Guy L. Curry
Richard M. Feldman

Manufacturing Systems Modeling and Analysis

Second Edition

Springer

Manufacturing Systems Modeling and Analysis

Second Edition

Guy L. Curry · Richard M. Feldman

Manufacturing Systems Modeling and Analysis

Second Edition

Springer

Prof. Guy L. Curry
Texas A & M University
Dept. Industrial & Systems
Engineering
TAMU 3131
77843-3131 College Station
Texas
241, Zachry
USA
g-curry@tamu.edu

Richard M. Feldman
Texas A & M University
Dept. Industrial & Systems
Engineering
TAMU 3131
77843-3131 College Station
Texas
241, Zachry
USA
richf@tamu.edu

ISBN 978-3-642-16617-4 e-ISBN 978-3-642-16618-1
DOI 10.1007/978-3-642-16618-1
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2010938859

© Springer-Verlag Berlin Heidelberg 2009, 2011
This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.
The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover design: eStudio Calamar S.L.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

This book is dedicated to the two individuals who keep us going, tolerate our work ethic, and make life a wondrous journey, our wives: Jerrie Curry and Alice Feldman.

Preface

This textbook was developed to fill the need for an accessible but comprehensive presentation of the analytical approaches for modeling and analyzing models of manufacturing and production systems. It is an out growth of the efforts within the Industrial and Systems Engineering Department at Texas A&M to develop and teach an analytically based undergraduate course on probabilistic modeling of manufacturing type systems. The level of this textbook is directed at undergraduate and masters students in engineering and mathematical sciences. The only prerequisite for students using this textbook is a previous course covering calculus-based probability and statistics. The underlying methodology is queueing theory, and we shall develop the basic concepts in queueing theory in sufficient detail that the reader need not have previously covered it. Queueing theory is a well-established discipline dating back to the early 1900's work of A. K. Erlang, a Danish mathematician, on telephone traffic congestion. Although there are many textbooks on queueing theory, these texts are generally oriented to the methodological development of the field and exact results and not to the practical application of using approximations in realistic modeling situations. The application of queueing theory to manufacturing type systems started with the approximation based work of Ward Whitt in the 1980's. His paper on QNA (a queueing network analyzer) in 1983 is the base from which most applied modeling efforts have evolved.

There are several textbooks with titles similar to this book. Principle among these are: Modeling and Analysis of Manufacturing Systems by Askin and Standridge, Manufacturing Systems Engineering by Stanley Gershwin, Queueing Theory in Manufacturing Systems Analysis and Design by Papadopoulos, Heavey and Browne, Performance Analysis of Manufacturing Systems by Tayfur Altiok, Stochastic Modeling and Analysis of Manufacturing Systems, edited by David Yao, and Stochastic Models of Manufacturing Systems by Buzacott and Shanthikumar. Each of these texts, along with several others contributes greatly to the field. The book that most closely aligns with the motivation, level, and intent of this book is Factory Physics by Hopp and Spearman. Their approach and analysis is highly recommended reading, however, their book's scope is on the larger field of production and operations management. Thus, it does not provide the depth and breath of analytical modeling procedures that are presented in this text.

This text is about the development of analytical approximation models and their use in evaluating factory performance. The tools needed for the analytical approach are fully developed. One useful non-analytical tool that is not fully developed in this textbook is simulation modeling. In practice as well as in the development of the models in this text, simulation is extensively used as a verification tool. Even though the development of simulation models is only modestly addressed, we would encourage instructors who use this book in their curriculum after a simulation course to ask students to simulate some of the homework problems so that a comparison can be made of the analysis using the models presented here with simulation models. By developing simulation models students will have a better understanding of the modeling assumptions and the accuracy of the analytical approximations. In addition several chapters include an appendix that contains instructions in the use of Microsoft Excel as an aid in modeling or in building simple simulation models.

For this second edition, suggestions from various instructors who have used the textbook have been incorporated. Because of the importance of simulation modeling, this second edition also includes an introduction to event-driven simulations.

Two special sections are included to help the reader organize the many concepts contained in the text. Immediately after the Table of Contents, we have included a symbol table that contains most of the notation used throughout the text. Second, immediately after the final chapter a glossary of terms is included that summarizes the various definitions used. It is expected that these will prove valuable resources as the reader progresses through the text.

Many individuals have contributed to this book through our interactions in research efforts and discussions. Special thanks go to Professor Martin A. Wortman, Texas A&M University, who designed and taught the first presentation of the course for which this book was originally developed and Professor Bryan L. Deuermeyer, Texas A&M University, for his significant contributions to our joint research activities in this area and his continued interest and criticism. In addition several individuals have helped in improving the text by using a draft copy while teaching the material to undergraduates including Eylem Tekin at Texas A&M, Natarajan Gautam also at Texas A&M, and Kevin Gue at Auburn University. We also wish to acknowledge the contributions of Professors John A. Fowler, Arizona State University, and Mark L. Spearman, Factory Physics, Inc., for their continued interactions and discussions on modeling manufacturing systems. And we thank Ciriaco Valdez-Flores, a co-author of the first chapter covering basic probability for permission to include it as part of our book. Finally, we acknowledge our thanks through the words of the psalmist, "Give thanks to the Lord, for He is good; His love endures forever." (Psalms 107:1, NIV)

College Station, Texas *Guy L. Curry*
March 2008 *Richard M. Feldman*

Contents

[1] Section 4.3 can be omitted without affecting the continuity of the remainder of the text.

[2] Section 6.5.3 can be omitted without affecting the continuity of the remainder of the text.

Symbols

$\boldsymbol{\alpha}$ Used in Chap. 9 as the row vector of initial probabilities associated with a phase type distribution.

α_k In Chap. 9, it is used as a parameter for the GE_2 distribution that approximates the distribution of inter-arrival times into Subsystem k.

β_k In Chap. 9, it is used as a parameter for the GE_2 distribution that approximates the distribution of inter-arrival times into Subsystem k.

$\boldsymbol{\gamma}$ Vector of mean arrival rates to the various workstations from an external source.

$\boldsymbol{\gamma}^i$ Vector of mean arrival rates of Type i jobs entering the various workstations from an external source.

$\gamma_{i,k}$ Mean rate of Type i jobs into Workstation k from an external source.

$\widetilde{\gamma}^i_\ell$ Mean rate of Type i jobs to the ℓ^{th} step of the production plan from an external source (Property 6.5).

γ_k Mean rate of jobs arriving from an external source to Workstation k. In Chap. 9, it is used as a parameter for the GE_2 distribution that approximates the distribution of service times for Subsystem k.

λ Mean arrival rate.

$\boldsymbol{\lambda}$ Vector of mean arrival rates into the various workstations.

$\lambda(B)$ Mean arrival rate of batches of jobs.

λ_e The effective mean arrival rate (Def. 3.1).

$\boldsymbol{\lambda}^i$ Vector of arrival rates of Type i jobs entering the various workstations.

$\lambda(I)$ Mean arrival rate of individual jobs.

$\lambda_{i,k}$ Mean arrival rate of Type i jobs entering Workstation k.

$\widetilde{\lambda}_{i,\ell}$ Mean arrival rate of Type i jobs to the ℓ^{th} step of the production plan (Property 6.5).

λ_k Mean arrival rate into Workstation k.

μ Mean service rate (the reciprocal of the mean service time).

μ_k Often used as the mean service rate for Workstation k. In Chap. 9, it is used as a parameter for the GE_2 distribution that approximates the distribution of service times for Subsystem k.

v_i Number of steps within the production plan for a Type i job (Def. 6.3). (Not to be confused with the letter v used in Chap. 9.)

a Availability (Def. 4.2).

c_k The number of (identical) machines at Workstation k.

C^2 Squared coefficient of variation which is the variance divided by the mean squared.

C_a^2 Squared coefficient of variation of inter-arrival times.

$\mathbf{c}_a^2$ A vector of the squared coefficients of variation of the inter-arrival times to the various workstations.

$C_a^2(B)$ Squared coefficient of variation of the inter-arrival times of batches of jobs.

$C_a^2(I)$ Squared coefficient of variation of the inter-arrival times of individual jobs.

$C_a^2(k)$ Squared coefficient of variation of the stream of inter-arrival times entering Workstation k.

$C_a^2(k,j)$ Squared coefficient of variation of the inter-arrival times into Workstation j that come from Workstation k. If $k=0$, it refers to externally arriving jobs into Workstation j.

$C_d^2(k)$ The squared coefficient of variation of the inter-departure times from Workstation k.

C_s^2 Squared coefficient of variation of service times.

$C_s^2(B)$ Squared coefficient of variation of the service times of batches of jobs.

$C_s^2(I)$ Squared coefficient of variation of the service times of individual jobs.

$C_s^2(k)$ Squared coefficient of variation of service times for an arbitrary job at Workstation k.

$C_s^2(i,k)$ Squared coefficient of variation of service times for Type i jobs at Workstation k.

CT Mean cycle time (Def. 2.1).

$CT_q(k)$ Mean cycle time within the queue of Workstation k.

CT_s Mean cycle time for the system which includes all time spent within the factory.

CT_s^i Mean cycle time of a Type i job for the system which includes all time spent within the factory.

$CT(i,k)$ Mean cycle time within Workstation k for a Type i job including the time spent in the queue plus the time spent processing.

$CT(k)$ Mean cycle time within Workstation k including the time spent in the queue plus the time spent processing.

$CT_k(\cdot)$ Mean cycle time at Workstation k as a function of the CONWIP level.

E Expectation operator or the mean.

F Random variable denoting the time to failure.

G Used in Chap. 9 for a generator matrix usually associated with a GE_2 or an MGE distribution.

i A general index. Starting with Chap. 6, it is most often used to indicate a job type.

$I(\cdot,\cdot)$ An indicator function or identity matrix (Def. 6.4).

k A general index. Starting with Chap. 6, it is most often used to indicate a workstation, and in Chap. 7 it is also used for batch size.

ℓ A general index. Most often used to denote the ℓ^{th} step of a production plan. In Chap. 8, it is sometimes used to indicate job type.

m Most often used for the total number of job types.

n Most often used for the total number of workstations.

N In Chap. 7, it is a random variable denoting batch size.

$P = (p_{j,k})$ Routing matrix (Def. 5.2).

$P^i = (p^i_{j,k})$ Routing matrix of Type i jobs.

$\widetilde{P}^i = (\widetilde{p}^i_{\ell,j})$ Step-wise routing matrix for Type i jobs (Def. 6.3).

$p^F_{a,n}$ In Chap. 9, the probability that an arrival to the n^{th} (or final) subsystem, finds the subsystem full.

$p^{(i,F)}_{a,k}$ In Chap. 9, the probability that an arrival to Subsystem k, for $k < n$, finds the subsystem full and the service-machine in Phase i.

$p^0_{d,1}$ In Chap. 9, the probability that a departure from Subsystem 1 leaves the subsystem empty.

$p^{(i,0)}_{d,k}$ In Chap. 9, the probability that a departure from Subsystem k, for $k > 1$, leaves the subsystem empty and the arrival-machine in Phase i.

p_k Often used for the steady-state probability of k jobs being within a system. In Chap. 9, it is used as a parameter for the GE_2 distribution that approximates the distribution of inter-arrival times into Subsystem k.

