$, the mean $\overline{X}_n$ is typically only about $\sigma/\sqrt{n}$ from $m$. This result quantifies the manner in which the individual deviations of the $X_i$'s from their common expected value tend to cancel one another out as they are added together in the process of forming the "sample mean" $\overline{X}_n$. Furthermore, the *central limit theorem* states that the distribution of $\overline{X}_n$ is *asymptotically normal* with mean and variance (17). Roughly speaking, this means that $(\overline{X}_n - m)/(\sigma/\sqrt{n})$ is approximately standard normal. This is one of the most fundamental results in probability and one of the key consequences of independence.

## 2. POINT PROCESSES

In the applications we consider, a point process is a model for a collection of random times. Typical examples are the arrival times of parts at a machine in a manufacturing system or the times at which 911 emergency calls are made in a city. Point processes are also sometimes known by the more informal term *streams*. Mathematically, a point process may be described as a collection of increasing times $0 \leq T_1 < T_2 < T_3 < \ldots$ We often refer to these times generically as arrival times, even though in a given application they may represent departure times or times of some other type of occurrence. Another way to describe a point process is to specify the corresponding *counting process*, which is defined by

$$N(t) = \text{the number of arrivals in the interval } [0, t],\ t \geq 0$$

So $N$ "counts" the number of arrivals up to time $t$. Yet another way to present a point process is through the *interarrival times* $X_1 = T_1$, $X_2 = T_2 - T_1$, $X_3 = T_3 - T_2$, . . . In summary, there are three natural ways to specify a point process: using the arrival times $T_i$ themselves, using the counting process $N$, or using the interarrival times $X_i$. For any particular point process, one or another of these views may give a special insight.

### 2.1. The Poisson Process

The Poisson process is a process of completely random arrivals. It may be described informally as follows. Suppose we break up time into small intervals of length, say, $h$. Suppose that in each time interval we toss a coin with probability of heads $p$, again a small value. (Generally we think of coin tosses as fair in that each outcome has an equal 50% chance. Here we need an unfair coin, but we keep two other properties of coin tossing: that the outcomes of different tosses are independent and that on each toss there is the same probability $p$ of heads.) For example, if we take $h$ to be one second and $p = 1/3600$, then we get one head per hour on average. If a given toss results in heads, we place an arrival at the center of the corresponding interval. Otherwise, there is no arrival in the interval. That this process represents arrivals completely at random is because of two observations: the probability of having an arrival in any one of these one-second intervals is the same, and what happens in one interval is independent of what happens in all the other intervals. On average, this model produces arrivals at rate $p/h$ per unit time, or $p$ arrivals per second. Using the standard symbol $\lambda$ for an arrival rate, we have $\lambda = p/h$. To define the true Poisson process of rate $\lambda$, we must take a limit as $h \to 0$ and $p \to 0$ in such a way that the overall arrival rate $p/h$ is always $\lambda$.

The main properties of the Poisson process are:

1. $N(t)$, the number of arrivals by time $t$, has the Poisson distribution with mean $\lambda t$, which is given by (9) with $m = \lambda t$.
2. The numbers of arrivals in several nonoverlapping time intervals are independent of one another.
3. The interarrival times $X_1, X_2, X_3, \ldots$ are independent with the exponential distribution, given by (11).

If, for a given point process $N$, the expected number of arrivals in any interval is proportional to the length of the interval, then we say that *the arrival rate of N is constant*. This is equivalent to the property that $E(N(t)) = \lambda t$ for all $t \geq 0$. It is a remarkable fact that if a point process has a constant arrival rate and satisfies property 2 above, then it is a Poisson process. The Poisson process arises naturally in many ways. For example, it holds quite generally that if a large number of independent point processes are combined, or superposed, each having a constant arrival rate and each making a small contribution to the total, then the superposition is approximately a Poisson process. The arrival times of telephone calls at a telephone exchange typically behave like a Poisson process, over any period where the overall arrival rate is fairly constant, because they are in fact the superposition of point processes generated by a great many people and computers acting somewhat independently. Similar effects are sometimes seen in manufacturing systems. For example, in a job shop with highly diverse routing (Buzacott and Shanthikumar 1993), arrival streams tend to behave like Poisson processes.

### 2.2. Renewal Processes

If the interarrival times $X_1$, $X_2$, $X_3$, . . . of a point process are independent and identically distributed, then it is called a *renewal process*. To motivate the use of renewal processes in modeling, consider the standard example of replacing light bulbs as they burn out. If the lifetimes $X_i$, $i = 1, 2, \ldots$ of the bulbs are independent and have the same distribution, then the times of replacement form a renewal process. By property 1 of Subsection 2.1, the Poisson process is a renewal process with exponentially distributed interarrival times.

Renewal processes, being a larger class than the Poisson processes, are more difficult to analyze. We do not have a simple description of the distribution of the number of arrivals by time $t$, $N(t)$, for a renewal process. However, we do have approximations for its mean and variance for large $t$. Let $m = E(X_1)$ and $\sigma^2 = \mathrm{Var}(X_1)$ denote the common mean and variance of the $X_i$'s, and let $c^2$ denote the SCV $(\sigma/m)^2$ (see Subsection 1.3). Then the (long-run) *arrival rate* and *asymptotic variance* of the counting process $N$ are defined in general and evaluated in the case of the renewal process in the following:

$$\lambda = \lim_{t\to\infty} \frac{E(N(t))}{t} = \frac{1}{m} \text{ and } \lim_{t\to\infty} \frac{\mathrm{Var}(N(t))}{t} = \lambda c^2 = \frac{\sigma^2}{m^3} \tag{18}$$

The first result here is known as the *elementary renewal theorem*. Simply put, it says that if the average time between arrivals is a half-hour, then on average there will be two arrivals per hour ($\lambda = 2$) in the long run. To enhance the results (2.18), we may invoke an extension of the central limit theorem of Subsection 1.3, stating that $N(t)$ is asymptotically normally distributed with mean $\lambda t$ and variance $\lambda c^2 t$ for large times $t$. This fact is useful in understanding queues in heavy traffic; see Subsection 5.2.

*Example 2.1:* Suppose that 911 calls arrive in a Poisson process of rate one per minute. The probability of more than 80 calls in a 60-minute period may be calculated approximately as follows. Let $Z$ denote a standard normal random variable. By (12) and (18), the mean and variance of $N(60)$ are both approximately equal to 60, and so we have $P(N(60) > 80) \approx P(Z > (80 - 60)/\sqrt{60}) = P(Z > 2.582) = 0.005$. Of course, in this case we are dealing with a special renewal process, namely the Poisson process. Therefore, property 1 of Subsection 2.1 may be used as an alternative approach to computing this probability.

## 3. MARKOV CHAINS

A stochastic process is a collection of random variables $X_t$ defined for a set of times $t$. The counting processes of Section 2 were our first examples of stochastic processes. The term *stochastic model* refers simply to a stochastic process used to model something. There is a large family of stochastic processes, known as the Markov chains, or simply chains, that have enough generality to be genuinely useful in applications and yet enough structure to be mathematically tractable. Markov chains may be thought of as the simplest processes that allow some level of dependence over time. We begin with a simple example to illustrate the Markov property in the context of building a stochastic model.

### 3.1. The Markov Property

Consider a machine that at any given time is either operational or not. We observe it in successive time periods and record a 1 if it is operational and a 0 if it is not. Over time, this produces a sequence of observations, perhaps in this example 1, 1, 1, 0, 0, 1, . . . , indicating that the machine ran for three periods, was then down for two, and so forth. If these observations cannot be predicted with certainty in advance, we may wish to model them as a sequence of random variables $X_1$, $X_2$, . . . , $X_T$, each taking as its value either 0 or 1, with $T$ denoting the number of observation periods. Then the collection of random variables $(X_1, X_2, \ldots, X_T)$ is an example of a stochastic model. It is convenient to denote the entire process simply by $X$. $X_t$ is referred to as the state of the process $X$ at time $t$. The set $S = \{0, 1\}$ of possible values of the $X_t$'s is called the *state space* of the process. The observations 1, 1, 1, 0, 0, 1, . . . constitute a *realization* or *path* of $X$. If we observe the process for $T$ time periods, then each path is a sequence of $T$ zeroes and ones.

A specification of the probabilities associated with a stochastic model is called the *law* or *distribution* of the model. To define the law of $X$, we must assign a probability to each possible path of $X$. Doing this directly is often cumbersome, and the first step in developing stochastic models is to find natural ways of specifying the probabilities of the paths. The simplest specification is to assume that the states at different times are independent and that the probability that the machine is operational at any fixed time $t$ is some constant $p$. Then $1 - p$ is the probability that the machine is inoperational at a time $t$. The probability of any particular path is then $p^r (1 - p)^{T-r}$, where $r$ is the number of time periods for which the machine is operational on this path (and so $T - r$ is the number of time periods for which it is inoperational). This comes from independence of all the $X_t$'s, using an extension of (2) to the effect that the probability of several independent events all occurring is the product of their probabilities.

In the model for machine failures just described, if we have observed the machine up to some time period $t$, these observations are irrelevant from the point of view of predicting the state of the machine in the next time period, period $t + 1$. This is because the model assumes that the successive states of the machine are independent. However, it is quite common in practice that the history of a process is informative regarding its future behavior. In the present example, it may be that when the machine is operational in one time period, it has a better than average chance of being operational in the next time period also. But to allow complete generality in the manner in which future states depend on present and past states would leave us with the problem of overchoice. We will never have enough data to fit such a detailed model. So we make an assumption that allows some dependence, but not too much. We suppose that, having observed the machine for periods 0, 1, 2, . . . , $t$, the distribution of the state of the machine in period $t + 1$ depends only on its state in period $t$, the immediately preceding period. In other words, if we know the condition of the machine at the "present" time $t$, the past conditions are uninformative as to future conditions. This is the *Markov property*. Mathematically, it is expressed by the condition

$$\begin{aligned} &P(X_{t+1} = j | X_t = i, X_{t-1} = i_{t-1}, \ldots, X_1 = i_1) \\ &\quad = P(X_{t+1} = j | X_t = i) \\ &\quad \text{for all states } i_1, i_2, \ldots, i_{t-1}, i, j \in S \text{ and times } t \geq 0 \end{aligned} \tag{19}$$

This conditional probability is called the *transition probability* from state $i$ to state $j$ at time $t$. Once these quantities are specified, the entire law of the Markov process is specified. The Markov process is said to have *stationary transition probabilities* if $P(X_{t+1} = j | X_t = i)$ does not depend on $t$, in which case it is denoted simply by $p_{ij}$. We assume this property from now on, as this is the most important situation in applications, being the situation where the least number of parameters is needed to specify the model.

## 3.2. Transition Matrices

For a Markov chain $X = (X_0, X_1, X_2, \ldots)$ observed at all times $t = 0, 1, 2, \ldots$ and having state space $S = \{1, 2, \ldots, n\}$, the matrix $P$ with entries $p_{ij}$ is called the *transition matrix*. This is an $n \times n$ matrix whose rows add to 1. In the example of the machine above, where the state space $S = \{0, 1\}$ has only two states, the transition matrix is a $2 \times 2$ matrix

$$P = \begin{pmatrix} p_{00} & p_{01} \\ p_{01} & p_{11} \end{pmatrix}$$

Here, for example, $p_{01}$ is the conditional probability that the machine is operational in the next time period, given that it is now inoperational. The other entries have parallel interpretations.