$p_k(j,w)$ The steady-state probability that there are j jobs at Workstation k when the CONWIP level for the factory is set to w.

Q Used in Chap. 9 for a generator matrix usually associated with finding the steady-state probabilities of two-node subsystems.

q_k In Chap. 9, it is used as a parameter for the GE_2 distribution that approximates the distribution of service times for Subsystem k.

R Random variable denoting repair time, except in Chap. 7 where it is the random variable denoting the setup time for a batch.

r_k The relative arrival rate into Workstation k.

T_e Random variable denoting the effective service time (Def. 4.1).

$T_a(B)$ Random variable denoting inter-arrival times of batches of jobs.

$T_a(I)$ Random variable denoting inter-arrival times of individual jobs.

$T_s(B)$ Random variable denoting service times of batches of jobs.

$T_s(I)$ Random variable denoting service times of individual jobs.

$T_s(i,k)$ Random variable denoting service times for a Type i job in Workstation k.

$T_s(k)$ Random variable denoting service times for an arbitrary job in Workstation k.

th Mean throughput rate (Def. 2.3).

$th(k)$ Mean throughput rate for Workstation k.

u Machine utilization.

u_k Utilization factor for Workstation k (Eq. (6.2)).

$u_k(\cdot)$ Utilization factor at Workstation k as a function of the CONWIP level.

V The variance which also equals the second moment minus the mean squared.

$\mathbf{v}$ In Chap. 9, a vector of steady-state probabilities derived for a generator matrix. (Not to be confused with the Greek letter ν used in Chap. 6.)

v_i In Chap. 9, the steady-state probability of being in State i. (Not to be confused with the Greek letter ν used in Chap. 6.)

w Used in Chaps. 8 and 9 as a variable for functions whose independent variable represents work-in-process.

$\mathbf{w}$ A vector of dimension m, where m is the number of job types, giving the CONWIP limits for each job type.

$w_{\max}$ In Chaps. 8 and 9 constant indicated a maximum limit placed on work-in-process.

$\widetilde{w}^i(\cdot)$ The workstation mapping function (Def. 6.2).

WIP Mean (time-averaged) work-in-process (Def. 2.2).

$WIP_q(k)$ Mean (time-averaged) work-in-process for the queue of Workstation k.

WIP_s Mean (time-averaged) work-in-process within the system which includes all jobs within the factory.

$WIP(k)$ Mean (time-averaged) work-in-process within Workstation k including jobs in the queue and job(s) within the processor.

$WIP_k(\cdot)$ Mean (time-averaged) work-in-process at Workstation k as a function of the CONWIP level.

WL_k Workload at Workstation k (Def. 6.1 and Eq. (6.1)).

Chapter 1
Basic Probability Review

The background material for this textbook is a general understanding of probability and the properties of various distributions; thus, before discussing the modeling of the various manufacturing and production systems, it is important to review the fundamental concepts of basic probability. This material is not intended to teach probability theory, but it is used for review and to establish a common ground for the notation and definitions used throughout the book. Much of the material in this chapter is from [3], and for those already familiar with probability, this chapter can easily be skipped.

1.1 Basic Definitions

To understand probability , it is best to envision an experiment for which the outcome (result) is unknown. All possible outcomes must be defined and the collection of these outcomes is called the sample space. Probabilities are assigned to subsets of the sample space, called events. We shall give the rigorous definition for probability. However, the reader should not be discouraged if an intuitive understanding is not immediately acquired. This takes time and the best way to understand probability is by working problems.

Definition 1.1. An element of a *sample space* is an *outcome*. A set of outcomes, or equivalently a subset of the sample space, is called an *event*.

Definition 1.2. A *probability space* is a three-tuple $(\Omega, \mathscr{F}, \Pr)$ where Ω is a sample space, $\mathscr{F}$ is a collection of events from the sample space, and Pr is a probability measure that assigns a number to each event contained in $\mathscr{F}$. Furthermore, Pr must satisfy the following conditions, for each event A, B within $\mathscr{F}$:

- $\Pr(\Omega) = 1$,
- $\Pr(A) \geq 0$,
- $\Pr(A \cup B) = \Pr(A) + \Pr(B)$ if $A \cap B = \phi$, where ϕ denotes the empty set,

G.L. Curry, R.M. Feldman, *Manufacturing Systems Modeling and Analysis*, 2nd ed.,
DOI 10.1007/978-3-642-16618-1_1, © Springer-Verlag Berlin Heidelberg 2011

- $\Pr(A^c) = 1 - \Pr(A)$, where A^c is the complement of A.

It should be noted that the collection of events, $\mathscr{F}$, in the definition of a probability space must satisfy some technical mathematical conditions that are not discussed in this text. If the sample space contains a finite number of elements, then $\mathscr{F}$ usually consists of all the possible subsets of the sample space. The four conditions on the probability measure Pr should appeal to one's intuitive concept of probability. The first condition indicates that something from the sample space must happen, the second condition indicates that negative probabilities are illegal, the third condition indicates that the probability of the union of two disjoint (or mutually exclusive) events is the sum of their individual probabilities and the fourth condition indicates that the probability of an event is equal to one minus the probability of its complement (all other events). The fourth condition is actually redundant but it is listed in the definitions because of its usefulness.

A probability space is the full description of an experiment; however, it is not always necessary to work with the entire space. One possible reason for working within a restricted space is because certain facts about the experiment are already known. For example, suppose a dispatcher at a refinery has just sent a barge containing jet fuel to a terminal 800 miles down river. Personnel at the terminal would like a prediction on when the fuel will arrive. The experiment consists of all possible weather, river, and barge conditions that would affect the travel time down river. However, when the dispatcher looks outside it is raining. Thus, the original probability space can be restricted to include only rainy conditions. Probabilities thus restricted are called conditional probabilities according to the following definition.

Definition 1.3. Let $(\Omega, \mathscr{F}, \Pr)$ be a probability space where A and B are events in $\mathscr{F}$ with $\Pr(B) \neq 0$. The *conditional probability* of A given B, denoted $\Pr(A|B)$, is

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)} .$$

Venn diagrams are sometimes used to illustrate relationships among sets. In the diagram of Fig. 1.1, assume that the probability of a set is proportional to its area. Then the value of $\Pr(A|B)$ is the proportion of the area of set B that is occupied by the set $A \cap B$.

Example 1.1. A telephone manufacturing company makes radio phones and plain phones and ships them in boxes of two (same type in a box). Periodically, a quality control technician randomly selects a shipping box, records the type of phone in the box (radio or plain), and then tests the phones and records the number that were defective. The sample space is

$$\Omega = \{(r,0), (r,1), (r,2), (p,0), (p,1), (p,2)\} ,$$

where each outcome is an ordered pair; the first component indicates whether the phones in the box are the radio type or plain type and the second component gives the number of defective phones. The set $\mathscr{F}$ is the set of all subsets, namely,

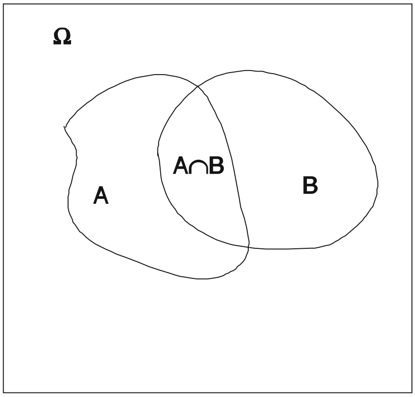


Fig. 1.1 Venn diagram illustrating events A, B, and $A \cap B$

$$\mathscr{F} = \{\phi, \{(r,0)\}, \{(r,1)\}, \{(r,0),(r,1)\}, \cdots, \Omega\} .$$

There are many legitimate probability laws that could be associated with this space. One possibility is

$$\begin{aligned} &\Pr\{(r,0)\} = 0.45 , \;\; \Pr\{(p,0)\} = 0.37 , \\ &\Pr\{(r,1)\} = 0.07 , \;\; \Pr\{(p,1)\} = 0.08 , \\ &\Pr\{(r,2)\} = 0.01 , \;\; \Pr\{(p,2)\} = 0.02 . \end{aligned}$$

By using the last property in Definition 1.2, the probability measure can be extended to all events; for example, the probability that a box is selected that contains radio phones and at most one phone is defective is given by

$$\Pr\{(r,0),(r,1)\} = 0.52 .$$

Now let us assume that a box has been selected and opened. We observe that the two phones within the box are radio phones, but no test has yet been made on whether or not the phones are defective. To determine the probability that at most one phone is defective in the box containing radio phones, define the event A to be the set $\{(r,0),(r,1),(p,0),(p,1)\}$ and the event B to be $\{(r,0),(r,1),(r,2)\}$. In other words, A is the event of having at most one defective phone, and B is the event of having a box of radio phones. The probability statement can now be written as

$$\Pr\{A|B\} = \frac{\Pr(A \cap B)}{\Pr(B)} = \frac{\Pr\{(r,0),(r,1)\}}{\Pr\{(r,0),(r,1),(r,2)\}} = \frac{0.52}{0.53} = 0.991 .$$

□

- *Suggestion: Do Problems 1.1–1.2 and 1.20.*

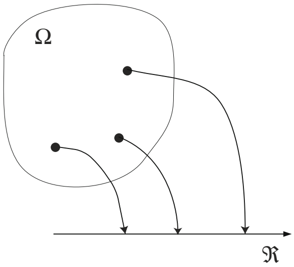


Fig. 1.2 A random variable is a mapping from the sample space to the real numbers

1.2 Random Variables and Distribution Functions

It is often cumbersome to work with the outcomes directly in mathematical terms. Random variables are defined to facilitate the use of mathematical expressions and to focus only on the outcomes of interest.

Definition 1.4. A *random variable* is a function that assigns a real number to each outcome in the sample space.

Figure 1.2 presents a schematic illustrating a random variable. The name "random variable" is actually a misnomer, since it is not random and is not a variable. As illustrated in the figure, the random variable simply maps each point (outcome) in the sample space to a number on the real line[1].

Revisiting Example 1.1, let us assume that management is primarily interested in whether or not at least one defective phone is in a shipping box. In such a case a random variable D might be defined such that it is equal to zero if all the phones within a box are good and equal to 1 otherwise; that is,

$$\begin{aligned} D(r,0) &= 0\ , \quad D(p,0) = 0\ , \\ D(r,1) &= 1\ , \quad D(p,1) = 1\ , \\ D(r,2) &= 1\ , \quad D(p,2) = 1\ . \end{aligned}$$

The set $\{D = 0\}$ refers to the set of all outcomes for which $D = 0$ and a legitimate probability statement would be

$$\Pr\{D = 0\} = \Pr\{(r,0),(p,0)\} = 0.82\ .$$

To aid in the recognition of random variables, the notational convention of using only capital Roman letters (or possibly Greek letters) for random variables is followed. Thus, if you see a lower case Roman letter, you know immediately that it can not be a random variable.

[1] Technically, the space into which the random variable maps the sample space may be more general than the real number line, but for our purposes, the real numbers will be sufficient.

Random variables are either discrete or continuous depending on their possible values. If the possible values can be counted, the random variable is called discrete; otherwise, it is called continuous. The random variable D defined in the previous example is discrete. To give an example of a continuous random variable, define T to be a random variable that represents the length of time that it takes to test the phones within a shipping box. The range of possible values for T is the set of all positive real numbers, and thus T is a continuous random variable.

A cumulative distribution function (CDF) is often used to describe the probability measure underlying the random variable. The cumulative distribution function (usually denoted by a capital Roman letter or a Greek letter) gives the probability accumulated up to and including the point at which it is evaluated.

Definition 1.5. The function F is the *cumulative distribution function* for the random variable X if

$$F(a) = \Pr\{X \le a\}$$

for all real numbers a.

The CDF for the random variable D defined above is

$$F(a) = \begin{cases} 0 & \text{for } a < 0 \\ 0.82 & \text{for } 0 \le a < 1 \\ 1.0 & \text{for } a \ge 1\,. \end{cases} \tag{1.1}$$

Figure 1.3 gives the graphical representation for F. The random variable T defined to represent the testing time for phones within a randomly chosen box is continuous and there are many possibilities for its probability measure since we have not yet defined its probability space. As an example, the function G (see Fig. 1.10) is the cumulative distribution function describing the randomness that might be associated with T:

$$G(a) = \begin{cases} 0 & \text{for } a < 0 \\ 1 - \mathrm{e}^{-2a} & \text{for } a \ge 0\,. \end{cases} \tag{1.2}$$

Property 1.1. *A cumulative distribution function F has the following properties:*

- $\lim_{a\to-\infty} F(a) = 0$,
- $\lim_{a\to+\infty} F(a) = 1$,
- $F(a) \le F(b)$ *if* $a < b$,
- $\lim_{a\to b^+} F(a) = F(b)$.

The first and second properties indicate that the graph of the cumulative distribution function always begins on the left at zero and limits to one on the right. The third property indicates that the function is nondecreasing. The fourth property indicates that the cumulative distribution function is right-continuous. Since the distribution function is monotone increasing, at each discontinuity the function value is defined

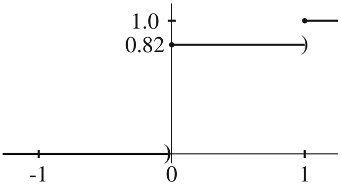


Fig. 1.3 Cumulative distribution function for Eq. (1.1) for the discrete random variable D

by the larger of two limits: the limit value approaching the point from the left and the limit value approaching the point from the right.

It is possible to describe the random nature of a discrete random variable by indicating the size of jumps in its cumulative distribution function. Such a function is called a probability mass function (denoted by a lower case letter) and gives the probability of a particular value occurring.

Definition 1.6. The function f is the *probability mass function* (pmf) of the discrete random variable X if

$$f(k) = \Pr\{X = k\}$$

for every k in the range of X.

If the pmf is known, then the cumulative distribution function is easily found by

$$\Pr\{X \le a\} = F(a) = \sum_{k \le a} f(k)\,. \tag{1.3}$$

The situation for a continuous random variable is not quite as easy because the probability that any single given point occurs must be zero. Thus, we talk about the probability of an interval occurring. With this in mind, it is clear that a mass function is inappropriate for continuous random variables; instead, a probability density function (denoted by a lower case letter) is used.

Definition 1.7. The function g is called the *probability density function* (pdf) of the continuous random variable Y if

$$\int_a^b g(u)\mathrm{d}u = \Pr\{a \le Y \le b\}$$

for all a, b in the range of Y.

From Definition 1.7 it should be seen that the pdf is the derivative of the cumulative distribution function and

$$G(a) = \int_{-\infty}^a g(u)\mathrm{d}u\,. \tag{1.4}$$

The cumulative distribution functions for the example random variables D and T are defined in Eqs. (1.1 and 1.2). We complete that example by giving the pmf for D and the pdf for T as follows:

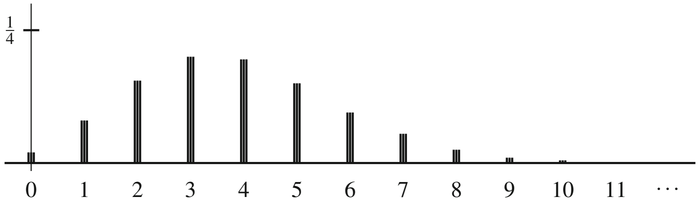


Fig. 1.4 The Poisson probability mass function of Example 1.2

$$f(k) = \begin{cases} 0.82 & \text{for } k = 0 \\ 0.18 & \text{for } k = 1 \ . \end{cases} \tag{1.5}$$

and

$$g(a) = \begin{cases} 2\mathrm{e}^{-2a} & \text{for } a \geq 0 \\ 0 & \text{otherwise} \ . \end{cases} \tag{1.6}$$

Example 1.2. Discrete random variables need not have finite ranges. A classical example of a discrete random variable with an infinite range is due to Rutherford, Chadwick, and Ellis from 1920 [7, pp. 209–210]. An experiment was performed to determine the number of α-particles emitted by a radioactive substance in 7.5 seconds. The radioactive substance was chosen to have a long half-life so that the emission rate would be constant. After 2608 experiments, it was found that the number of emissions in 7.5 seconds was a random variable, N, whose pmf could be described by

$$\Pr\{N = k\} = \frac{(3.87)^k \mathrm{e}^{-3.87}}{k!} \quad \text{for } k = 0, 1, \cdots \ .$$

It is seen that the discrete random variable N has a countably infinite range and the infinite sum of its pmf equals one. In fact, this distribution is fairly important and will be discussed later under the heading of the Poisson distribution. Figure 1.4 shows its pmf graphically. □

The notion of independence is very important when dealing with more than one random variable. Although we shall postpone the discussion on multivariate distribution functions until Sect. 1.5, we introduce the concept of independence at this point.

Definition 1.8. The random variables $X_1, \cdots, X_n$ are *independent* if

$$\Pr\{X_1 \leq x_1, \cdots, X_n \leq x_n\} = \Pr\{X_1 \leq x_1\} \times \cdots \times \Pr\{X_n \leq x_n\}$$

for all possible values of $x_1, \cdots, x_n$.

Conceptually, random variables are independent if knowledge of one (or more) random variable does not "help" in making probability statements about the other random variables. Thus, an alternative definition of independence could be made using conditional probabilities (see Definition 1.3) where the random variables X_1

and X_2 are called independent if $\Pr\{X_1 \leq x_1 | X_2 \leq x_2\} = \Pr\{X_1 \leq x_1\}$ for all values of x_1 and x_2.

For example, suppose that T is a random variable denoting the length of time it takes for a barge to travel from a refinery to a terminal 800 miles down river, and R is a random variable equal to 1 if the river condition is smooth when the barge leaves and 0 if the river condition is not smooth. After collecting data to estimate the probability laws governing T and R, we would not expect the two random variables to be independent since knowledge of the river conditions would help in determining the length of travel time.

One advantage of independence is that it is easier to obtain the distribution for sums of random variables when they are independent than when they are not independent. When the random variables are continuous, the pdf of the sum involves an integral called a *convolution*.

Property 1.2. *Let X_1 and X_2 be independent continuous random variables with pdf's given by $f_1(\cdot)$ and $f_2(\cdot)$. Let $Y = X_1 + X_2$, and let $h(\cdot)$ be the pdf for Y. The pdf for Y can be written, for all y, as*

$$h(y) = \int_{-\infty}^{\infty} f_1(y-x) f_2(x) \mathrm{d}x \,.$$

Furthermore, if X_1 and X_2 are both nonnegative random variables, then

$$h(y) = \int_{0}^{y} f_1(y-x) f_2(x) \mathrm{d}x \,.$$

Example 1.3. Our electronic equipment is highly sensitive to voltage fluctuations in the power supply so we have collected data to estimate when these fluctuations occur. After much study, it has been determined that the time between voltage spikes is a random variable with pdf given by (1.6), where the unit of time is hours. Furthermore, it has been determined that the random variables describing the time between two successive voltage spikes are independent. We have just turned the equipment on and would like to know the probability that within the next 30 minutes at least two spikes will occur.

Let X_1 denote the time interval from when the equipment is turned on until the first voltage spike occurs, and let X_2 denote the time interval from when the first spike occurs until the second occurs. The question of interest is to find $\Pr\{Y \leq 0.5\}$, where $Y = X_1 + X_2$. Let the pdf for Y be denoted by $h(\cdot)$. Property 1.2 yields

$$\begin{aligned} h(y) &= \int_0^y 4\mathrm{e}^{-2(y-x)} \mathrm{e}^{-2x} \mathrm{d}x \\ &= 4\mathrm{e}^{-2y} \int_0^y \mathrm{d}x = 4y\mathrm{e}^{-2y} \,, \end{aligned}$$

for $y \geq 0$. The pdf of Y is now used to answer our question, namely,

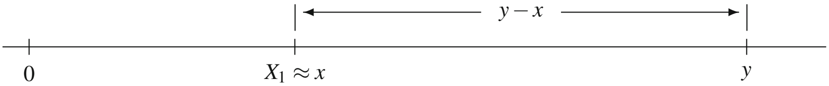


Fig. 1.5 Time line illustrating the convolution

$$\Pr\{Y \le 0.5\} = \int_0^{0.5} h(y)\mathrm{d}y = \int_0^{0.5} 4y\mathrm{e}^{-2y}\mathrm{d}y = 0.264\,.$$

□

It is also interesting to note that the convolution can be used to give the cumulative distribution function if the first pdf in the above property is replaced by the CDF; in other words, for *nonnegative* random variables we have

$$H(y) = \int_0^y F_1(y-x)f_2(x)\mathrm{d}x\,. \tag{1.7}$$

Applying (1.7) to our voltage fluctuation question yields

$$\Pr\{Y \le 0.5\} \equiv H(0.5) = \int_0^{0.5} (1-\mathrm{e}^{-2(0.5-x)})2\mathrm{e}^{-2x}\mathrm{d}x = 0.264\,.$$

We rewrite the convolution of Eq. (1.7) slightly to help in obtaining an intuitive understanding of why the convolution is used for sums. Again, assume that X_1 and X_2 are independent, nonnegative random variables with pdf's f_1 and f_2, then

$$\Pr\{X_1 + X_2 \le y\} = \int_0^y F_2(y-x)f_1(x)\mathrm{d}x\,.$$

The interpretation of $f_1(x)dx$ is that it represents the probability that the random variable X_1 falls in the interval $(x, x+dx)$ or, equivalently, that X_1 is approximately x. Now consider the time line in Fig. 1.5. For the sum to be less than y, two events must occur: first, X_1 must be some value (call it x) that is less than y; second, X_2 must be less than the remaining time that is $y-x$. The probability of the first event is approximately $f_1(x)dx$, and the probability of the second event is $F_2(y-x)$. Since the two events are independent, they are multiplied together; and since the value of x can be any number between 0 and y, the integral is from 0 to y.