With the introduction of $P$, it is natural to define the vectors

$$\pi(t) = [\pi_1(t), \pi_2(t), \ldots, \pi_n(t)] = [P(X_t = 1), P(X_t = 2), \ldots, P(X_t = n)]$$

which give the distribution of the chain at time $t$. These may all be computed, once the initial distribution $\pi(0)$ of the chain is known, using the following:

$$P(X_{t+1} = j) = \sum_{i=1}^{n} P(X_{t+1} = j | X_t = i) P(X_t = i) = \sum_{i=1}^{n} \pi_i(t) p_{ij} \tag{20}$$

In matrix form, this is the recurrence relation $\pi(t + 1) = \pi(t)P$, repeated application of which gives

$$\pi(t + s) = \pi(t)P^s. \tag{21}$$

It is the compactness of (21) as compared to (20) that motivates the use of vector and matrix notation in dealing with Markov chains. The matrix $P^s$, which is the $s$th power of the matrix $P$, is known as the *s-step transition matrix,* its entry $p_{ij}(s)$ being the probability that the chain, when initialized in state $i$, enters state $j$ upon the $s$th transition.

### 3.3. Irreducibility and Steady State

A *steady-state distribution* of a Markov chain is a distribution $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$, say, such that if $X_0$ has the distribution $\pi$, then $X_1$ also has that distribution, as do $X_2$, $X_3$, and so forth. This is also known as a *stationary distribution.* Using (21), the condition that $\pi$ be a stationary distribution may be expressed in the form

$$\pi P = \pi \tag{22}$$

These are called the *steady-state equations.* A Markov chain is said to be irreducible if any state can be reached from any other through a sequence of transitions whose probabilities are positive. For irreducible chains, there is precisely one steady-state distribution $\pi$, and it is the only vector $x$ that satisfies the steady-state equations $xP = x$ and also satisfies the condition $x_1 + x_2 + \ldots + x_n = 1$.

*Example 3.1:* Figure 1($a$) represents a Markov chain on the state space $S = \{1, 2\}$ with transition matrix

$$P = \begin{pmatrix} 0.5 & 0.5 \\ 1 & 0 \end{pmatrix}$$

This chain is irreducible because each state may be reached from the other. Its steady-state distribution may be computed by solving (22) to give $\pi_1 = 2/3$ and $\pi_2 = 1/3$. Figure 1($b$) represents a Markov chain that is not irreducible because, for example, state 1 cannot be reached from state 3.

Another important concept related to long-run behavior is that of a *limiting distribution* of a chain. This is a limit of the form

$$\pi_j(\infty) = \lim_{t \to \infty} P(X_t = j) \text{ or, using vectors and matrices,}$$

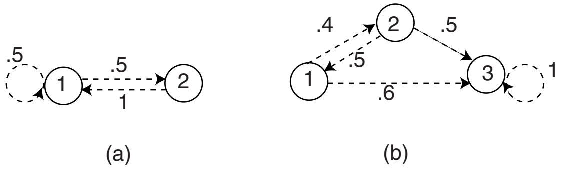
$$\pi(\infty) = \lim_{t \to \infty} \pi(t) = \lim_{t \to \infty} \pi(0)P^t \tag{23}$$

when we have convergence. We see that a steady-state distribution $\pi$ is also a limiting distribution by taking $\pi(0) = \pi$ here and using (22). Conversely, any limiting distribution is also a steady-state distribution. To establish this, we have

$$\pi(\infty)P = (\lim_{t \to \infty} \pi(0)P^t)P = \lim_{t \to \infty} \pi(0)P^{t+1} = \pi(\infty)$$

showing that $\pi(\infty) = \pi(\infty)P$. Because of this, in later sections of this chapter we do not carefully distinguish between steady-state distributions and limiting distributions.

For some chains, the limit in (23) may fail to exist. Existence is guaranteed by irreducibility along with a new property, *aperiodicity,* which requires that, for sufficiently large $s$, every entry of $P^s$ is positive. Under irreducibility and aperiodicity, the limits of (23) exist and are equal for any initial distribution. The common limit is the unique steady-state distribution $\pi$. In what follows, we always assume aperiodicity and irreducibility, unless otherwise stated.



(a) (b)

**Figure 1** Two Simple Markov Chains.

Given a Markov chain, a natural question to ask is What is the long-run fraction of time that the chain spends in a given state $i \in S$? The idea of long-run behavior is quite different from that of steady-state/limiting distributions. In the latter, the motivating idea is to understand the behavior of the chain at some fixed time very far in the future. But the question of long-run behavior concerns what happens to the chain over a very long time interval. This distinction will be discussed further in the context of queues in Subsection 5. The answer to the question above is again the solution $\pi$ to the steady-state equations (22), as long the chain is irreducible. To illustrate, the chain of Example 3.1 spends two-thirds of its time in state 1 and one-third in state 2, in the long run.

We summarize the discussion of steady-state behavior of Markov chains as follows. For "well-behaved" chains, such as irreducible, aperiodic, finite-state chains, there is a unique steady-state distribution that is also the limiting distribution. Furthermore, this distribution gives the long-run fraction of time spent in each state. It may be computed easily, by solving the steady-state equations (22). The steady-state behavior of the chain does not depend on the initial state.

Many interesting models require an infinite state space. The results just stated for the finite-state case extend to the infinite-state case once we adopt the condition of *positive recurrence* in addition to irreducibility and aperiodicity. A chain is positive recurrent if the expected time to return to any state is finite. We now turn to a simple infinite-state example.

## 3.4. Analysis of a Simple Discrete-Time Queue

In this subsection, we develop a very simple queueing model. This model is a Markov chain $L(t)$ representing the number of jobs present in a queueing system observed at regular discrete times $t = 0, 1, 2, \ldots$ The state space is $\{0, 1, 2, \ldots\}$. There are two types of transitions possible: arrivals and departures. We write $p$ for the probability that a job arrives in the next time step. We write $q$ for the probability that a job will complete service in the next time step, assuming that there is at least one job present ($L(t) > 0$). If we write $r$ for $1 - p - q$, which is the probability of no state change when there is at least one job present, then the transition matrix of the chain is

$$p = \begin{pmatrix} 1-p & p & 0 & 0 & \ldots \\ q & r & p & 0 & \ldots \\ 0 & q & r & p & \ldots \\ 0 & 0 & q & r & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{pmatrix}$$

If there are no jobs present, then there cannot be a departure, and this is why the first row of $P$ does not follow the same pattern as the others. In the case of this chain, the steady-state equations (22) become

$$\pi_0(1 - p) + \pi_1 q = \pi_0 \text{ and, for } i > 1,\ \pi_{i-1} p + \pi_i(1 - p - q) + \pi_{i+1} q = \pi_i$$

These may be reorganized as

$$-p\pi_0 + q\pi_1 = 0, \text{ and, for } i > 1,\ - p\pi_{i-1} + q\pi_i = -p\pi_i + q\pi_{i+1}$$

Thus $p\pi_i = q\pi_{i+1}$ for all $i$, and, once we require that the $\pi_i$'s sum to 1, we get the following unique solution:

$$\pi_i = (1 - \rho)\rho^i,\ i = 0, 1, 2, \ldots, \text{ when } \rho = \frac{p}{q} \tag{24}$$

From this, the steady-state probability that the system is empty is $\pi_0 = 1 - \rho$. This is also the long-run fraction of time that the system is empty (see Subsection 3.3). The probability that the system is not empty is $1 - \pi_0 = \rho$. Fleshing out the interpretation of the model, we imagine that the server is busy as long as there is at least one job in the system. Then $\rho$ may be interpreted as the long-run proportion of time that the server is busy. This is known as the *server utilization*. The distribution (24) is called the *geometric distribution with parameter* $\rho$. Its mean is given by

$$\frac{\rho}{1 - \rho}, \tag{25}$$

and this may be described as the steady-state expected number of jobs in the system. As the utilization $\rho$ approaches 1, the expected number of jobs in the system increases in approximate inverse proportion to $1 - \rho$. If the utilization is 99%, then the expected number of jobs waiting is 99. This quantifies the trade-off between utilization of the server, which we would like to keep high, and work-in-process, which we would like to keep low. This is a significant insight into the behavior of queues, gained from a very simple model.

### 3.5. Markov Chains in Continuous Time

Suppose now that we have a stochastic process $X(t)$, $t \geq 0$, observed for all times, not just integer times. We take the state space to be the finite set $\{1, 2, \ldots, n\}$ as before. The Markov property in this situation is just as described in Subsection 3.1 for the discrete-time case: knowing the state of $X$ at a time $t$, its evolution after time $t$ is independent of its history before time $t$. Briefly, the future is independent of the past, given the present.

Just as the rule whereby a discrete-time chain evolves is defined by a matrix $P$, the transition matrix, the rule whereby a continuous-time chain evolves is defined by a matrix $Q$, the generator matrix. The off-diagonal entry $q_{ij}$, $i \neq j$, of $Q$ has the interpretation that $q_{ij}h$ is approximately the probability that the chain, starting from state $i$ at a time $t$, makes a transition to state $j$ by time $t + h$, where $h$ is small:

$$P(X(t + h) = j \,|\, X(t) = i) \approx q_{ij}h \quad \text{for} \quad i \neq j \tag{26}$$

(The error in this approximation is small compared to $h$.) Because of this, $q_{ij}$ is called the *transition rate* from state $i$ to state $j$. It may be described more precisely as the rate of transitions into state $j$ while the process is in state $i$. The diagonal entries $q_{ii}$ of $Q$ are determined by the condition that the rows of $Q$ must add to 0, which is analogous to the property that the rows of a transition matrix add to 1. So $q_{ii}$ is negative and we denote it by $-q_i$, where

$$q_i = \sum_{j \,|\, j \neq i} q_{ij} \tag{27}$$

the sum on the right being over all states except $i$. (A technical condition assumed throughout is that these quantities are finite.) The transition matrix of $X$ over a time $t$ is defined by

$$P(t) = (p_{ij}(t)), \quad \text{where } p_{ij}(t) = P(X(t) = j \,|\, X(0) = i) \tag{28}$$

(Here and below, the notation $(a_{ij})$ means the matrix of the $a_{ij}$'s.) Soon we shall develop a simple formula for $P(t)$ in terms of the basic data of the chain, its generator $Q$.

By the Markov property of $X$, the process $X(0)$, $X(h)$, $X(2h)$, . . . that results when we observe $X$ at intervals of length $h$ is a discrete-time Markov chain with transition matrix $P(h)$. If we think of $h$ as a small time interval, then equation (26) shows that the transition probabilities of this discrete-time chain are given approximately by

$$\begin{gathered} p_{ij}(h) \approx q_{ij}h \quad \text{for} \quad i \neq j, \text{ and } p_{ii}(h) \approx 1 - q_i h, \text{ or,} \\ \text{in matrix form, } P(h) \approx I + hQ \end{gathered} \tag{29}$$

The rows of the matrix $I + hQ$ add to 1 because those of $Q$ add to 0.