- *Suggestion: Do Problems 1.3–1.6.*

1.3 Mean and Variance

Many random variables have complicated distribution functions and it is therefore difficult to obtain an intuitive understanding of the behavior of the random variable by simply knowing the distribution function. Two measures, the mean and variance, are defined to aid in describing the randomness of a random variable. The mean equals the arithmetic average of infinitely many observations of the random variable and the variance is an indication of the variability of the random variable. To illustrate this concept we use the square root of the variance which is called the *standard deviation*. In the 19*th* century, the Russian mathematician P. L. Chebyshev showed that for any given distribution, *at least* 75% of the time the observed value of a random variable will be within two standard deviations of its mean and *at least* 93.75% of the time the observed value will be within four standard deviations of the mean. These are general statements, and specific distributions will give much tighter bounds. (For example, a commonly used distribution is the normal "bell shaped" distribution. With the normal distribution, there is a 95.44% probability of being within two standard deviations of the mean.) Both the mean and variance are defined in terms of the expected value operator, that we now define.

Definition 1.9. Let h be a function defined on the real numbers and let X be a random variable. The *expected value* of $h(X)$ is given, for X discrete, by

$$E[h(X)] = \sum_k h(k) f(k)$$

where f is its pmf, and for X continuous, by

$$E[h(X)] = \int_{-\infty}^{\infty} h(s) f(s) \mathrm{d}s$$

where f is its pdf.

Example 1.4. A supplier sells eggs by the carton containing 144 eggs. There is a small probability that some eggs will be broken and he refunds money based on broken eggs. We let B be a random variable indicating the number of broken eggs per carton with a pmf given by

k	$f(k)$
0	0.779
1	0.195
2	0.024
3	0.002

.

A carton sells for $4.00, but a refund of 5 cents is made for each broken egg. To determine the expected income per carton, we define the function h as follows

k	$h(k)$
0	4.00
1	3.95
2	3.90
3	3.85

.

Thus, $h(k)$ is the net revenue obtained when a carton is sold containing k broken eggs. Since it is not known ahead of time how many eggs are broken, we are interested in determining the *expected* net revenue for a carton of eggs. Definition 1.9 yields

$$\begin{aligned} E[h(B)] = 4.00 \times 0.779 + 3.95 \times 0.195 \\ + 3.90 \times 0.024 + 3.85 \times 0.002 = 3.98755 \,. \end{aligned}$$

□

The expected value operator is a linear operator, and it is not difficult to show the following property.

Property 1.3. *Let X and Y be two random variables with c being a constant, then*

- $E[c] = c$,
- $E[cX] = cE[X]$,
- $E[X+Y] = E[X] + E[Y]$.

In the egg example since the cost per broken egg is a constant $(c = 0.05)$, the expected revenue per carton could be computed as

$$\begin{aligned} E[4.0 - 0.05B] &= 4.0 - 0.05E[B] \\ &= 4.0 - 0.05 \; (\; 0 \times 0.779 + 1 \times 0.195 + 2 \times 0.024 + 3 \times 0.002 \;) \\ &= 3.98755 \,. \end{aligned}$$

The expected value operator provides us with the procedure to determine the mean and variance.

Definition 1.10. The *mean*, μ or $E[X]$, and *variance*,σ^2 or $V[X]$, of a random variable X are defined as

$$\mu = E[X], \quad \sigma^2 = E[(X-\mu)^2] \,,$$

respectively. The *standard deviation* is the square root of the variance.

Property 1.4. *The following are often helpful as computational aids:*

- $V[X] = \sigma^2 = E[X^2] - \mu^2$
- $V[cX] = c^2 V[X]$
- *If* $X \geq 0$, $E[X] = \int_0^\infty [1 - F(s)]\mathrm{d}s$ *where* $F(x) = \Pr\{X \leq x\}$
- *If* $X \geq 0$, *then* $E[X^2] = 2\int_0^\infty s[1 - F(s)]\mathrm{d}s$ *where* $F(x) = \Pr\{X \leq x\}$.

Example 1.5. The mean and variance calculations for a discrete random variable can be easily illustrated by defining the random variable N to be the number of defective phones within a randomly chosen box from Example 1.1. In other words, N has the pmf given by

$$\Pr\{N = k\} = \begin{cases} 0.82 & \text{for } k = 0 \\ 0.15 & \text{for } k = 1 \\ 0.03 & \text{for } k = 2\,. \end{cases}$$

The mean and variance is, therefore, given by

$$\begin{aligned} E[N] &= 0 \times 0.82 + 1 \times 0.15 + 2 \times 0.03 \\ &= 0.21, \end{aligned}$$

$$\begin{aligned} V[N] &= (0 - 0.21)^2 \times 0.82 + (1 - 0.21)^2 \times 0.15 + (2 - 0.21)^2 \times 0.03 \\ &= 0.2259\,. \end{aligned}$$

Or, an easier calculation for the variance (Property 1.4) is

$$\begin{aligned} E[N^2] &= 0^2 \times 0.82 + 1^2 \times 0.15 + 2^2 \times 0.03 \\ &= 0.27 \end{aligned}$$

$$\begin{aligned} V[N] &= 0.27 - 0.21^2 \\ &= 0.2259\,. \end{aligned}$$

□

Example 1.6. The mean and variance calculations for a continuous random variable can be illustrated with the random variable T whose pdf was given by Eq. 1.6. The mean and variance is therefore given by

$$E[T] = \int_0^\infty 2s\mathrm{e}^{-2s}\mathrm{d}s = 0.5\,,$$

$$V[T] = \int_0^\infty 2(s - 0.5)^2\mathrm{e}^{-2s}\mathrm{d}s = 0.25\,.$$

Or, an easier calculation for the variance (Property 1.4) is

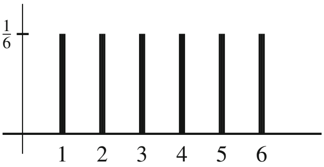


Fig. 1.6 A discrete uniform probability mass function

$$E[T^2] = \int_0^\infty 2s^2 e^{-2s} \mathrm{d}s = 0.5 \,,$$

$$V[T] = 0.5 - 0.5^2 = 0.25 \,.$$

□

The final definition in this section is used often as a descriptive statistic to give an intuitive feel for the variability of processes.

Definition 1.11. The *squared coefficient of variation*, C^2, of a nonnegative random variable T is the ratio of the the variance to the mean squared; that is,

$$C^2[T] = \frac{V[T]}{E[T]^2} \,.$$

- *Suggestion: Do Problems 1.7–1.14.*

1.4 Important Distributions

There are many distribution functions that are used so frequently that they have become known by special names. In this section, some of the major distribution functions are given. The student will find it helpful in years to come if these distributions are committed to memory. There are several textbooks (my favorite is [6, chap. 6]) that give more complete descriptions of distributions, and we recommend gaining a familiarity with a variety of distribution functions before any serious modeling is attempted.

Uniform-Discrete: The random variable N has a discrete uniform distribution if there are two integers a and b such that the pmf of N can be written as

$$f(k) = \frac{1}{b-a+1} \quad \text{for } k = a, a+1, \cdots, b \,. \tag{1.8}$$

Then,

$$E[N] = \frac{a+b}{2}; \quad V[N] = \frac{(b-a+1)^2 - 1}{12} \,.$$

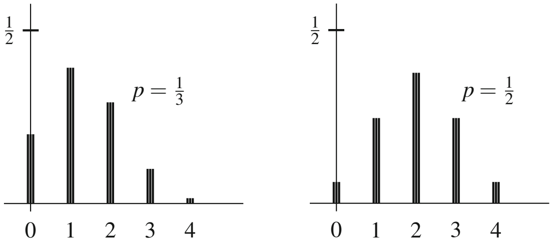


Fig. 1.7 Two binomial probability mass functions

Example 1.7. Consider rolling a fair die. Figure 1.6 shows the uniform pmf for the "number of dots" random variable. Notice in the figure that, as the name "uniform" implies, all the probabilities are the same. □

Bernoulli: The random variable N has a Bernoulli distribution if there is a number $0 < p < 1$ such that the pmf of N can be written as

$$f(k) = \begin{cases} 1-p & \text{for } k = 0 \\ p & \text{for } k = 1\,. \end{cases} \tag{1.9}$$

Then,

$$E[N] = p; \quad V[N] = p(1-p); \quad C^2[N] = \frac{1-p}{p}\,.$$

Binomial: (By James Bernoulli, 1654-1705; published posthumously in 1713.) The random variable N has a binomial distribution if there is a number $0 < p < 1$ and a positive integer n such that the pmf of N can be written as

$$f(k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} \ \text{ for } k = 0, 1, \cdots, n\,. \tag{1.10}$$

Then,

$$E[N] = np; \quad V[N] = np(1-p); \quad C^2[N] = \frac{1-p}{np}\,.$$

The number p is often though of as the probability of a success. The binomial pmf evaluated at k thus gives the probability of k successes occurring out of n trials. The binomial random variable with parameters p and n is the sum of n (independent) Bernoulli random variables each with parameter p.

Example 1.8. We are monitoring calls at a switchboard in a large manufacturing firm and have determined that one third of the calls are long distance and two thirds of the calls are local. We have decided to pick four calls at random and would like to know how many calls in the group of four are long distance. In other words, let N be a random variable indicating the number of long distance calls in the group of four. Thus, N is binomial with $n = 4$ and $p = 1/3$. It also happens that in this company, half of the individuals placing calls are women and half are men. We would also like to know how many of the group of four were calls placed by men. Let M denote

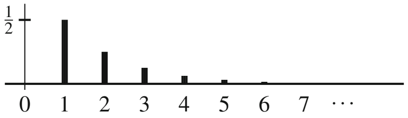


Fig. 1.8 A geometric probability mass function

the number of men placing calls; thus, M is binomial with $n = 4$ and $p = 1/2$. The pmf's for these two random variables are shown in Fig. 1.7. Notice that for $p = 0.5$, the pmf is symmetric, and as p varies from 0.5, the graph becomes skewed. □

Geometric: The random variable N has a geometric distribution if there is a number $0 < p < 1$ such that the pmf of N can be written as

$$f(k) = p(1-p)^{k-1} \text{ for } k = 1, 2, \cdots . \tag{1.11}$$

Then,

$$E[N] = \frac{1}{p}; \quad V[N] = \frac{1-p}{p^2}; \quad C^2[N] = 1-p.$$

The idea behind the geometric random variable is that it represents the number of trials until the first success occurs. In other words, p is thought of as the probability of success for a single trial, and we continually perform the trials until a success occurs. The random variable N is then set equal to the number of trial that we had to perform. Note that although the geometric random variable is discrete, its range is infinite.

Example 1.9. A car saleswoman has made a statistical analysis of her previous sales history and determined that each day there is a 50% chance that she will sell a luxury car. After careful further analysis, it is also clear that a luxury car sale on one day is independent of the sale (or lack of it) on another day. On New Year's Day (a holiday in which the dealership was closed) the saleswoman is contemplating when she will sell her first luxury car of the year. If N is the random variable indicating the day of the first luxury car sale ($N = 1$ implies the sale was on January 2), then N is distributed according to the geometric distribution with $p = 0.5$, and its pmf is shown in Fig. 1.8. Notice that theoretically the random variable has an infinite range, but for all practical purposes the probability of the random variable being larger than seven is negligible. □

Poisson: (By Simeon Denis Poisson, 1781-1840; published in 1837.) The random variable N has a Poisson distribution if there is a number $\lambda > 0$ such that the pmf of N can be written as

$$f(k) = \frac{\lambda^k e^{-\lambda}}{k!} \text{ for } k = 0, 1, \cdots . \tag{1.12}$$

Then,

$$E[N] = \lambda; \quad V[N] = \lambda; \quad C^2[N] = 1/\lambda .$$

The Poisson distribution is the most important discrete distribution in stochastic modeling. It arises in many different circumstances. One use is as an approximation to the binomial distribution. For n large and p small, the binomial is approximated by the Poisson by setting $\lambda = np$. For example, suppose we have a box of 144 eggs and there is a 1% probability that any one egg will break. Assuming that the breakage of eggs is independent of other eggs breaking, the probability that exactly 3 eggs will be broken out of the 144 can be determined using the binomial distribution with $n = 144$, $p = 0.01$, and $k = 3$; thus

$$\frac{144!}{141!3!}(0.01)^3(0.99)^{141} = 0.1181\ ,$$

or by the Poisson approximation with $\lambda = 1.44$ that yields

$$\frac{(1.44)^3 \mathrm{e}^{-1.44}}{3!} = 0.1179\ .$$

In 1898, L. V. Bortkiewicz [7, p. 206] reported that the number of deaths due to horse-kicks in the Prussian army was a Poisson random variable. Although this seems like a silly example, it is very instructive. The reason that the Poisson distribution holds in this case is due to the binomial approximation feature of the Poisson. Consider the situation: there would be a small chance of death by horse-kick for any one person (i.e., p small) but a large number of individuals in the army (i.e., n large). There are many analogous situations in modeling that deal with large populations and a small chance of occurrence for any one individual within the population. In particular, arrival processes (like arrivals to a bus station in a large city) can often be viewed in this fashion and thus described by a Poisson distribution. Another common use of the Poisson distribution is in population studies. The population size of a randomly growing organism often can be described by a Poisson random variable. W. S. Gosset, using the pseudonym of Student, showed in 1907 that the number of yeast cells in 400 squares of haemocytometer followed a Poisson distribution. Radioactive emissions are also Poisson as indicated in Example 1.2. (Fig. 1.4 also shows the Poisson pmf.)

Many arrival processes are well approximated using the Poisson probabilities. For example, the number of arriving telephone calls to a switchboard during a specified period of time, or the number of arrivals to a teller at a bank during a fixed period of time are often modeled as a Poisson random variable. Specifically, we say that an arrival process is a Poisson process with mean rate λ if arrivals occur one-at-a-time and the number of arrivals during an interval of length t is given by the random variable N_t where

$$\Pr\{N_t = k\} = \frac{(\lambda t)^k \mathrm{e}^{-\lambda t}}{k!} \quad \text{for } k = 0, 1, \cdots\ . \tag{1.13}$$

Uniform-Continuous: The random variable X has a continuous uniform distribution if there are two numbers a and b with $a < b$ such that the pdf of X can be written as

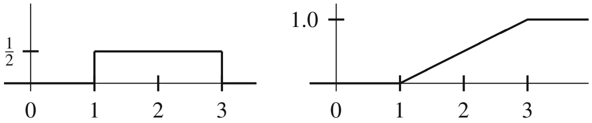


Fig. 1.9 The probability density function and cumulative distribution function for a continuous uniform distribution between 1 and 3

$$f(s) = \begin{cases} \frac{1}{b-a} & \text{for } a \le s \le b \\ 0 & \text{otherwise .} \end{cases} \tag{1.14}$$

Then its cumulative probability distribution is given by

$$F(s) = \begin{cases} 0 & \text{for } s < a \\ \frac{s-a}{b-a} & \text{for } a \le s < b \\ 1 & \text{for } s \ge b \text{ ,} \end{cases}$$

and

$$E[X] = \frac{a+b}{2}; \;\; V[X] = \frac{(b-a)^2}{12}; \;\; C^2[X] = \frac{(b-a)^2}{3(a+b)^2} \; .$$

The graphs for the pdf and CDF of the continuous uniform random variables are the simplest of the continuous distributions. As shown in Fig. 1.9, the pdf is a rectangle and the CDF is a “ramp” function.

Exponential: The random variable X has an exponential distribution if there is a number $\lambda > 0$ such that the pdf of X can be written as

$$f(s) = \begin{cases} \lambda \mathrm{e}^{-\lambda s} & \text{for } s \ge 0 \\ 0 & \text{otherwise .} \end{cases} \tag{1.15}$$

Then its cumulative probability distribution is given by

$$F(s) = \begin{cases} 0 & \text{for } s < 0, \\ 1 - \mathrm{e}^{-\lambda s} & \text{for } s \ge 0; \end{cases}$$

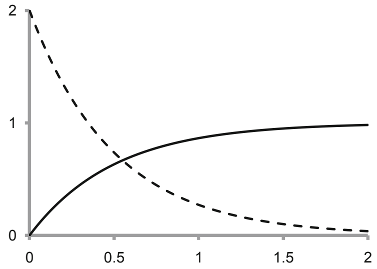
and

$$E[X] = \frac{1}{\lambda}; \;\; V[X] = \frac{1}{\lambda^2}; \;\; C^2[X] = 1 \; .$$

The exponential distribution is an extremely common distribution in probabilistic modeling. One very important feature is that the exponential distribution is the only continuous distribution that contains no memory. Specifically, an exponential random variable X is said to be memoryless if

$$\Pr\{X > t + s | X > t\} = \Pr\{X > s\} \; . \tag{1.16}$$

Fig. 1.10 Exponential CDF (solid line) and pdf (dashed line) with a mean of $\frac{1}{2}$



That is if, for example, a machine's failure time is due to purely random events (like voltage surges through a power line), then the exponential random variable would properly describe the failure time. However, if failure is due to the wear out of machine parts, then the exponential distribution would not be suitable (see Problem 1.24).

As a result of this lack of memory, a very important characteristic is that if the number of events within an interval of time are according to a Poisson random variable, then the time between events is exponential (and vice versa). Specifically, if an arrival process is a Poisson process (Eq. 1.13) with mean rate λ, the times between arrivals are governed by an exponential distribution with mean $1/\lambda$. Furthermore, if an arrival process is such that the times between arrivals are exponentially distributed with mean $1/\lambda$, the number of arrivals in an interval of length t is a Poisson random variable with mean λt.

Example 1.10. A software company has received complaints regarding their responsiveness for customer service. They have decided to analyze the arrival pattern of phone calls to customer service and have determined that the arrivals form a Poisson process with a mean of 120 calls per hour. Since a characteristic of a Poisson process is exponentially distributed inter-arrival times, we know that the distribution of the time between calls is exponentially distributed with a mean of 0.5 minutes. Thus, the graphs of the pdf and CDF describing the randomness of inter-arrival times are shown in Fig. 1.10. □

Erlang: (Named after the Danish mathematician A. K. Erlang for his extensive use of it and his pioneering work in queueing theory in the early 1900's.) The nonnegative random variable X has an Erlang distribution if there is a positive integer k and a positive number β such that the pdf of X can be written as

$$f(s) = \frac{k(ks)^{k-1}\mathrm{e}^{-(k/\beta)s}}{\beta^k\,(k-1)!} \quad \text{for } s \geq 0\,. \tag{1.17}$$

Then,

$$E[X] = \beta;\ \ V[X] = \frac{\beta^2}{k};\ \ C^2[X] = \frac{1}{k}\,.$$

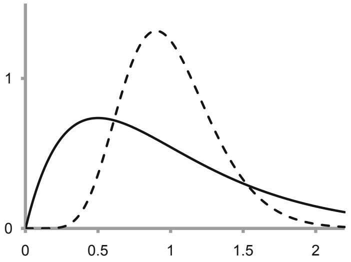


Fig. 1.11 Two Erlang probability density functions with mean 1 and shape parameters $k = 2$ (solid line) and $k = 10$ (dashed line)

Note that the constant β is often called the scale factor because changing its value is equivalent to either stretching or compressing the x-axis, and the constant k is often called the shape parameter because changing its value changes the shape of the pdf.

The usefulness of the Erlang is due to the fact that an Erlang random variable with parameters k and β is the sum of k (independent) exponential random variables each with mean β/k. In modeling process times, the exponential distribution is often inappropriate because the standard deviation is as large as the mean. Engineers usually try to design systems that yield a standard deviation of process times significantly smaller than their mean. Notice that for the Erlang distribution, the standard deviation decreases as the square root of the parameter k increases so that processing times with a small standard deviation can often be approximated by an Erlang random variable.

Figure 1.11 illustrates the effect of the parameter k by graphing the pdf for a type-2 Erlang and a type-10 Erlang. (The parameter k establishes the "type" for the Erlang distribution.) Notice that a type-1 Erlang is an exponential random variable so its pdf would have the form shown in Fig. 1.10.

Gamma: The Erlang distribution is part of a larger class of nonnegative random variables called gamma random variables. It is a common distribution used to describe process times and has two parameters: a shape parameter, α, and a scale parameter, β. A shape parameter is so named because varying its value results in different shapes for the pdf. Varying the scale parameter does not change the shape of the distribution, but it tends to ”stretch” or ”compress” the x-axis. Before giving the density function for the gamma, we must define the *gamma function* because it is used in the definition of the gamma distribution. The gamma function is defined, for $x > 0$, as

$$\Gamma(x) = \int_0^\infty s^{x-1} \mathrm{e}^{-s} \mathrm{d}s \,. \tag{1.18}$$

One useful property of the gamma function is the relationship $\Gamma(x+1) = x\Gamma(x)$, for $x \geq 1$. Thus, if x is a positive integer, $\Gamma(x) = (x-1)!$. (For some computational issues, see the appendix to this chapter.) The density function for a gamma random variable is given by

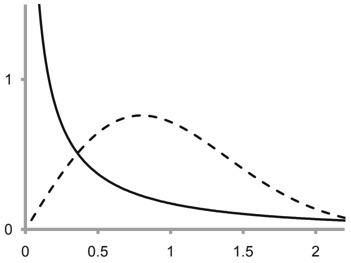


Fig. 1.12 Two Weibull probability density functions with mean 1 and shape parameters $\alpha = 0.5$ (solid line) and $\alpha = 2$ (dashed line)

$$f(s) = \frac{s^{\alpha-1} e^{-s/\beta}}{\beta^\alpha \Gamma(\alpha)} \text{ for } s \geq 0 \,. \tag{1.19}$$

Then,

$$E[X] = \beta\alpha; \;\; V[X] = \beta^2\alpha; \;\; C^2[X] = \frac{1}{\alpha} \,.$$

Notice that if it is desired to determine the shape and scale parameters for a gamma distribution with a known mean and variance, the inverse relationships are

$$\alpha = \frac{E[X]^2}{V[X]} \text{ and } \beta = \frac{E[X]}{\alpha} \,.$$

Weibull: In 1939, W. Weibull [2, p. 73] developed a distribution for describing the breaking strength of various materials. Since that time, many statisticians have shown that the Weibull distribution can often be used to describe failure times for many different types of systems. The Weibull distribution has two parameters: a scale parameter, β, and a shape parameter, α. Its cumulative distribution function is given by

$$F(s) = \begin{cases} 0 & \text{for } s < 0 \\ 1 - e^{-(s/\beta)^\alpha} & \text{for } s \geq 0 \,. \end{cases}$$

Both scale and shape parameters can be any positive number. As with the gamma distribution, the shape parameter determines the general shape of the pdf (see Fig. 1.12) and the scale parameter either expands or contracts the pdf. The moments of the Weibull are a little difficult to express because they involve the gamma function (1.18). Specifically, the moments for the Weibull distribution are

$$E[X] = \beta\Gamma(1 + \frac{1}{\alpha}); \quad E[X^2] = \beta^2\Gamma(1 + \frac{2}{\alpha}); \quad E[X^3] = \beta^3\Gamma(1 + \frac{3}{\alpha}) \,. \tag{1.20}$$

It is more difficult to determine the shape and scale parameters for a Weibull distribution with a known mean and variance, than it is for the gamma distribution because the gamma function must be evaluated to determine the moments of a Weibull. Some computational issues for obtaining the shape and scale parameters of a Weibull are discussed in the appendix to this chapter.