We have related the continuous-time chain to a discrete-time chain with a fast clock, whose time unit is the small quantity $h$ but whose transition probabilities $p_{ij}(h)$ are proportionately small for $i \neq j$ by (29). This allows us to analyze the continuous-time chain using discrete-time results. All the basic calculations for continuous-time, finite-state Markov chains may be carried out by taking a limit as $h \to 0$ of the discrete-time approximation. For example, the transition matrix $P(t)$, defined in (28), may be derived as follows. We divide the time interval $[0, t]$ into a large number $N$ of short intervals of length $h = t/N$, so that the transition matrix $P(t)$ is the $N$-step transition matrix corresponding to $P(h)$. It follows from (29) that $P(t)$ is approximately the $N$-step transition matrix corresponding to the transition matrix $I + hQ$. This approximation becomes exact as $h \to 0$, and we have

$$\begin{aligned} P(t) &= (p_{ij}(t)) = (P(X(t) = j \,|\, X(0) = i)) \\ &= \lim_{h \to 0} (I + hQ)^N = \lim_{N \to \infty} \left(I + \frac{tQ}{N}\right)^N = e^{tQ} \end{aligned} \tag{30}$$

In the final equality here we are using the well-known limit $\lim_{N\to\infty} (1 + x/N)^N = e^x$, which holds even when $x$ is replaced by a matrix, here by $tQ$. The exponential of a matrix is defined by the Taylor series

$$e^A = I + A + \frac{A^2}{2} + \ldots = \sum_{n=0}^{\infty} \frac{A^n}{n!}$$

Now, some consequences of (30). If we differentiate with respect to $t$ we get

$$\frac{d}{dt} e^{tQ} = Qe^{tQ} \quad \text{or} \quad P'(t) = QP(t), \text{ and similarly } P'(t) = P(t)Q \tag{31}$$

These are called *Kolmogorov's backward and forward equations,* respectively. The *Chapman–Kolmogorov equations* $P(s + t) = P(s)P(t)$ may be deduced from (30) as follows:

$$P(s + t) = e^{(s+t)Q} = e^{sQ+tQ} = e^{sQ}e^{tQ} = P(s)P(t) \tag{32}$$

The generator $Q$ determines how a continuous-time Markov chain evolves via (26). There is another, more direct prescription for the evolution of the chain in terms of $Q$. If the chain is now in state $i$, then the time $T$ until the next change of state has the exponential distribution with rate $q_i$:

$$P(T \leq t) = 1 - e^{-q_i t}, \ t \geq 0 \tag{33}$$

When the chain does leave state $i$, it chooses its next state $j \neq i$ according to the probabilities $q_{ij}/q_i$. Why is the time $T$ exponentially distributed? Because the Markov property of $X$ implies that $T$ must have the memoryless property (13), and this in turn implies that $T$ has the exponential distribution.

*Example 3.2:* Let us illustrate this using the machine-failure example introduced in Subsection 3.1, now reworked in continuous time. The state space $S = \{0, 1\}$ has two states, representing inoperational and operational, as before. The order-2 generator matrix is of the form

$$Q = \begin{pmatrix} -q_0 & q_0 \\ q_1 & -q_1 \end{pmatrix}$$

Here, $q_0$ may be interpreted as the rate of repair when the machine is inoperational, and $q_1$ as the rate of breakdown when the machine is operational. In other words, the chain evolves with alternating exponentially distributed inoperational and operational periods, the inoperational periods having rate $q_0$ and the operational periods having rate $q_1$.

The Poisson counting process of Section 2 is a continuous-time Markov chain $N$ on the infinite state space $\{0, 1, 2, \ldots\}$, with generator

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \ldots \\ 0 & -\lambda & \lambda & 0 & \ldots \\ 0 & 0 & -\lambda & \lambda & \ldots \\ 0 & 0 & 0 & -\lambda & \ldots \\ \ldots & & \ldots & & \ldots \end{pmatrix} \tag{34}$$

Thus, the transitions are always from a state $n$ to the state $n + 1$. The transitions are, of course, arrivals because they cause the count $N$ to increase by 1. The probability of a transition in a short interval of time $h$ is approximately $\lambda h$ for any $n$ by (26). This observation corresponds precisely with the description of the Poisson process in terms of coin tossing in Section 2. Moreover, the fact that the time between arrivals in a Poisson process is exponential may be seen now as a consequence of the fact, expressed in (33), that the holding times in any continuous-time Markov chain are exponentially distributed.

Let us briefly describe the long-run behavior of continuous-time, finite-state Markov chains. We assume irreducibility as before, which in this case means simply that the entries of $P(t)$ are all positive for $t > 0$. Periodicity does not arise in continuous time. There is a unique distribution $\pi$ satisfying

$$\pi P(t) = \pi \text{ for all } t \geq 0$$

This is the steady-state distribution of the chain. By differentiation at $t = 0$ using (31), we find that

$$\pi Q = 0, \text{ or } \pi_j q_j = \sum_{i|i \neq j} \pi_i q_{ij} \text{ for all states } j = 1, 2, \ldots, n \tag{35}$$

This relates $\pi$ to the parameters of the chain—the entries of its generator $Q$. These are the steady-state equations in the continuous-time case. For finite-state irreducible chains, these equations have a unique solution whose components add to 1, and this solution is the steady-state distribution $\pi$. As in the discrete-time case, $\pi$ is also the limiting distribution of the Markov chain and gives the long-run proportion of time spent in each state. These results extend to the infinite-state case, assuming positive recurrence, as in Subsection 3.3.

*Example 3.3:* The steady-state distribution of the machine-failure model of Example 3.3 is given by

$$\pi_0 = \frac{q_1}{q_0 + q_1} \text{ and } \pi_1 = \frac{q_0}{q_0 + q_1}$$

This may be seen by solving the equations (35) or by noting that the steady-state probabilities must be proportional to the mean times spent in the states, which are given by (12) as $1/q_0$ and $1/q_1$.

### 3.6. Reversible Markov Chains and Birth-and-Death Models

Now we consider some special continuous-time Markov chains that arise in discussing simple queueing models. We restrict attention to the continuous-time case, as this is the setting of most of the practical models treated later.

Let $X$ be a continuous-time Markov chain with generator $Q$ and steady-state distribution $\pi$. The quantity $\pi_i q_{ij}$, sometimes called the *flux from i to j*, is the rate at which transitions from $i$ to $j$ occur in steady state. Contrast this with the transition rate $q_{ij}$ itself, which is the rate of transitions from $i$ to $j$ when the chain is in state $i$.

Suppose now that $X$ has state space $S = \{0, 1, 2, \ldots\}$ and each transition is to a neighboring state: the chain goes either up by one or down by one on any transition. Such a process is known as a *birth-and-death process* and is characterized by the *birth rates* $\lambda_i = q_{i,i+1}$ and *death rates* $\mu_i = q_{i,i-1}$. The generator of a birth-and-death process takes the form

$$Q = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & \ldots \\ \mu_1 & -\mu_1 - \lambda_1 & \lambda_1 & 0 & \ldots \\ 0 & \mu_2 & -\mu_2 - \lambda_2 & \lambda_2 & \ldots \\ 0 & 0 & \mu_3 & -\mu_3 - \lambda_3 & \ldots \\ \ldots & & \ldots & & \ldots \end{pmatrix} \tag{36}$$

Now, in the long run, the rate of transitions from state $i$ to $i + 1$ must equal the rate from $i + 1$ to $i$, because the number of the former type of transition must always be within 1 of the number of the latter type, and therefore

$$\pi_i q_{ij} = \pi_j q_{ji} \tag{37}$$

for $j = i + 1$. This equation states that the flux from $i$ to $j = i + 1$ equals the flux from $j$ to $i$. This is easily seen to be true not just for neighboring pairs $i, j$, but for all pairs of distinct states $i \neq j$. The equations (37) are called the *detailed balance equations*. It is easy to deduce the steady-state distribution from (37), the result being

$$\pi_n = C \frac{\lambda_0 \lambda_1 \ldots \lambda_{n-1}}{\mu_1 \mu_2 \ldots \mu_n}, \; n = 0, 1, 2, \ldots \tag{38}$$

where $C$ is a constant chosen to ensure that the $\pi_n$'s add to 1. (If no such constant exists, then the process is not positive recurrent.) This formula determines the steady-state distribution of a great many simple queueing models, several of which will be discussed in the next section.

A Markov chain satisfying (37) for all pairs of distinct states $i \neq j$ is said to be *reversible,* a somewhat unfortunate choice of term referring to the fact that, if viewed in reverse time while in steady state, it is probabilistically indistinguishable from its forward-time version. (Such a chain might be described more precisely by saying that it is unchanged by time reversal rather than reversible.) Many of the queueing models to be treated later are in fact Markov chains satisfying the property (37) or some related property. See Kelly (1979) for a thorough treatment of such chains.

## 4. SIMPLE QUEUEING MODELS

Of central interest in designing industrial and service systems is the problem of providing service to a stream of arriving customers or jobs. Because of irregularities in the arrival or service processes, there are times when too much work has arrived in too short a time and so jobs have to wait for service. Even though the system has the capacity to serve all arrivals, randomness generally results in some waiting. Waiting could be eliminated if the irregularity could be eliminated, without increasing overall service capacity or diminishing the overall flow of arriving work. Waiting is therefore a consequence of irregularity. This waiting is central: it occupies resources in the form of waiting space, contributes to work-in-process, causes due dates to be missed, and may cause machines and workers to be starved of work. Much system design focuses on minimizing waiting time. Therefore, many stochastic models also focus on waiting, and models of waiting are called queueing systems, or simply queues.

A queueing system may be divided loosely into three subsystems: the arrival process, the waiting area, and the service system. Each of these subsystems can operate in a variety of ways. The following are some of the more common possibilities. The arrival process specifies a sequence of jobs or batches of jobs and the times at which they enter the system. The arrival times may constitute a Poisson process or a renewal process, for example. The jobs may be indistinguishable from one another, or they may be of several classes, to be treated differently in the waiting or service area. The waiting area may be managed in various ways. Jobs may be ordered according to time of arrival, the most recent arrival being the first to be served (first-in-first-out, or FIFO). This is the most common discipline used in serving people because of the sense of fairness it engenders. To give some simple alternatives, jobs may be served in random order (SIRO) or last-in-first-out (LIFO). Under the processor sharing (PS) discipline, all jobs present share the server's attention equally. Jobs may also be served according to priorities determined based on job class or service needs. Preemption, in which an arriving high-priority job ejects a low-priority job from service, may or may not be allowed. The service system is where the jobs receive what they have waited for. There may be one or many servers, and the servers may be subject to breakdowns. Jobs may be served singly or in batches, and service times may depend on job class. See Hall (1991) for a discussion of modeling real systems as queues, with a careful treatment of the role of these three subsystems.

### 4.1. Notation

The queues of this section all belong to the class $G/G/s/K$. The four elements in this notation have the following meanings. The first element refers to the arrival process, $G$ indicating generally distributed interarrival times. Independence of these times is assumed. Thus, the first $G$ indicates that arrivals form a renewal process. The second element describes the service times, the $G$ again indicating a general distribution. Independence is again assumed, and moreover, it is assumed that interarrival times are independent of service times. The third element, $s$, is the number of servers, and the last element, $K$, is the capacity of the system. $K - s$ is therefore the capacity of the waiting area. If the parameter $K$ is not present, an unlimited capacity is assumed. If one of the $G$'s is replaced by $M$, this indicates that the corresponding distribution is exponential. The $M$ connotes the property of memorylessness (13). For example, the $M/M/1$ queue is the special case of the $G/G/1$ queue in which the arrivals are Poisson and the service distribution is exponential. $D$, standing for deterministic, is used to indicate constant service or interarrival times.

It is standard to denote the long-run arrival rate of a queue by $\lambda$ and the long-run service rate (of one server) by $\mu$. To avoid having to deal with unimportant special cases, we always assume that these rates are positive—that is, that they are not zero. In the case of renewal arrivals and independent, identically distributed service times, the expected interarrival time is then $1/\lambda$ and the expected service time is $1/\mu$. Throughout, $\rho$ denotes the *traffic intensity* $\lambda/\mu$ of the arriving stream of jobs, which is the rate at which work arrives, work itself being measured in terms of the time it takes a server to perform it. Another quantity of importance is the server utilization, which is the proportion of time that a server is busy. For a single-server queue, the utilization and the traffic intensity are equal.

$L(t)$ will denote the load, or number of jobs in the system, waiting or being served, at time $t \geq 0$. For each model considered in this section, the distribution of $L(t)$ approaches a limit as $t \rightarrow \infty$, and the limit is referred to as the steady-state distribution of the number in the system. (As we said in Subsection 3.3, we do not distinguish carefully between steady-state and limiting distributions.) We denote by $L$ a random variable whose distribution is this steady-state distribution. Let $W_n$ denote the time-in-system or *flow time* of the $n$th arrival to the system. Then, for the models considered here, the distribution of $W_n$ also has a limit as $n \rightarrow \infty$, which we call the *steady-state flow-time distribution*. We denote by $W$ a random variable with this distribution. In the same way we can define a random variable $W_Q$ whose distribution is the steady-state distribution of waiting time. Flow time and waiting time differ only in that the latter does not include service time. Most of the results given here concern the expected values of the random variables $L$, $W$, and $W_Q$ for various queues. These expected values are called *steady-state expected values.* For some technicalities concerning steady-state and long-run averages, see Subsection 5.1. For more on these models, see the books referred to in Subsection 1.1 or any of the many introductory texts on queues.

## 4.2. Simple Markovian Queueing Models

In this subsection we treat several queues of the $M/M/s/K$ type. These queues have Poisson arrivals, exponential service times, $s$ servers, and capacity $K$. For these queues, the number-in-system $L(t)$, $t \geq 0$, is a continuous-time Markov chain, in fact, a birth-and-death process (Subsection 3.6). The Markov property arises from the exponentiality of service and interarrival times—see the discussion following (33). The queueing discipline is taken to be FIFO in every case. The results presented here follow fairly directly from (38).

### *4.2.1. M/M/1 Queue*

This is the single-server queue with Poisson arrivals and exponential service. It is the continuous-time analog of the simple queue treated in Subsection 3.4. It may be viewed as a birth-and-death process, with generator (36) where $\lambda_n = \lambda$ and $\mu_n = \mu$ for all $n \geq 0$. The fact that the birth rates are constant is what makes the arrival process Poisson. Suppose that the traffic intensity $\rho = \lambda/\mu$ is less than 1. This condition is an example of the general *capacity condition* (57) below, as it concerns the capacity of the server to handle the arriving work. It implies that the process is positive recurrent. The steady-state distribution of the number in the system $L$ may be derived directly from (38) and is the same geometric ($\rho$) distribution that arose in (24) for the discrete-time queue of Subsection 3.4. The steady-state distribution of flow time $W$ is exponential with rate $\mu(1 - \rho)$. The steady-state expected number-in-system, time-in-system, and waiting time are

$$E(L) = \frac{\rho}{1-\rho};\ E(W) = \frac{1}{\mu(1-\rho)};\ E(W_Q) = \frac{\rho}{\mu(1-\rho)} \tag{39}$$

### *4.2.2. M/M/2 Queue*

Here we increase the number of servers to two. Assume now that $\rho < 2$, so that the two servers are sufficient to handle the arriving work. The $M/M/2$ queue is the birth-and-death process with $\lambda_n = \lambda$ for all $n \geq 0$, $\mu_1 = \mu$, and $\mu_n = 2\mu$ for $n \geq 2$. The new $\mu_n$'s express the idea that when there are two or more jobs in the system, both servers are busy and so the overall rate at which service is provided is doubled. The steady-state probabilities are given by

$$\pi_0 = \frac{2-\rho}{2+\rho};\ \pi_n = 2\pi_0\left(\frac{\rho}{2}\right)^n,\ n > 0 \tag{40}$$

For this queue we record the steady-state expected number-in-system and flow time:

$$E(L) = \frac{\rho}{1-\rho^2/4};\ E(W) = \frac{1}{\mu(1-\rho^2/4)} \tag{41}$$

### *4.2.3. M/M/∞ Queue*

This is the birth-and-death process with $\lambda_n = \lambda$ and $\mu_n = n\mu$ for all $n \geq 0$. The service rate is proportional to the number of jobs in the system, and this captures the idea that each job is being served simultaneously at the same rate, $\mu$. The steady-state distribution of the number in the system is Poisson ($\rho$), again by (38), where $\rho = \lambda/\mu$ is the traffic intensity as usual. Thus, the expected number in the system is simply $E(L) = \rho$. Since there is no waiting in this system, other quantities are easy to derive but perhaps not very informative. For example, the distribution of the steady-state time-in-system $W$ is the same as the service distribution, namely, exponential with rate $\mu$.

### *4.2.4. Erlang Loss System M/M/s/s*

Although this model is associated with telephony rather than industrial engineering, its historical importance and simplicity justify mentioning it. Suppose we have a telephone exchange with $s$ circuits, meaning that it can handle at most $s$ simultaneous calls. The key measure of performance of this system is the fraction of calls that are "lost," i.e., that arrive to find all circuits busy. Erlang modeled this system as an $M/M/s/s$ queue in which calls arrive in a Poisson process of rate $\lambda$ and last an exponential time with mean $1/\mu$. The servers here are the circuits. This may be viewed as the birth-and-death process with $\mu_n = n\mu$ for $n \geq 0$, $\lambda_n = \lambda$ for $n \leq s$, and $\lambda_n = 0$ for $n > s$. The steady-state distribution $\pi$ may be found from (38), giving

$$\pi_n = \frac{\lambda^n/n!}{1 + \lambda + \lambda^2/2! + \ldots + \lambda^s/s!}, \; n = 0, 1, \ldots, s \tag{42}$$

With $n = s$, this is the proportion of time that all $s$ circuits are busy. This is also the proportion of arriving calls that are lost, because of the PASTA property to be discussed below in Section 5.5. Equation (42) in the case $n = s$ is the celebrated *Erlang loss formula*.

## 4.3. A Comparison of Systems

To illustrate what can be gained from the mathematical results of the previous subsection, we present a comparison of three systems. The three systems are identical in the load that must be served and in the service capacity available, but differ slightly in the queueing discipline. Here are the systems.

- *System 1* consists of two separate single-server queues, each having arrivals at rate $\lambda$ and service at rate $\mu$. This system is modeled as two independent $M/M/1$ queues.
- *System 2* consists of a single queue served by a pair of servers. The arrival stream has rate $2\lambda$, and each server serves at rate $\mu$. This is modeled as an $M/M/2$ queue.
- *System 3* consists of a single queue with an arrival stream of rate $2\lambda$ and a single server serving at rate $2\mu$. This is modeled again as an $M/M/1$ queue.

Denoting $\lambda/\mu$ by $\rho$, the overall traffic intensity in each case is $2\rho$ (not $\rho$). Using formula (39) above for the first and third systems and (41) for the second, the steady-state mean numbers-in-system for the three cases may be found, leading to

$$E(L_1) = 2\,\frac{\rho}{1-\rho} \geq E(L_2) = \frac{2\rho}{1-\rho^2} = \left(\frac{2}{1+\rho}\right)\frac{\rho}{1-\rho} \geq E(L_3) = \frac{\rho}{1-\rho}$$

Thus, from the viewpoint of work-in-process, the third system is the most favorable and the first is the least favorable. This has a simple explanation. System 1 compares poorly to system 2 in that the former may have an idle server when there are two jobs in the system, whereas this cannot occur in the latter. Also, system 2 compares poorly to system 3 in that the former serves at half the rate of the latter when there is only one job in the system. Combining the queues almost halves average work-in-process for high traffic intensities. Using a single fast server to serve the combined queue halves average work-in-process for all traffic intensities. The comparison is intuitive. The process of modeling has led to quantification of the intuition that is satisfying and natural, even if the assumptions of the models, such as exponentiality, are somewhat arbitrary. The advantage of maintaining a single queue served by several servers over having a separate queue for each server, demonstrated here in the comparison of system 1 and system 2, is well known and is put to good use in many retail banks and at airline check-in counters.

## 4.4. Models Based on the $M/G/1$ Queue

The $M/G/1$ queue is a single-server queue with Poisson arrivals and independent, identically distributed service times having an arbitrary distribution. Here we analyze this queue and several of its variants. The results show that quite a broad range of phenomena can be modeled simply enough to allow basic performance measures to be given by explicit formulas. As always, $\lambda$ and $\mu$ denote the arrival and service rates, respectively.

### *4.4.1. M/G/1 Waiting Time Formula*

The *Pollaczek–Khintchine formula* gives the steady-state expected waiting time for this queue as

$$E(W_Q) = \frac{\lambda E(S^2)}{2(1-\rho)} = \frac{\rho(1 + c_s^2)}{2\mu(1-\rho)} \tag{43}$$

where $S$ is a random variable whose distribution is the service distribution and $c_s^2$ is its SCV. The $M/M/1$ queue is a special case, for which $c_s^2 = 1$. See (39).

### *4.4.2. M/G/1 under a Priority Discipline*

Suppose that, instead of a homogeneous stream of arriving jobs, the jobs are of different classes that are treated differently in the queue. The classes range from 1 to $p$, say, and jobs of class $k$, $1 \leq k \leq p$, are given priority $k$. This means that a job of class $k$ in the queue would be allowed to enter service only if there were no higher priority jobs (i.e., jobs of class $k' < k$). The discipline is taken to be nonpreemptive, so that a low-priority job in service will complete service rather than being ejected by an arriving high-priority job. Jobs of class $k$ are assumed to arrive in a Poisson process, with rate $\lambda_k$, and are assumed to have their own characteristic service distribution function $G_k$. We write $\lambda$ for the overall arrival rate, $\rho_k^+$ for the traffic intensity of jobs of priority $k$ or higher, and $\mu_k$ for the service rate of class $k$:

$$\lambda = \sum_{i=1}^{p} \lambda_k,\ \rho_k^+ = \sum_{i=1}^{k} \frac{\lambda_i}{\mu_i} \text{ and } \frac{1}{\mu_k} = \int_0^\infty x dG_k(x)$$

We also introduce $S$, a random variable with the aggregated service distribution, whose distribution function is defined as $G = \Sigma\ \lambda_k G_k/\lambda$. With this notation, we have the following formulas for the steady-state expected waiting time of jobs of each class.

$$E(W_Q^1) = \frac{\lambda E(S^2)}{2(1 - \rho_1^+)} \text{ and, for } k > 1,\ E(W_Q^k) = \frac{\lambda E(S^2)}{2(1 - \rho_k^+)\ (1 - \rho_{k-1}^+)}$$

It may be shown from this that, in order to minimize the expected number of jobs in the system (work-in-process), the shorter jobs should receive higher priority. This is a manifestation of the general principle that short jobs should be served first. For details, see for example Buzacott and Shanthikumar (1993). This underlies the idea of having express checkout lines in supermarkets.