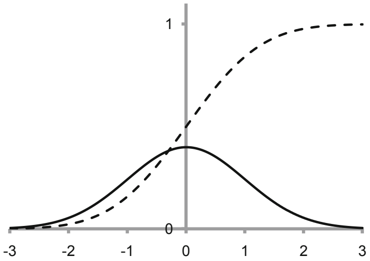


Fig. 1.13 Standard normal pdf (solid line) and CDF (dashed line)

When the shape parameter is greater than 1, the shape of the Weibull pdf is unimodal similar to the Erlang with its type parameter greater than 1. When the shape parameter equals 1, the Weibull pdf is an exponential pdf. When the shape parameter is less than 1, the pdf is similar to the exponential except that the graph is asymptotic to the y-axis instead of hitting the y-axis. Figure 1.12 provides an illustration of the effect that the shape parameter has on the Weibull distribution. Because the mean values were held constant for the two pdf's shown in the figure, the value for β varied. The pdf plotted with a solid line in the figure has $\beta = 0.5$ that, together with $\alpha = 0.5$, yields a mean of 1 and a standard deviation of 2.236; the dashed line is pdf that has $\beta = 1.128$ that, together with $\alpha = 2$, yields a mean of 1 and a standard deviation 0.523.

Normal: (Discovered by A. de Moivre, 1667-1754, but usually attributed to Karl Gauss, 1777-1855.) The random variable X has a normal distribution if there are two numbers μ and σ with $\sigma > 0$ such that the pdf of X can be written as

$$f(s) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-(s-\mu)^2/(2\sigma^2)} \text{ for } -\infty < s < \infty \,. \tag{1.21}$$

Then,

$$E[X] = \mu; \;\; V[X] = \sigma^2; \;\; C^2[X] = \frac{\sigma^2}{\mu^2} \,.$$

The normal distribution is the most common distribution recognized by most people by its "bell shaped" curve. Its pdf and CDF are shown in Fig. 1.13 for a normally distributed random variable with mean zero and standard deviation one.

Although the normal distribution is not widely used in stochastic modeling, it is, without question, the most important distribution in statistics. The normal distribution can be used to approximate both the binomial and Poisson distributions. A common rule-of-thumb is to approximate the binomial whenever n (the number of trials) is larger than 30. If $np < 5$, then use the Poisson for the approximation with $\lambda = np$. If $np \geq 5$, then use the normal for the approximation with $\mu = np$ and $\sigma^2 = np(1-p)$. Furthermore, the normal can be used to approximate the Poisson whenever $\lambda > 30$. When using a continuous distribution (like the normal) to approximate a discrete distribution (like the Poisson or binomial), the interval between the discrete values is usually split halfway. For example, if we desire to approximate the probability that a Poisson random variable will take on the values 29, 30, or 31 with a continuous distribution, then we would determine the probability that the continuous random variable is between 28.5 and 31.5.

Example 1.11. The software company mentioned in the previous example has determined that the arrival process is Poisson with a mean arrival rate of 120 per hour. The company would like to know the probability that in any one hour 140 or more calls arrive. To determine that probability, let N be a Poisson random variable with $\lambda = 120$, let X be a random variable with $\mu = \sigma^2 = 120$ and let Z be a standard normal random variable (i.e., Z is normal with mean 0 and variance 1). The above question is answered as follows:

$$\begin{aligned}\Pr\{N \geq 140\} &\approx \Pr\{X > 139.5\} \\ &= \Pr\{Z > (139.5 - 120)/10.95\} \\ &= \Pr\{Z > 1.78\} = 1 - 0.9625 = 0.0375\,.\end{aligned}$$

□

The importance of the normal distribution is due to its property that sample means from almost any practical distribution will limit to the normal; this property is called the *Central Limit Theorem*. We state this property now even though it needs the concept of statistical independence that is not yet defined. However, because the idea should be somewhat intuitive, we state the property at this point since it is so central to the use of the normal distribution.

Property 1.5. Central Limit Theorem. *Let $\{X_1, X_2, \cdots, X_n\}$ be a sequence of n independent random variables each having the same distribution with mean μ and (finite) variance σ^2, and define*

$$\overline{X} = \frac{X_1 + X_2 + \cdots + X_n}{n}\,.$$

Then, the distribution of the random variable Z defined by

$$Z = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}}$$

approaches a normal distribution with zero mean and standard deviation of one as n gets large.

Log Normal: The final distribution that we briefly mention is based on the normal distribution. Specifically, if X is a normal random variable with mean μ_N and variance σ_N^2, the random variable $Y = \mathrm{e}^X$ is called a log-normal random variable with mean μ_L and variance σ_L^2. (Notice that the name arrises because the random variable defined by the natural log of Y; namely $\ln(Y)$, is normally distributed.) This distribution is always non-negative and can have a relatively large right-hand tail. It is often used for modeling repair times and also for modeling many biological characteristics. It is not difficult to obtain the mean and variance of the log-normal distribution from the characteristics of the normal:

$$\mu_L = \mathrm{e}^{\mu_N + \frac{1}{2}\sigma_N^2}\ , \quad \text{and } \sigma_L^2 = \mu_L^2 \times (\mathrm{e}^{\sigma_N^2} - 1)\ . \tag{1.22}$$

Because the distribution is skewed to the right (long right-hand tail), the mean is to the right of the mode which is given by $\mathrm{e}^{\mu_N - \sigma_N^2}$. If the mean and variance of the log-normal distribution is known, it is straight forward to obtain the characteristics of the normal random variable that generates the log-normal, specifically

$$\sigma_N^2 = \ln(\,c_L^2 + 1\,)\ , \quad \text{and } \mu_N = \ln(\,\mu_L\,) - \frac{1}{2}\,\sigma_N^2\ , \tag{1.23}$$

where the squared coefficient of variation is given by $c_L^2 = \sigma_L^2/\mu_L^2$.

Skewness: Before moving to the discussion of more than one random variable, we mention an additional descriptor of distributions. The first moment gives the central tendency for random variables, and the second moment is used to measure variability. The third moment, that was not discussed previously, is useful as a measure of skewness (i.e., non-symmetry). Specifically, the coefficient of skewness, ν, for a random variable T with mean μ and standard deviation σ is defined by

$$\nu = \frac{E[(T-\mu)^3]}{\sigma^3}\ , \tag{1.24}$$

and the relation to the other moments is

$$E[(T-\mu)^3] = E[T^3] - 3\mu E[T^2] + 2\mu^3\ .$$

A symmetric distribution has $\nu = 0$; if the mean is to the left of the mode, $\nu < 0$ and the left-hand side of the distribution will have the longer tail; if the mean is to the right of the mode, $\nu > 0$ and the right-hand side of the distribution will have the longer tail. For example, $\nu = 0$ for the normal distribution, $\nu = 2$ for the exponential distribution, $\nu = 2/\sqrt{k}$ for a type-k Erlang distribution, and for a gamma distribution, we have $\nu = 2/\sqrt{\alpha}$. The Weibull pdf's shown in Fig. 1.12 have skewness coefficients of 3.9 and 0.63, respectively, for the solid line figure and dashed line graphs. Thus, the value of ν can help complete the intuitive understanding of a particular distribution.

- *Suggestion: Do Problems 1.15–1.19.*

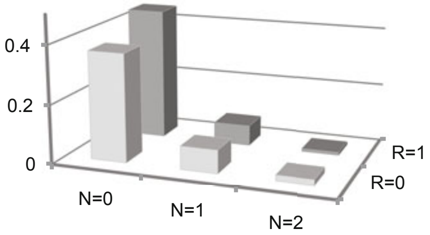


Fig. 1.14 Probability mass function for the two discrete random variables from Example 1.12

1.5 Multivariate Distributions

The analysis of physical phenomena usually involves many distinct random variables. In this section we discuss the concepts involved when two random variables are defined. The extension to more than two is left to the imagination of the reader and the numerous textbooks that have been written on the subject.

Definition 1.12. The function F is called the *joint cumulative distribution function* for X_1 and X_2 if

$$F(a,b) = \Pr\{X_1 \le a, X_2 \le b\}$$

for a and b any two real numbers.

In a probability statement as in the right-hand-side of the above equation, the comma means intersection of events and is read as "The probability that X_1 is less than or equal to *a and* X_2 is less than or equal to b". The initial understanding of joint probabilities is easiest with discrete random variables.

Definition 1.13. The function f is a *joint pmf* for the discrete random variables X_1 and X_2 if

$$f(a,b) = \Pr\{X_1 = a, X_2 = b\}$$

for each (a,b) in the range of (X_1,X_2).

For the single-variable pmf, the height of the pmf at a specific value gives the probability that the random variable will equal that value. It is the same for the joint pmf except that the graph is in three-dimensions. Thus, the height of the pmf evaluated at a specified ordered pair gives the probability that the random variables will equal those specified values (Fig. 1.14).

It is sometimes necessary to obtain from the joint pmf the probability of one random variable without regard to the value of the second random variable.

Definition 1.14. The *marginal pmf* for X_1 and X_2, denoted by f_1 and f_2, respectively, are

$$f_1(a) = \Pr\{X_1 = a\} = \sum_k f(a,k)$$

for a in the range of X_1, and

$$f_2(b) = \Pr\{X_2 = b\} = \sum_k f(k,b)$$

for b in the range of X_2.

Example 1.12. We return again to Example 1.1 to illustrate these concepts. The random variable R will indicate whether a randomly chosen box contains radio phones or plain phones; namely, if the box contains radio phones then we set $R = 1$ and if plain phones then $R = 0$. Also the random variable N will denote the number of defective phones in the box. Thus, according to the probabilities defined in Example 1.1, the joint pmf,

$$f(a,b) = \Pr\{R = a, N = b\} ,$$

has the probabilities as listed in Table 1.1. By summing in the "margins", we obtain the marginal pmf for R and N separately as shown in Table 1.2. Thus we see, for example, that the probability of choosing a box with radio phones (i.e., $\Pr\{R = 1\}$) is 53%, the probability of choosing a box of radio phones that has one defective phone (i.e., $\Pr\{R = 1, N = 1\}$) is 7%, and the probability that both phones in a randomly chosen box (i.e., $\Pr\{N = 2\}$) are defective is 3%. □

Table 1.1 Joint probability mass function of Example 1.12

	$N=0$	$N=1$	$N=2$
$R=0$	0.37	0.08	0.02
$R=1$	0.45	0.07	0.01

Table 1.2 Marginal probability mass functions of Example 1.12

	$N=0$	$N=1$	$N=2$	$f_1(\cdot)$
$R=0$	0.37	0.08	0.02	0.47
$R=1$	0.45	0.07	0.01	0.53
$f_2(\cdot)$	0.82	0.15	0.03	

Continuous random variables are treated in an analogous manner to the discrete case. The major difference in moving from one continuous random variable to two is that probabilities are given in terms of a volume under a surface instead of an area under a curve (see Fig. 1.15 for representation of a joint pdf).