### *4.4.3. The M/G/1 Queue with Batch Arrivals*

Suppose that, instead of jobs arriving singly at the Poisson arrival times, they arrive in batches. Suppose that the batch sizes are independent and identically distributed. This arrival process is known as a *compound Poisson process* and is fundamentally more variable than the Poisson process. We assume the FIFO discipline. We write $m_B$ and $c_B^2$ for the mean and SCV of batch size. Then the traffic intensity is $\rho = \lambda m_B/\mu$, where $\lambda$ is the arrival rate of batches and $\mu$ is the service rate of individual jobs. It is easy to derive the steady-state expected waiting time in this system. The $n$th *batch waiting time* is the time between the arrival of the $n$th batch and initiation of service for the first job in the batch. By treating an entire batch as a single job, the steady-state batch waiting time may be deduced directly from the Pollaczek–Khintchine formula (43) as

$$E(W_{Q(\text{Batch})}) = \frac{\rho(1 + c^2)}{2\mu(1 - \rho)} \tag{44}$$

where $c^2$ is the SCV of the service time of an entire batch, which may be expressed in terms of the SCV of service for a single job as $c^2 = c_B^2 + c_s^2/m_B$. The expected waiting time for a job, rather than a batch, is found by adding to (44) the expected total service time of all jobs ahead of a typical job in a batch, and this gives

$$E(W_Q) = \frac{\rho(1 + c^2)}{2\mu(1 - \rho)} + \frac{1}{\mu}\frac{E(B(B - 1))}{2m_B} \tag{45}$$

where $B$ is a random variable having the batch-size distribution.

## 4.5. Examining the Assumptions of a Simple Queueing Model

In building a stochastic model for a real system, we make various simplifying assumptions for reasons of tractablility or parsimony. To be effective in modeling, we must appreciate the consequences of these assumptions. In this subsection we illustrate what is involved in this aspect of model building by considering an idealized example. We consider the $M/M/1$ queue as a model for a single worker performing an operation on arriving workpieces. The primary reason for adopting such a model is the mathematical tractability that comes as a consequence of the Markov property. Now we consider the implications of this model.

As we saw in Section 2, Poisson arrivals may be thought of as completely random arrivals in a very exact sense. They are neither too regular, as for example if they came once every five minutes, nor too irregular, as for example if they came in batches of between 1 and 20 with highly variable interarrival times. The Poisson process implies memorylessness, in that observations of past arrivals give no hint as to future arrivals. To expand on this property, if the arrivals to the real system have been unusually heavy recently, this is no indication that they will continue to be heavy, nor is it an indication that they will become less heavy. If actual arrivals tend to come in clusters, or if the arrival rate fluctuates throughout the workday, the Poisson model may be inappropriate. The qualitative behavior of the Poisson process may or may not be consistent with the actual nature of the system. The success of the model may depend heavily on having consistency here.

The assumption that the service times are independent implies, for example, that knowledge of one service time tells us nothing about the next service time. If one service time is particularly long, then this gives us no reason to expect the next one to be long, or short, or unusual in any way. In reality, there may be serial correlations between the service times, caused for example by fluctuations in the attentiveness of the worker or in the quality of the arriving workpieces. This, of course, violates the assumption of independent service times. These effects may give rise to a tendency to have several long or short service times in a row, which would tend to increase waiting times.

Service times are also assumed to follow the exponential distribution in the $M/M/1$ model. One feature of the exponential distribution is that it is highly variable. For this distribution, 14% of service times are over twice the average service time and 39% of service times are less than half the average. This variability is inconsistent with a routine manual task performed on uniform parts. It may be consistent with a cognitive task, such as diagnosis of the cause of a malfunction. So whether this assumption is reasonable depends heavily on the nature of the task being performed.

Good modeling requires that we understand the qualitative consequences of modeling assumptions and that we use this understanding to make choices consistent with the reality we are trying to model.

# 5. SOME GENERAL PRINCIPLES

## 5.1. Long-Run Behavior

The first focus of the mathematics of stochastic models is to determine long-run or steady-state behavior. This is because long-run behavior requires less detail to describe than the full evolution of the system, and it often gives a good sense of the overall behavior of the system over the time frame for which it is to be in operation. In interpreting results on long-run behavior, one must bear in mind a sense of how long a time interval is needed before average performance may be approximated by long-run performance.

### *5.1.1. Steady-State vs. Long-Run Averages*

As we saw in Subsection 3.3, there are several competing descriptors of long-run behavior, including steady-state expected values, which are averages with respect to the steady-state or stationary distribution, and long-run or time averages. Usually, the mathematics leads most easily to steady-state averages. This is exemplified by the fact that, in a Markov chain, the steady-state distribution $\pi$ is easy to characterize and to compute. *Ergodic properties* are equalities between these two kinds of averages. To relate this to queues, as before let us denote by $L(t)$ the number-in-system at time $t$ for a certain queueing system, and by $W_n$ the time-in-system for the $n$th job to arrive. Define the long-run average number-in-system by

$$\overline{L} = \lim_{T\to\infty} \frac{\int_0^T L(t)\, dt}{T} \tag{46}$$

Similarly, define the long-run average time-in-system by

$$\overline{W} = \lim_{n\to\infty} \frac{W_1 + W_2 + \ldots + W_n}{n} \tag{47}$$

Let $L$ and $W$ be random variables whose distributions are the steady-state distributions of the number-in-system and time-in-system for the queue, respectively. Then, in great generality, we have the ergodic properties

$$\overline{L} = E(L) \text{ and } \overline{W} = E(W) \tag{48}$$

The difficulty in understanding these statements is not to see that they are true but to distinguish between the concepts represented by the two sides of the equalities.

### *5.1.2. What Goes In Must Come Out*

This simple principle, more formally expressed, says that the long-run arrival rate to a system equals the long-run departure rate. The reason is clear: the difference between the number of arrivals and the number of departures over a time interval equals the change in the number of jobs actually in the system, and, as long as the number in the system is not allowed to grow relentlessly, the numbers of arrivals and departures cannot differ by very much. This simple fact leads quickly to the important *traffic equations* (53) for a queueing network.

### *5.1.3. Little's Law and Other Conservation Laws*

Suppose jobs arrive at a system, spend some time there, perhaps waiting and being served, and then depart. The system may be a warehouse or other storage system, a queueing system, or perhaps something more complex, such as a queueing network. Using the notation of (46) and (47), it may be shown in great generality that

$$\overline{L} = \lambda \overline{W} \tag{49}$$

This is *Little's law*. Illustrations of it may be seen in (39) and (41), once the ergodic relations (48) are taken into account. To justify (49) heuristically, following Ross (1997), if jobs arrive at rate $\lambda$ and on average spend time $\overline{W}$ in the system, then demand for occupancy of the system arrives at a rate of $\lambda\overline{W}$, and so occupancy must be supplied at a rate of $\lambda\overline{W}$ also, meaning that $\lambda\overline{W}$ jobs must be present on average.

This simple fact may be applied to various systems. For example, in the context of a queue, the "system" could be taken to mean the waiting area or the service area rather than the entire system. If applied to the service area of a single-server queue, Little's law yields

$$1 - P(L = 0) = \lambda \left(\frac{1}{\mu}\right), \text{ or } P(L = 0) = 1 - \rho \tag{50}$$

To see that this is a special case of (49), note that $1 - P(L = 0)$ is not only the steady-state probability that the server is busy, it is also the steady-state expected number of jobs in service, which in turn, by an ergodic property, is the time-average number of jobs in the system in question. Of course, $1/\mu$ is the expected time in service and plays the role of $\overline{W}$ in (49). When Little's law is applied to the waiting area, it tells us that the expected number of jobs waiting is the product of the arrival rate and the average waiting time.

## 5.2. Behavior of a Bottleneck Queue

A bottleneck station is a station with sufficiently high server utilization to cause substantial delays. These are also called stations in *heavy traffic*. In a manufacturing system, such stations are often the ones of greatest interest. They typically represent the critical resources and may be responsible for most of the work-in-process and waiting. Despite the complexity of the exact analysis of the $G/G/1$ queue, in heavy traffic the behavior is somewhat simple. It may be shown that the steady-state expected waiting time satisfies

$$E(W_Q) \approx \frac{\rho(c_a^2 + c_s^2)}{2\mu(1 - \rho)} \tag{51}$$

in the sense that the ratio of the two sides of the approximation converges to 1 as $\rho \rightarrow 1$. Here, $c_a^2$ and $c_s^2$ are the SCVs of interarrival and service time, respectively. For a more precise statement, see Asmussen (1987). This approximation is exact for the $M/G/1$ queue, in which case $c_a^2 = 1$. See (43).

A common feature of all the waiting-time formulas we have had, from the simple (25) to the general (51), is the presence of the factor $1 - \rho$ in the denominator. This is an important observation. As utilization at a station increases from 90% of capacity to 99%, waiting times [and so also loads, by (49)] typically increase by a factor of 10.

## 5.3. Deleterious Effects of Variability

In the Pollaczek–Khintchine formula, increasing the SCV of service causes the mean waiting time to increase also. This illustrates the general principle that increasing variability reduces performance. The same principle is illustrated more fully by the approximation (51) for the steady-state expected waiting time in a $G/G/1$ queue in heavy traffic. There we see that the SCVs of the interarrival and service times contribute equally to the approximate expected waiting time.

In the $G/G/1$ queue, it is not true in complete generality that increasing the *variance* of service times, say, will cause average waiting time to increase. To formalize this intuitive idea, we need a sufficiently stringent definition of increased variability. See Ross (1996) for a discussion of the stochastic ordering that makes the intuition precise.

## 5.4. Rate of Convergence to Steady State

An important example of the usefulness of the mathematics of stochastic models arises when simulating queues with fairly high server utilizations. It is not unusual to find that the mean waiting time, say, is highly variable. Here is a fairly common experience. We carry out a simulation run on a model of a manufacturing system, involving the processing of $n = 100{,}000$ jobs. We compute the mean waiting time $\overline{W}_n$ over the run. Now we repeat the experiment, with a different seed for the random number generator, and compute another mean waiting time $\overline{W}'_n$. $\overline{W}_n$ and $\overline{W}'_n$ may well differ by on the order of 20%. The source of this extreme variability may not be clear from the model being simulated, and yet such variability in a real system would be indicative of poor design. To explain this phenomenon mathematically, suppose we are simulating a very simple system, say a $G/G/1$ queue. It may be shown that the asymptotic distribution of the mean $\overline{W}_n$ of the first $n$ waiting times, for $n$ large and $\rho$ near 100%, is normal with expected value given by (51) and CV

$$\frac{1}{(1-\rho)}\sqrt{\frac{2(c_a^2 + c_s^2)}{n}} \tag{52}$$

(See Whitt 1989 and Asmussen 1992 for more on these limit results.) Because we must choose the simulation run-length $n$ large enough to make this small, a substantial multiple of $(1 - \rho)^{-2}$ waiting times is needed to get a reliable estimate of $\overline{W}$ from simulation. If $\rho = 0.95$, then $(1 - \rho)^{-2} = 400$, and for an $M/M/1$ queue, for which $c_a^2 = c_s^2 = 1$ (see Subsection 1.3), the indications are that about 160,000 waiting times are needed to estimate $\overline{W}$ to about 20% accuracy with 95% probability. One interpretation of this result is that the queue converges to steady state very slowly for high traffic intensities, resulting in persistent correlations between the successive waiting times and slow attenuation of the variance of the mean waiting time as run-length $n$ increases. This capability that the mathematical analysis of stochastic models provides to explain qualitatively and thereby validate the behavior of simulations is perhaps one of its most important uses in engineering.

### 5.5. ASTA and PASTA

Consider an $M/G/1$ queue with batch arrivals, the average batch size being 10 jobs. Suppose that the traffic intensity is low, say $\rho = 0.1$. Then arriving jobs see the queue in atypical conditions because 90% of them see other jobs ahead of them in the queue, whereas, since the traffic intensity is 0.1, in fact the queue is empty 90% of the time by (50). So jobs may tend to arrive when the station is busy, as in the example just given, or they may tend to arrive when it is not busy. The latter occurs in the $D/D/1$ queue, in which arrival and service times are constant. Suppose that initially the queue is empty and each interarrival time is 1 unit while each service time is 0.9 units. Then, although the utilization is 90%, no job ever has to wait. Incidentally, this illustrates the remark made earlier that waiting can be eliminated without increasing server capacity if variability can be eliminated.

What of the $M/M/1$ queue? Again, it illustrates something interesting—a perfect balance between the two extreme examples of the previous paragraph. It illustrates the property that *Poisson arrivals see time averages* (PASTA). More informally, Poisson arrivals see typical conditions. This property means that if we were to make observations on the queue at the arrival times, the distribution of these observations would be the time-average distribution of the system. In particular, the average number of jobs in the system just before arrival times is the same as the long-run average number of jobs in the system. Similarly, the proportion of arrivals that find the queue empty is precisely $1 - \rho$, the long-run proportion of time that the queue is empty.

This property extends to some more complex situations. For example, in the *open product-form queueing networks* to be discussed in the next section, the arrivals to the individual queues see the entire network in time-average (or steady-state) conditions. These arrivals are not necessarily Poisson, and so the acronym ASTA (arrivals see time averages) has been coined to describe this situation.

## 6. QUEUEING NETWORKS

A queueing network is a collection of stations among which jobs move and compete for service. Each station is a queueing system in its own right. Queueing networks are commonly used as models for manufacturing, telecommunication, and transportation systems. These models are usually analyzed by simulation. The level of complexity of queueing network models, from the viewpoint of mathematical analysis, is far beyond that of the individual stations. This is because the process of going from a single queue to a queueing network is the process of increasing the dimensionality of the model—going from a single dimension to many. As in all areas of applied mathematics, this brings with it a qualitative increase in difficulty. This is Bellman's "curse of dimensionality." Because of this, queueing networks must be approached with modest expectations as to what may be accomplished.

To understand what is known about queueing networks, one must recognize first that some networks have truly complex behavior and that a general and detailed theoretical understanding of them is probably unattainable. On the other hand, there is a remarkable class of networks, the *product-form networks,* for which a great deal is known. To a certain extent, these networks behave as if they consisted of collections of independent stations, and to that extent the stations may be analyzed separately. Thus, a high-dimensional problem is replaced by several low-dimensional problems. In Subsections 6.1 and 6.2, we treat product-form networks. Then, to highlight the special nature of these networks, in Subsection 6.3 we discuss some networks whose behavior is complex and difficult to analyze. In particular, these networks may allow queues to grow without bound even when there is sufficient service capacity to process all arriving work. In between these extremes, there appear to be many queueing networks which, although not product form, are fairly amenable to simple *decomposition approximations.* This is the topic of Section 7.

It is difficult to describe the family of product-form networks qualitatively. One attempt is to say that they are a family of networks in which jobs behave in a highly random but nonconspiratorial manner. This is to say, the jobs do not interfere with one another in complex ways: they do not tend to batch together in their movements, and when a job arrives at a station it tends to see that station, and the larger network, in a typical condition, in the sense of the ASTA property discussed in Subsection 5.5. In particular, jobs are neither more nor less likely to arrive when a station is heavily loaded.

### 6.1. Jackson Networks

We describe the famous *Jackson networks* and discuss their remarkably simple steady-state distributions. Consider a network of $M$ FIFO stations. Let $s_m$ denote the number of servers at station $m$, $m = 1, 2, \ldots, M$. *Exogenous* arrivals, that is, arrivals from outside the network, enter station $m$ in a Poisson process of rate $\lambda_m^*$. Jobs at station $m$ require exponential service with rate $\mu_m$. When a job completes service at a station, it either visits another station or leaves the network. The probability that a job that has just completed service at station $m$ next goes to station $n$ is written as $p_{mn}$, the choice of next station being independent of everything that has happened up to the time that choice is made. These routing probabilities are collected into an $M \times M$ routing matrix $P = (p_{mn})$. The technical condition that $I - P$ be invertible is needed to ensure that all jobs eventually leave the network.

This is quite a complicated process. Jackson's (1957) remarkable discovery was that the steady-state distribution is very simple. To identify it, we first find the overall arrival rate $\lambda_n$ at each station $n$—this is the total of the exogenous arrival rate and the arrival rate of jobs from within the network. The exogenous arrival rate is $\lambda_n^*$. The exogenous arrival stream must be combined with the streams of jobs routed to station $m$ after completing a service. A proportion $p_{mn}$ of the jobs completing service at station $m$ is routed to station $n$, for each $m = 1, 2, \ldots, M$. But the departure rate from station $m$ is the same as its arrival rate, $\lambda_m$ ("what goes in must come out"—see Section 5.1), and so we have

$$\lambda_n = \lambda_n^* + \sum_{m=1}^{M} \lambda_m p_{mn} \tag{53}$$

These equations are called the traffic equations. Invertibility of $I - P$ implies that these equations have a unique solution in the $\lambda_m$s. Suppose now that $\rho_m < s_m$, $\lambda_m > 0$ and $\mu_m > 0$ for each $m = 1, 2, \ldots, M$. These conditions ensure that the network is an irreducible, positive recurrent Markov chain. Jackson proved the following result: The steady-state distribution of the number of jobs at each station in the Jackson network is the same as that of $M$ independent stations, the $m$th being an $M/M/s_m$ queue with arrival rate $\lambda_m$ and service rate $\mu_m$.

Here is a simple consequence. If we take each station to be a single-server queue, and denote by $\rho_m$ the traffic intensity $\lambda_m/\mu_m$ of the $m$th station, then the steady-state expected flow time (of an exogenous arrival), written $E(T)$, of a job through the network is

$$E(T) = \frac{1}{\sum_{m=1}^{M} \lambda_m^*} \sum_{m=1}^{M} \frac{\rho_m}{1 - \rho_m} \tag{54}$$

To explain, the rightmost sum is the expected total number in the system by (39). This is divided by the overall exogenous arrival rate to get the average time-in-system, using Little's law (49).

*Example 7.1:* Consider the simple network depicted in Figure 2. There are two stations in tandem, processing an exogenous stream of rate $\lambda_1^* = 1$ arriving at the first station. There are no exogenous arrivals at the second station, so $\lambda_2^* = 0$. The service rates are $\mu_1 = 5$ and $\mu_2 = 4$. For any job completing service at the second station, there is a probability $p_{21} = 0.5$ that it will be sent back through the two stations again. Otherwise, it departs the network. By solving the traffic equations (53), we find the total arrival rates at the stations to be $\lambda_1 = \lambda_2 = 2$. Therefore the traffic intensities are $\rho_1 = \lambda_1/\mu_1 = 2/4 = .5$ and $\rho_2 = \lambda_2/\mu_2 = 2/5 = 0.4$, which are both less than 1 and so the

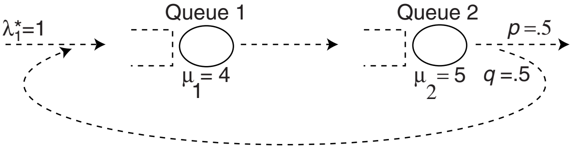


**Figure 2** A Simple Queueing Network.

network is positive recurrent. Now from (54), the steady-state expected flow time of an exogenous arrival through the network is

$$E(T) = \frac{1}{\lambda_1^*}\left(\frac{\rho_1}{1-\rho_1} + \frac{\rho_2}{1-\rho_2}\right) = 1 + \frac{2}{3} = 1.666$$

The mean waiting times per arrival (rather than per exogenous arrival) at the two stations are given by

$$E(W_Q^{-1}) = \frac{0.5}{(4)\,(1-0.5)} = 0.25 \text{ and } E(W_Q^2) = \frac{0.4}{(5)\,(1-0.4)} = 0.133$$

## 6.2. General Product-Form Networks

In the Jackson network, the jobs at a station are indistinguishable in that they all have the same service requirements and all follow the same routing rules. This is a feature possessed by few real-world systems. Jobs in a queue often differ in urgency, routing, and service requirements. In this subsection we discuss *multiclass product-form networks,* in which jobs are distinguishable into classes, each class having its own distinctive routing pattern. In the networks considered, jobs behave in the same highly random but nonconspiratorial manner we observed in Jackson networks. We also consider some variations involving nonexponential service distributions and priority disciplines.

The first step in defining multiclass networks is to introduce a framework for describing a variety of routing rules in a simple, unified way. In this framework, each job in the network has a class. This class characterizes which station the job is presently at and the rule whereby it chooses its next class. Each class is associated with one and only one station, and so, when a job chooses its next class, this also determines its next station. Suppose our network is composed of $M$ stations serving a total of $K$ classes of jobs. Let $P = (p_{k\ell})$ denote a $K \times K$ *routing matrix,* which is to say $p_{k\ell}$ is the probability that a class-$k$ job becomes a class-$\ell$ job after completing service. We suppose that $I - P$ is invertible, which is to say each job eventually leaves the network. One important special case allowed under this model is that of several different *types* of job, each type making transitions from one station to another according to its own $M \times M$ routing matrix. In this special case, job class is determined by both current station and type, and the type of a job never changes.

Once we have define the routing rules, the traffic equations for our multiclass network may be written down, using the same logic as in (53). These are

$$\lambda_\ell = \lambda_\ell^* + \sum_{k=1}^{n} \lambda_k p_{k\ell},\ \ell = 1, 2, \ldots, K \tag{55}$$

where the exogenous arrival rates $\lambda_\ell^*$ and the total arrival rates $\lambda_\ell$ are now specified for each class $\ell = 1, 2, \ldots, K$ rather than for each station as in (53). Note that the service distributions do not enter into this equation. Once we specify the service rates $\mu_\ell$ for each class, the class traffic intensities $\rho_\ell = \lambda_\ell / \mu_\ell$ are determined. The traffic intensity at a *station* is the sum of the traffic intensities of its classes:

$$\rho^{(m)} = \sum_{\substack{\text{all classes } \ell \text{ served}\\ \text{at station } m}} \rho_\ell,\ m = 1, 2, \ldots, M \tag{56}$$

To complete the description of a family of multiclass queueing networks, we suppose that all exogenous arrival processes are independent Poisson processes, that the queues are FIFO, and that the service distributions at each station are exponential. We write $s_m$ for the number of servers at station $m$, $m = 1, 2, \ldots, M$. This network is not a product-form network, in general. However, if we further suppose that all classes $\ell$ served at a given station $m$ have the same service rate $\mu^{(m)}$, so that $\mu_\ell = \mu^{(m)}$ for each such $\ell$ and $m$, then the network is product form. We refer to this network as the *standard multiclass product-form network*. We assume that the traffic intensity $\rho^{(m)}$ at each station $m$ is less than the number of servers:

$$\rho^{(m)} < s_m, \; m = 1, 2, \ldots, M \tag{57}$$

This condition is known as the *capacity condition*. It ensures that sufficient service capacity is available to handle the arriving work. Special cases of this condition have arisen at several earlier points in this chapter. If the station service rates $\mu^{(m)}$ and the class total arrival rates $\lambda_k$ are all positive, then (57) implies that the standard multiclass product-form network is an irreducible, positive recurrent Markov chain. Remarkably, an explicit formula for the steady-state distribution of this network is known (Kelly 1979). While the formula is complicated, it has a simple description in words.