Definition 1.15. The functions g, g_1, and g_2 are the *joint pdf* for X_1 and X_2, the *marginal pdf* for X_1, and the *marginal pdf* for X_2, respectively, as the following

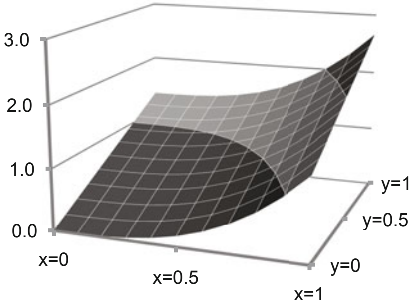


Fig. 1.15 Probability density function for the two continuous random variables from Example 1.13

hold:

$$\Pr\{a_1 \le X_1 \le b_1, a_2 \le X_2 \le b_2\} = \int_{a_2}^{b_2} \int_{a_1}^{b_1} g(s_1, s_2)\mathrm{d}s_1 \mathrm{d}s_2$$
$$g_1(a) = \int_{-\infty}^{\infty} g(a, s)\mathrm{d}s$$
$$g_2(b) = \int_{-\infty}^{\infty} g(s, b)\mathrm{d}s\,,$$

where

$$\Pr\{a \le X_1 \le b\} = \int_a^b g_1(s)\mathrm{d}s$$
$$\Pr\{a \le X_2 \le b\} = \int_a^b g_2(s)\mathrm{d}s\,.$$

We return now to the concept of conditional probabilities (Definition 1.3). The situation often arises in which the experimentalist has knowledge regarding one random variable and would like to use that knowledge in predicting the value of the other (unknown) random variable. Such predictions are possible through conditional probability functions

Definition 1.16. Let f be a joint pmf for the discrete random variables X_1 and X_2 with f_2 the marginal pmf for X_2. Then the *conditional pmf* for X_1 given that $X_2 = b$ is defined, if $\Pr\{X_2 = b\} \ne 0$, to be

$$f_{1|b}(a) = \frac{f(a,b)}{f_2(b)}\,,$$

where

$$\Pr\{X_1 = a | X_2 = b\} = f_{1|b}(a)\,.$$

Definition 1.17. Let g be a joint pdf for continuous random variables X_1 and X_2 with g_2 the marginal pdf for X_2. Then the *conditional pdf* for X_1 given that $X_2 = b$ is defined to be

$$g_{1|b}(a) = \frac{g(a,b)}{g_2(b)} ,$$

where

$$\Pr\{a_1 \le X_1 \le a_2 | X_2 = b\} = \int_{a_1}^{a_2} g_{1|b}(s)\mathrm{d}s .$$

The conditional statements for X_2 given a value for X_1 are made similarly to Definitions 1.16 and 1.17 with the subscripts reversed. These conditional statements can be illustrated by using Example 1.12. It has already been determined that the probability of having a box full of defective phones is 3%; however, let us assume that it is already known that we have picked a box of radio phones. Now, given a box of radio phones, the probability of both phones being defective is

$$f_{2|a=1}(2) = \frac{f(1,2)}{f_1(1)} = \frac{0.01}{0.53} = 0.0189 ;$$

thus, knowledge that the box consisted of radio phones enabled a more accurate prediction of the probabilities that both phones were defective. Or to consider a different situation, assume that we know the box has both phones defective. The probability that the box contains plain phones is

$$f_{1|b=2}(0) = \frac{f(0,2)}{f_2(2)} = \frac{0.02}{0.03} = 0.6667 .$$

Example 1.13. Let X and Y be two continuous random variables with joint pdf given by

$$f(x,y) = \frac{4}{3}(x^3 + y) \text{ for } 0 \le x \le 1, 0 \le y \le 1 .$$

Utilizing Definition 1.15, we obtain

$$f_1(x) = \frac{4}{3}(x^3 + 0.5) \text{ for } 0 \le x \le 1$$

$$f_2(y) = \frac{4}{3}(y + 0.25) \text{ for } 0 \le y \le 1 .$$

To find the probability that Y is less than or equal to 0.5, we perform the following steps:

$$\begin{aligned}\Pr\{Y \le 0.5\} &= \int_0^{0.5} f_2(y)\mathrm{d}y \\ &= \frac{4}{3}\int_0^{0.5}(y + 0.25)\mathrm{d}y = \frac{1}{3} .\end{aligned}$$

To find the probability that Y is less than or equal to 0.5 given we know that $X = 0.1$, we perform

$$\begin{aligned}\Pr\{Y \leq 0.5|X = 0.1\} &= \int_0^{0.5} f_{2|0.1}(y)dy \\ &= \int_0^{0.5} \frac{0.1^3 + y}{0.1^3 + 0.5} dy \\ &= \frac{0.1255}{0.501} \approx \frac{1}{4} .\end{aligned}$$

□

Example 1.14. Let U and V be two continuous random variables with joint pdf given by

$$g(u,v) = 8u^3 v \text{ for } 0 \leq u \leq 1, 0 \leq v \leq 1 .$$

The marginal pdf's are

$$\begin{aligned} g_1(u) &= 4u^3 \quad \text{for } 0 \leq u \leq 1 \\ g_2(v) &= 2v \quad \text{for } 0 \leq v \leq 1 . \end{aligned}$$

The following two statements are easily verified.

$$\Pr\{0.1 \leq V \leq 0.5\} = \int_{0.1}^{0.5} 2v\mathrm{d}v = 0.24$$

$$\Pr\{0.1 \leq V \leq 0.5|U = 0.1\} = 0.24 .$$

□

The above example illustrates independence. Notice in the example that knowledge of the value of U did not change the probabilities regarding the probability statement of V.

Definition 1.18. Let f be the joint probability distribution (pmf if discrete and pdf if continuous) of two random variables X_1 and X_2. Furthermore, let f_1 and f_2 be the marginals for X_1 and X_2, respectively. If

$$f(a,b) = f_1(a)f_2(b)$$

for all a and b, then X_1 and X_2 are called *independent*.

Independent random variables are much easier to work with because of their separability. However, in the use of the above definition, it is important to test the property for *all* values of a and b. It would be easy to make a mistake by stopping after the equality was shown to hold for only one particular pair of a, b values. Once independence has been shown, the following property is very useful.

Property 1.6. *Let X_1 and X_2 be independent random variables. Then*

$$E[X_1X_2] = E[X_1]E[X_2]$$

and

$$V[X_1 + X_2] = V[X_1] + V[X_2] \, .$$

Example 1.15. Consider again the random variables R and N defined in Example 1.12. We see from the marginal pmf's given in that example that $E[R] = 0.53$ and $E[N] = 0.21$. We also have

$$\begin{aligned} E[R \cdot N] &= 0 \times 0 \times 0.37 + 0 \times 1 \times 0.08 + 0 \times 2 \times 0.02 \\ &\qquad + 1 \times 0 \times 0.45 + 1 \times 1 \times 0.07 + 1 \times 2 \times 0.01 = 0.09 \, . \end{aligned}$$

Thus, it is possible to say that the random variables R and N are not independent since $0.53 \times 0.21 \neq 0.09$. If, however, the expected value of the product of two random variables equals the product of the two individual expected values, the claim of independence is *not* proven. □

We close this section by giving two final measures that are used to express the relationship between two dependent random variables. The first measure is called the *covariance* and the second measure is called the *correlation coefficient*.

Definition 1.19. The *covariance* of two random variables, X_1 and X_2, is defined by

$$cov(X_1,X_2) = E[\,(X_1 - E[X_1])(X_2 - E[X_2])\,] \, .$$

Property 1.7. *The following is often helpful as a computational aid:*

$$cov(X_1,X_2) = E[X_1X_2] - \mu_1\mu_2 \, ,$$

where μ_1 and μ_2 are the means for X_1 and X_2, respectively.

Comparing Property 1.6 to Property 1.7, it should be clear that random variables that are independent have zero covariance. However, it is possible to obtain random variables with zero covariance that are not independent. (See Example 1.17 below.) A principle use of the covariance is in the definition of the correlation coefficient, that is a measure of the linear relationship between two random variables.

Definition 1.20. Let X_1 be a random variable with mean μ_1 and variance σ_1^2. Let X_2 be a random variable with mean μ_2 and variance σ_2^2. The *correlation coefficient* , denoted by ρ, of X_1 and X_2 is defined by

$$\rho = \frac{cov(X_1,X_2)}{\sqrt{V(X_1)V(X_1)}} = \frac{E[X_1X_2]-\mu_1\mu_2}{\sigma_1\sigma_2}\ .$$

The correlation coefficient is always between negative one and positive one. A negative correlation coefficient indicates that if one random variable happens to be large, the other random variable is likely to be small. A positive correlation coefficient indicates that if one random variable happens to be large, the other random variable is also likely to be large. The following examples illustrate this concept.

Example 1.16. Let X_1 and X_2 denote two discrete random variables, where X_1 ranges from 1 to 3 and X_2 ranges from 10 to 30. Their joint and marginal pmf's are given in Table 1.3.

Table 1.3 Marginal probability mass functions of Example 1.16

	$X_2 = 10$	$X_2 = 20$	$X_2 = 30$	$f_1(\cdot)$
$X_1 = 1$	0.28	0.08	0.04	0.4
$X_1 = 2$	0.04	0.12	0.04	0.2
$X_1 = 3$	0.04	0.08	0.28	0.4
$f_2(\cdot)$	0.36	0.28	0.36	

The following facts should not be difficult to verify: $\mu_1 = 2.0$, $\sigma_1^2 = 0.8$, $\mu_2 = 20.0$, $\sigma_2^2 = 72.0$, and $E[X_1X_2] = 44.8$. Therefore the correlation coefficient of X_1 and X_2 is given by

$$\rho = \frac{44.8 - 2\times 20}{\sqrt{0.8\times 72}} = 0.632\ .$$

The conditional probabilities will help verify the intuitive concept of a positive correlation coefficient. Figure 1.16 contains a graph illustrating the conditional probabilities of X_2 given various values of X_1; the area of each circle in the figure is proportional to the conditional probability. Thus, the figure gives a visual representation that as X_1 increases, it is likely (but *not* necessary) that X_2 will increase. For example, the top right-hand circle represents $\Pr\{X_2 = 30|X_1 = 3\} = 0.7$, and the middle right-hand circle represents $\Pr\{X_2 = 20|X_1 = 3\} = 0.2$.