1. The configurations (i.e., the classes and positions in queue of all the jobs) of the different stations at a fixed time in steady state are independent.
2. The number of jobs at station $m$ has the same steady-state distribution as the $M/M/s_m$ queue with traffic intensity $\rho^{(m)}$.
3. Given that there are $n$ jobs present at a particular station $m$, the number of jobs of each class has the multinomial distribution (Ross 1997) with $n$ trials and with outcome probabilities $\rho_\ell/\rho^{(m)}$ for classes $\ell$ served at station $m$.

The distribution of the overall numbers of jobs at the stations described in (1) and (2) is exactly as in the Jackson network. Formulas for steady-state waiting and flow times in this multiclass network may be written down just as we did in the case of Jackson networks—see, for example, (54).

We can generalize the standard multiclass product-form network in several ways and still get the same simple steady-state distribution. Instead of the FIFO discipline, we can use any queueing discipline that depends only on the position of the jobs in the queue (and not, for example, on a job's class or service time) and the same steady-state distribution will apply. Under certain queue disciplines, nonexponential service distributions may be allowed, and the service distribution may be allowed to depend on class. These disciplines include SIRO, PS, and preemptive LIFO, which were discussed in the introduction to Section 4. The $M/M/s$ queue under any of these disciplines becomes what is known as a *symmetric queue* (Kelly 1979), and this property underlies the possibility of relaxing the exponentiality assumption while maintaining the product-form steady-state distribution.

All of these networks have very special structure, however, either having exponential service with a rate that does not depend on class or a somewhat exotic service discipline. And yet the product-form networks constitute a broad family having a simple steady-state behavior. It seems reasonable to view this steady-state behavior as in some sense typical of a well-behaved queueing network and to use the product-form solution as a first approximation to any reasonable network.

### 6.3. General Networks: Stability and Instability

A simple example of an unstable queue is an $M/M/1$ queue in which the arrival rate $\lambda$ exceeds the service rate $\mu$. For this queue, the number in the system grows without bound. In fact, the number of jobs in the system at time $t$, $L(t)$, satisfies

$$\lim_{t\to\infty} \frac{L(t)}{t} = \lambda - \mu > 0 \tag{58}$$

so the amount of work in the system grows roughly linearly over time. This kind of behavior cannot persist for very long in real systems, as waiting jobs will ultimately overwhelm any finite storage capacity. This is a strong form of instability. In contrast, if $\lambda < \mu$ then the limit of (58) is zero and, moreover, the expected queue size at time $t$ converges to the value given in (39) as $t \to \infty$. This is typical of a stable queue. The critical case, in which $\lambda = \mu$, is more subtle; here the expected queue size at time $t$ grows like $\sqrt{t}$. This we classify as unstable also.

For Markovian networks (with a discrete state space), there is a convenient definition of stability: a stable network is one that is a positive recurrent Markov chain. This definition may be extended to more general networks, but doing this formally would require a mathematical digression. For a detailed discussion of stability, see Meyn and Tweedie (1993).

The $M/M/1$ queue with $\lambda > \mu$ fails to be stable for a very simple reason: there is insufficient service capacity to handle the load. A feature of this situation is that the traffic equations (53) actually fail. The arrival and departure rates are *not* equal, the former being $\lambda$ and the latter being the smaller value $\mu$. In the case of a multiclass queueing network, we refer to the quantities $\lambda_\ell$ and $\rho_\ell$ determined by the traffic equations (53) as the nominal arrival rate and traffic intensity for class $\ell$. Similarly, $\rho^{(m)}$

in (57) is the nominal traffic intensity for station $m$. A simple necessary condition for a queueing network to be stable, then, is that there be sufficient capacity to handle the load, which is condition (57) above. We have referred to this as the capacity condition for stability. This condition is sufficient to guarantee stability of product-form networks. However, as indicated earlier, the relative tractability of product-form networks conveys a false sense of how complex the behavior of general networks can be. Only in the late 1980s did researchers focus on the serious complications that may arise with non-product-form queueing networks. The Rybko–Stolyar (1992) network is an early example of a network exhibiting a certain key behavior, *subcritical instability*. This network is unstable in the sense that the amount of work in the system grows linearly with time, even though the capacity condition (57) is satisfied. The network is somewhat contrived in that it follows a discipline in which priority is given to slow jobs over fast jobs, contrary to the advice of Subsection 4.4 above. More recently, Bramson (1994) showed that the phenomenon of subcritical instability arises even under the FIFO discipline. While Bramson's network is again somewhat contrived, it clearly points to a need for a deeper understanding of instability. General necessary and sufficient conditions for stability of multiclass networks are not known at the present time and do not appear to be in the offing.

On the other hand, several results guaranteeing the stability of a queueing network are known. One approach to the question of stability is to initialize the network with a very large number of jobs in it and then to study its behavior over a proportionately long period of time. With this perspective, the jobs have so small an individual impact that they behave collectively like a fluid, and the network approaches a *fluid limit*, which is described by a *fluid model* reflecting the original network behavior. It has been proved in substantial generality that if a unit of fluid is emptied out of the fluid network within a fixed time, however it may be distributed among the stations, then the original network is stable. See Dai (1995) for details.

A *reentrant line* is a network in which all jobs follow the same route. The same station may be visited several times along the route—-hence the name "reentrant." These are natural models of certain semiconductor manufacturing systems. A feature of these networks is that the different classes served at a station have a natural ordering, from the earliest stage in the route ("first") to the latest stage in the route ("last"). Two common disciplines, first-buffer-first-served (FBFS) and last-buffer-first-served (LBFS), in which the buffers at a station receive priority according to how advanced they are along the common route, are known to be stable under subcritical nominal loads. See Dai and Weiss (1995) for details.

## 7. TWO-MOMENT APPROXIMATIONS AND DECOMPOSITION METHODS

Consider a queueing network, perhaps the Jackson network of Subsection 6.1 or the standard multiclass FIFO network described in Subsection 6.2. Suppose now we relax the assumption that service and external interarrival times are exponential. We may wish to do this, for example, because there are automated workstations in the system being modeled and service times at these stations are much less variable than the exponential distribution. Without exponential distributions, the explicit formulas for expected flow times and numbers-in-system of Section 6 fail, and, moreover, it is a formidable problem to devise a direct algorithm to compute these quantities. Simulation may be an effective answer to the problem. For this approach, see Subsection 1.2 and the chapters of the Handbook dealing with simulation referenced there. Here we address the question, What can be said about these more general networks, using a basic knowledge of stochastic models? In answer to this question, we present an approach to developing sensible and intuitive approximations. These approximations are built around two themes: (1) they are two-moment methods, requiring only means and variances (equivalently, SCVs) of service times and external interarrival times, and (2) they are decomposition methods, treating different stations as independent in steady state.

In support of this approach to queueing networks, there are the following points. In many situations, rough information on the mean and variability of service times and exogenous interarrival times is all we have to work with, even if the method of analysis is simulation. In a given context, it may also be intuitively clear that these quantities largely determine performance. Mathematics supports the two-moment approach to some degree. For example, the heavy-traffic mean of the single-server queue (51) is determined by the first two moments of the interarrival and service distributions. This phenomenon extends to single-class networks, whose heavy-traffic behavior depends only on two moments. See Reiman (1981) for an early result along these lines and Williams (1996) for a more recent survey. In support of decomposition as a theme for developing approximations, we note that a product-form network is a network for which decomposition is not merely an approximation but an exact mathematical property. It is natural in approximating more general networks to suppose that this decomposition property continues to hold.

On the other hand, there are many multiclass networks, for example the Rybko–Stolyar network discussed in Subsection 6.2, whose behavior is clearly not consistent with decomposition approximations, and so we must be modest in our expectations as to how far decomposition methods will take us in understanding queueing networks.

### 7.1. Introduction to Decomposition

Decomposition methods give quick, rough approximations for queueing networks using two-moment information. The effectiveness of these methods depends heavily on the nature of the network being approximated. It is not hard to find examples where any particular decomposition method will perform badly.

Decomposition is the process of approximating a system by breaking it up into parts, analyzing the parts, and then putting them back together. This general approximation strategy may be used in many diverse settings, offering fast approximate solutions to intractable computational problems. Typically, a decomposition method for queueing networks is based on two things: a method for approximating the behavior of individual stations and a method for approximating the flows of jobs between stations. The entire network is analyzed by characterizing the behavior of the individual stations in a manner that is consistent with the flows between the stations. Not only are decomposition methods fast, but they often also provide explicit approximate formulas that give insight into how variability affects performance in fairly complex systems. See Subsection 7.3 for an example. Decomposition methods are a central theme of two books on manufacturing systems, Buzacott and Shanthikumar (1993) and Gershwin (1994), and also feature in Hopp and Spearman (1996) and Compton (1997). They may be viewed as analogous to the analysis of variance (ANOVA) in statistics: a sensible elementary analysis, of great value as a benchmark, despite concerns about validity of approximations or assumptions. Just as ANOVA is valid under strict normality assumptions, the decomposition approach is exact for product-form networks.

In the next subsection we present a simple two-moment decomposition approximation scheme for queueing networks, based on the literature up to the time of publication of Whitt's 1983 paper describing the Queueing Network Analyzer (QNA). The underlying methodology is elegant and simple and makes small computational demands both in terms of CPU cycles and lines of computer code. The basic methodology is a part of many software packages for analyzing production systems, for example MPX (Suri and de Treville 1991).

### 7.2. A Simple Decomposition Method

We generally follow Whitt (1983), which contains both an elegant synthesis of early decomposition work and significant innovations. See also Nelson (1995, Subsection 8.10.2) for another simple analyzer. The network for which we develop the methodology is the *generalized Jackson network*, which is like the Jackson network of Subsection 6.1 with regard to routing, but its service times and external interarrival times are allowed to have general distributions. This network is stable under the capacity condition (57), and so we know that its behavior is not as erratic as, say, the Rybko–Stolyar network. This gives us some assurance that it is a reasonable network to attempt to approximate. The only change in the parameterization used in Subsection 6.1 is that we now must define an SCV of external interarrivals, $c_{am}^{*2}$, and of service times, $c_{sm}^2$, for each station $m$, $m = 1, 2, \ldots, M$.

We describe the behavior of a queueing network in terms of a collection of interacting point processes, or streams. These streams are the job flows of the network. All streams will be approximated by renewal processes. In dealing with these approximating renewal processes, we suppose that the only information we have about them is the mean $m$ and variance $\sigma^2$ of the interarrival times. Knowing $m$ and $\sigma^2$ is, of course, equivalent to knowing the rate $\lambda = 1/m$ and SCV $c^2$. All streams are treated as independent even if they are not.

In approximating a given stream $N$ by a renewal process, we require the approximation to have the same rate $\lambda$ as $N$. Here are the two main strategies for choosing the SCV $\tilde{c}^2$ of the approximating renewal process.

- *The stationary-interval method:* Here we choose the interarrival time SCV $\tilde{c}^2$ of the approximating renewal process to be the same as the SCV of the steady-state distribution of the interarrival times of $N$.
- *The asymptotic method:* Here we choose the interarrival-time SCV of the renewal process so that its asymptotic variance [defined in (18)] agrees with that of $N$. This means we choose $\tilde{c}^2$ so that

$$\lim_{t \to \infty} \frac{\mathrm{Var}(N(t))}{t} = \lambda \tilde{c}^2 \tag{59}$$

These two approaches typically lead to different approximations to the SCV, unless $N$ itself is renewal, because the variance of $N(t)$ is affected by the correlations between the interarrival times.

Having decided how to deal with streams, we turn to the network. The dynamics of the network are described in terms of three key operations on streams: (1) splitting, (2) superposition, and (3) queueing. Splitting happens when a stream is divided into two or more streams that are directed to different destinations. Superposition happens when two or more streams combine to form a single stream. Queueing is the process of going through a queue—viewed as an operation that transforms the arrival stream into the departure stream. We treat each of these operations on streams through simple approximations, and this leads to a natural approximation for the entire network.

The arrival rates of all streams are determined exactly through the traffic equations (53), and so we need only explain how the SCVs are handled.

### *7.2.1. Approximation for Splitting*

First consider a renewal process $N$ with rate $\lambda$ and interarrival time SCV $c^2$, and suppose a proportion $p$ of this stream is to be routed to a certain station. The routing model described in Section 6 has the following effect. For each arrival of $N$ we toss a coin with probability of heads $p$, independently of everything. We reject the points for which the coin comes up tails and keep those for which the coin comes up heads. Now the split stream is again a true renewal process—no approximation here—and its interarrival time SCV $\tilde{c}^2$ is given by

$$\tilde{c}^2 = pc^2 + 1 - p \tag{60}$$

This may be used as an approximation for the splitting operation in the case that $N$ is not renewal. The approximation is linear (more precisely, affine) in the SCV of the input stream. Even when the original stream is not renewal, as $p \to 0$ the split stream approaches a Poisson process and so its SCV approaches 1. This behavior is mimicked by the approximate SCV $\tilde{c}^2$, which also approaches 1 as $p \to 0$.

### *7.2.2. Approximation for Superposition*

Suppose we have two independent streams, $N_1$ and $N_2$, to be superposed. Denote their rates and SCVs by $\lambda_1$, $c_1^2$, $\lambda_2$, and $c_2^2$. Then, as the variance of the sum of independent things is the sum of the variances by (16), it follows that the asymptotic variance (59) of the superposition is the sum of the asymptotic variances of the individual streams. Using the asymptotic method, this leads to approximating the superposition by a renewal process with SCV $\tilde{c}^2$ determined by

$$\lambda\tilde{c}^2 = \lambda_1 c_1^2 + \lambda_2 c_2^2 \tag{61}$$

where $\lambda = \lambda_1 + \lambda_2$ is the rate of the superposition. Again, $\tilde{c}^2$ is linear in the input SCVs $c_1^2$ and $c_2^2$. This formula, extended to nonrenewal and nonindependent streams, is the approximation for superposition.

### *7.2.3. Approximation for Queueing*

A queue is viewed as transforming its arrival stream into its departure stream. A natural approximation is

$$\tilde{c}^2 = \rho^2 c_s^2 + (1 - \rho^2)c_a^2 \tag{62}$$

relating the approximation for the SCV of the departure stream, $\tilde{c}^2$, to those of the arrival stream and service time, $c_a^2$ and $c_s^2$. See Whitt (1983) for justification. Again, the approximation is linear in the input SCVs. It has the intuitively natural behavior that, as $\rho \to 1$, the departure SCV approaches the service SCV, whereas as $\rho \to 0$, the departure SCV approaches the interarrival SCV.

Rather than giving general equations to complete the specification of the analyzer, we illustrate the process for a simple example.

*Example 7.2:* We choose the network of Figure 2, treated in Example 7.1, making a single modification. We suppose that the service distribution at station 1 has SCV $c_{s1}^2 = 2$. All other distributions and parameters are as before. Writing $c_{am}^2$ and $c_{dm}^2$ for the arrival and departure SCVs at stations $m = 1$ and 2, upon applying (62) to the first station we have

$$c_{d1}^2 = \rho_{\mathit{1}}^2 c_{s1}^2 + (1 - \rho_1^2)c_{a1}^2 = 0.5 + 0.75c_{a1}^2$$

Clearly $c_{a2}^2 = c_{d1}^2$, as the arrivals to the second station are the departures from the first. and so upon applying (62) to the second station we get

$$c_{d2}^2 = \rho_2^2 c_{s2}^2 + (1 - \rho_2^2)c_{a2}^2 = 0.16 + 0.84c_{a2}^2$$

Using (60) and writing $c_{21}^2$ for the SCV of the feedback stream from station 2 to station 1, we have

$$c_{21}^2 = p_{21}c_{d2}^2 + (1 - p_{21}) = 0.5c_{d2}^2 + 0.5$$

An equation for $c_{a1}^2$ may now be written down, using (61). The result is

$$2c_{a1}^2 = (1)c_{a1}^2 + (1)c_{21}^2 = 1 + c_{21}^2$$

These equations may be solved to give $c_{a1}^2 = 1.06$ and $c_{a2}^2 = 1.30$. Substituting these values into the approximation (5.6), we find the approximate waiting times per arrival at the stations to be

$$E(W_Q^{(1)}) = \frac{0.5(1.06 + 2)}{2(4)(1 - 0.5)} = 0.38 \text{ and } E(W_Q^{(2)}) = \frac{0.4(1.30 + 1)}{2(5)(1 - 0.4)} = 0.15$$

The first of these is substantially inflated over the result of Example 7.1, mostly because of the increase in $c_{s1}^2$ itself but also partly because of the increase in $c_{a1}^2$ due to the extra variability manifesting in the feedback stream.

### 7.3. Insights from the Decomposition Approach

An excellent use of the decomposition approach is in devising simple insights into practical questions. We follow Buzacott (1996) in treating the following question using decomposition approximations. *When several tasks must be performed on each of a stream of jobs, should these tasks be carried out by separate servers in a flow-line arrangement, or should they all be carried out by parallel single-server stations?* We invoke two models to draw a conclusion. The first model consists of a flow line of $n$ buffered stations, that is, a tandem network. Each task is performed by a different server. In the second model we have $N$ parallel single-server stations where each server can process a job from start to finish. The question is to quantify the trade-offs. A difference is to be expected in the work-in-process levels, and we focus on quantifying this. Since the capacity of the flow line is constrained by its slowest station, a feature that does not arise in the parallel system, we begin by assuming that the service times for the different tasks are exactly balanced in the sense that their distributions are all the same. We suppose that, in the parallel system, arrivals are allocated at random, each job being assigned independently to one of the stations with equal probability. We may now analyze this model in a variety of ways. We can consider heavy-traffic theory, product-form network theory, and approximations, to make the comparison. Using the heavy-traffic approximation (51) and the splitting approximation (60) in conjunction with the asymptotic method of Subsection 7.1 yields the following approximations for the steady-state expected work-in-process when $\rho$ is close to 1. We have

$$\overline{L}_{\text{series}} \approx \frac{c_a^2 + (2m - 1)c_s^2}{2(1 - \rho)} \tag{63}$$

for the tandem system, whereas for the parallel system we have

$$\overline{L}_{\text{parallel}} \approx \frac{m - 1 + c_a^2 + mc_s^2}{2(1 - \rho)} \tag{64}$$

In the case of Poisson arrivals and exponential service, (7.5) is exact because in this case the network is product-form. In the case of Poisson arrivals but general service, (7.6) is exact as each station becomes an $M/G/1$ queue, for which the Pollaczek–Khintchine formula (43) holds. We see by comparing (63) and (64) that the flow line has a smaller approximate expected number-in-system if and only if $c_s^2 \leq 1/2$. This leads us to conclude that low variability favors the flow line. Thus, repetitive tasks are suited to the flow-line arrangement, whereas tasks demanding cognitive skills may be better grouped.

Had arrivals to the parallel system been assigned to stations *cyclically*, so that the first arrival goes to the first queue, the second to the second, and so forth, the conclusion would have been that the parallel system is always better. This is because of the reduced variability in the arrival processes. Buzacott (1996) gives much more on this and related issues.

**Acknowledgements**

The author thanks Sunkyo Kim and Graeme Warren for many helpful ideas and suggestions.

## REFERENCES

Asmussen, S. (1987), *Applied Probability and Queues.* John Wiley & Sons, New York.

Asmussen, S. (1992), ‘‘Queueing Simulation in Heavy Traffic,’’ *Mathematics of Operations Research,* Vol. 17, No. 1, pp. 84–111.

Bramson, M. (1994), ‘‘Instability of FIFO Queueing Networks,’’ *Annals of Applied Probability,* Vol. 4, No. 2, pp. 414–431.

Buzacott, J. A. (1996), ‘‘Commonalities in Reengineered Business Processes: Models and Issues,’’ *Management Science,* Vol. 42, No. 5, pp. 768–781.

Buzacott, J. A., and Shanthikumar, J. G. (1993), *Stochastic Models of Manufacturing Systems,* Prentice Hall, Englewood Cliffs, NJ.

Compton, W. D. (1997), *The Management of World-Class Manufacturing Enterprises.* Prentice Hall, Upper Saddle River, NJ.

Dai, J. G. (1995), ‘‘On Positive Harris Recurrence of Multiclass Queueing Networks: A Unified Approach Via Fluid Limit Models,’’ *Annals of Applied Probability,* Vol. 5, pp. 49–77.

Dai, J. G., and Weiss, G. (1996), ‘‘Stability and Instability of Fluid Models for Re-entrant Lines,’’ *Mathematics of Operations Research,* Vol. 21, pp. 115–134.

Gershwin, S. B. (1994), *Manufacturing Systems Engineering,* Prentice Hall, Englewood Cliffs, NJ.

Hall, R. W. (1991), *Queueing Methods for Services and Manufacturing,* Prentice Hall, Englewood Cliffs, NJ.

Hopp, W. J., and Spearman, M. L. (1996), *Factory Physics: The Foundations of Manufacturing Management,* Irwin, Chicago.

Jackson, J. R. (1957), ‘‘Networks of Waiting Lines,’’ *Operations Research,* Vol. 5, pp. 518–521.

Kelly, F. P. (1979), *Reversibility and Stochastic Networks,* John Wiley & Sons, New York.

Larson, R. C., and Odoni, A. R. (1981), *Urban Operations Research,* Prentice Hall, Englewood Cliffs, NJ.

Meyn, S. P., and Tweedie, R. L. (1993), *Markov Chains and Stochastic Stability,* Springer, New York.

Nelson, B. L. (1995), *Stochastic Modeling, Analysis and Simulation,* McGraw Hill, New York.

Reiman, M. I. (1984), ‘‘Open Queueing Networks in Heavy Traffic,’’ *Mathematics of Operations Research,* Vol. 9, pp. 441–458.

Ross, S. M. (1996), *Stochastic Processes,* 2nd Ed., John Wiley & Sons, New York.

Ross, S. M. (1997), *Introduction to Probability Models,* 6th Ed., Academic Press, Boston.

Rybko, A. N., and Stolyar, A. L. (1992), ‘‘Ergodicity of Stochastic Processes Describing the Operation of Open Queueing Networks,’’ *Problems of Information Transmission,* Vol. 28, pp. 199–220.

Suri, R., and de Treville, S. (1991), ‘‘Full Speed Ahead: A Timely Look at Rapid Modeling Technology in Operations Management,’’ *OR/MS Today,* June, pp. 34–42.

Whitt, W. (1989), ‘‘Planning Queueing Simulations,’’ *Management Science,* Vol. 35, pp. 1341–1366.

Whitt, W. (1983), ‘‘The Queueing Network Analyzer,’’ *Bell System Technical Journal,* Vol. 62, No. 9, pp. 2279–2815.

Williams, R. J. (1996), ‘‘On the Approximation of Queueing Networks in Heavy Traffic,’’ in *Stochastic Networks: Theory and Applications,* F. P. Kelly, S. Zachary, and I. Zeidins, Eds, Oxford University Press, Oxford, pp. 35–56.

Wolff, R. W. (1989), *Stochastic Modeling and the Theory of Queues,* Prentice Hall, Englewood Cliffs, NJ.