As a final example, we switch the top and middle right-hand circles in Fig. 1.16 so that the appearance is not so clearly linear. (That is, let $\Pr\{X_1 = 3, X_2 = 20\} = 0.28$, $\Pr\{X_1 = 3, X_2 = 30\} = 0.08$, and all other probabilities the same.) With this change, μ_1 and σ_1^2 remain unchanged, $\mu_2 = 18$, $\sigma_2^2 = 48.0$, $\mathrm{cov}(X_1,X_2) = 2.8$ and the correlation coefficient is $\rho = 0.452$. Thus, as the linear relationship between X_1 and X_2 weakens, the value of ρ becomes smaller. □

If the random variables X and Y have a linear relationship (however "fuzzy"), their correlation coefficient will be non-zero. Intuitively, the square of the correlation coefficient, ρ^2, indicates that amount of variability that is due to that linear relationship. For example, suppose that the correlation between X and Y is 0.8 so that $\rho^2 = 0.64$. Then 64% of the variability in Y is due the variability of X through their linear relationship.

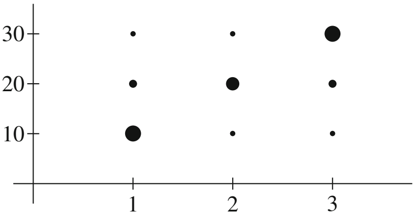


Fig. 1.16 Graphical representation for conditional probabilities of X_2 given X_1 from Example 1.16, where the correlation coefficient is 0.632

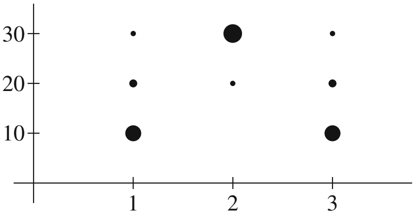


Fig. 1.17 Graphical representation for conditional probabilities of X_2 given X_1 from Example 1.17, where the correlation coefficient is zero

Example 1.17. Let X_1 and X_2 denote two discrete random variables, where X_1 ranges from 1 to 3 and X_2 ranges from 10 to 30. Their joint and marginal pmf's are given in Table 1.4.

Table 1.4 Marginal probability mass functions of Example 1.17

	$X_2 = 10$	$X_2 = 20$	$X_2 = 30$	$f_1(\cdot)$
$X_1 = 1$	0.28	0.08	0.04	0.4
$X_1 = 2$	0.00	0.02	0.18	0.2
$X_1 = 3$	0.28	0.08	0.04	0.4
$f_2(\cdot)$	0.56	0.18	0.26	

Again, we give the various measures and allow the reader to verify their accuracy: $\mu_1 = 2$, $\mu_2 = 17$, and $E[X_1X_2] = 34$. Therefore the correlation coefficient of X_1 and X_2 is zero so there is no *linear* relation between X_1 and X_2; however, the two random variables are clearly dependent. If X_1 is either one or three, then the most likely value of X_2 is 10; whereas, if X_1 is 2, then it is impossible for X_2 to have the value of 10; thus, the random variables must be dependent. If you observe the representation of the conditional probabilities in Fig. 1.17, then the lack of a linear relationship is obvious. □

- *Suggestion: Do Problems 1.21–1.26.*

1.6 Combinations of Random Variables

This probability review is concluded with a discussion of a problem type that will be frequently encountered in the next several chapters; namely, combinations of random variables. The properties of the sum of a fixed number of random variables is a straightforward generalization of previous material; however when the sum has a random number of terms, an additional variability factor must be taken into account. The final combination discussed in this section is called a mixture of random variables. An example of a mixture is the situation where the random processing time at a machine will be from different probability distributions based on the (random) product type being processed. Each of these three combinations of random variables are considered in turn.

1.6.1 Fixed Sum of Random Variables

Consider a collection of n random variables, $X_1, X_2, \cdots, X_n$ and let their sum be denoted by S; namely,

$$S = \sum_{i=1}^{n} X_i \,. \tag{1.25}$$

By a generalization of Property 1.3, we have

$$\begin{aligned} E[S] &= E[X_1 + X_2 + \cdots + X_n] \\ &= E[X_1] + E[X_2] + \cdots + E[X_n] \,. \end{aligned} \tag{1.26}$$

Note that (1.26) is valid even if the random variables are not independent.

The variance of the random variable S is obtained in a similar manner to the expected value

$$\begin{aligned} V[S] &= E[(S - E[S])^2] \\ &= E[S^2] - E[S]^2 \\ &= E[(X_1 + X_2 + \cdots + X_n)^2] - (E[X_1] + E[X_2] + \cdots + E[X_n])^2 \\ &= \sum_{i=1}^{n} E[X_i^2] + 2\sum_{i=1}^{n}\sum_{j>i}^{n} E[X_i X_j] - (E[X_1] + E[X_2] + \cdots + E[X_n])^2 \end{aligned}$$

$$\begin{aligned} &= \sum_{i=1}^{n} \left(E[X_i^2] - E[X_i]^2\right) + 2\sum_{i=1}^{n}\sum_{j>i}^{n} (E[X_i X_j] - E[X_i]E[X_j]) \\ &= \sum_{i=1}^{n} V[X_i] + 2\sum_{i=1}^{n}\sum_{j>i}^{n} cov[X_i, X_j] \,. \end{aligned} \tag{1.27}$$

Notice that when the random variables are pair-wise independent, i.e., X_i and X_j are independent for all i and j, then $E[X_iX_j] = E[X_i]E[X_j]$ and Property 1.6 is generalized indicating that the variance of the sum of n independent random variables is the sum of the individual variances. In addition, when $X_1, \cdots, X_n$ are independent *and* identically distributed (called *i.i.d.*), we have that

$$\begin{aligned} E[S] &= nE[X_1] \\ V[S] &= nV[X_1] \,. \end{aligned} \tag{1.28}$$

1.6.2 Random Sum of Random Variables

Before discussing the random sum of random variables, we need a property of conditional expectations. For this discussion we follow the development in [4] in which these properties are developed assuming discrete random variables because the discrete case is more intuitive than the continuous case. (Although the development below only considers the discrete case, our main result — given as Property 1.8 — is true for both discrete and continuous random variables.)

Let Y and X be two random variables. The conditional probability that the random variable Y takes on a value b given that the random variable X takes the value a is written as

$$\Pr\{Y = b|X = a\} = \frac{\Pr\{Y = b, X = a\}}{\Pr\{X = a\}}, \quad \text{if } \Pr\{X = a\} \neq 0$$

(see Definition 1.16). Thus, the conditional expectation of Y given that $X = a$ changes as the value a changes so it is a function, call it g, of a; namely,

$$E[Y|X = a] = \sum_b b\Pr\{Y = b|X = a\} = g(a) \,.$$

Hence, the conditional expectation of Y given X is a random variable since it depends on the value of X, expressed as

$$E[Y|X] = g(X) \,. \tag{1.29}$$

Taking the expectation on both sides of (1.29), yields the (unconditional) expectation of Y and gives the following important property.

Property 1.8. *Let Y and X be any two random variables with finite expectation. The conditional expectation of Y given X is a random variable with expectation given by*

$$E\left[E[Y|X]\right] = E[Y] \,.$$

Property 1.8 can now be used to obtain the properties of a random sum of random variables. Let S be defined by

$$S = \sum_{i=1}^{N} X_i \, ,$$

where $X_1, X_2, \cdots$ is a sequence of $i.i.d.$ random variables, and N is a nonnegative discrete random variable independent of each X_i. (When $N = 0$, the random sum is interpreted to be zero.) For a fixed n, Eq. (1.28) yields

$$E\left[\sum_{i=1}^{N} X_i | N = n\right] = nE[X_1] \, , \text{ thus}$$
$$E\left[\sum_{i=1}^{N} X_i | N\right] = NE[X_1] \, .$$

The expected value of the random sum can be derived from the above result using Property 1.8 regarding conditional expectations as follows:

$$\begin{aligned} E[S] &= E\left[E\left[\sum_{i=1}^{N} X_i | N\right]\right] \\ &= E[NE[X_1]] \\ &= E[N]E[X_1] \, . \end{aligned}$$

Note that the final equality in the above arises using Property 1.6 regarding independence and the fact that each random variable in an $i.i.d.$ sequence has the same mean.

We obtain the variance of the random variable S in a similar fashion, using $V[S] = E[S^2] - E[S]^2$ but we shall leave its derivation for homework with some hints (see Problem 1.29). Thus, we have the following property:

Property 1.9. *Let $X_1, X_2, \cdots$ be a sequence of i.i.d. random variables where for each i, $E[X_i] = \mu$ and $V[X_i] = \sigma^2$. Let N be a nonnegative discrete random variable independent of the i.i.d. sequence, and let $S = \sum_{i=1}^{N} X_i$. Then*

$$\begin{aligned} E[S] &= \mu E[N] \\ V[S] &= \sigma^2 E[N] + \mu^2 V[N] \, . \end{aligned}$$

Notice that the squared coefficient of variation of the random sum can also be easily written as

$$C^2[S] = C^2[N] + \frac{C^2[X]}{E[N]} \, , \text{ where } C^2[X] = \frac{\sigma^2}{\mu^2} \, .$$

1.6.3 Mixtures of Random Variables

The final type of random variable combination that we consider is a mixture of random variables. For example, consider two products processed on the same machine, where the two product types have different processing characteristics. Specifically, let X_1 and X_2 denote the random processing times for types 1 and 2, respectively, and then let T denote the processing time for an arbitrarily chosen part. The processing sequence will be assumed to be random with p_1 and p_2 being the probability that type 1 and type 2, respectively, are to be processed. In other words, T will equal X_1 with probability p_1 and T will equal X_2 with probability p_2. Intuitively, we have the following relationship.

$$T = \begin{cases} X_1 & \text{with probability } p_1 , \\ X_2 & \text{with probability } 1 - p_1 \ . \end{cases}$$

Thus, T is said to be a mixture of X_1 and X_2. In generalizing this concept, we have the following definition.

Definition 1.21. Let $X_1, \cdots, X_n$ be a sequence of independent random variables and let I be a positive discrete random variable with range $1, \cdots, n$ independent of the $X_1, \cdots, X_n$ sequence. The random variable T is called a *mixture of random variables with index I* if it can be written as

$$T = X_I \ .$$

Making use of Property 1.8, it should not be too difficult to show the following property.

Property 1.10. *Let T be a mixture of $X_1, \cdots, X_n$ where the mean of X_i is μ_i and variance of X_i is σ_i^2. Then*

$$E[T] = \sum_{i=1}^{n} p_i \mu_i$$
$$E[T^2] = \sum_{i=1}^{n} p_i \left(\sigma_i^2 + \mu_i^2 \right) ,$$

where $\Pr\{I = i\} = p_i$ are the probabilities associated with the index.

Notice that the above property gives the first and second moment, not the variance directly. If the variance is desired, the equation $V[T] = E[T^2] - E[T]^2$ must be used.

- *Suggestion: Do Problems 1.27–1.31.*

Appendix

In this appendix, two numerical problems are discussed: the computation of the gamma function (Eq. 1.18) and the determination of the shape and scale parameters for the Weibull distribution. We give suggestions for those using Microsoft Excel and those who are interested in doing the computations within a programming environment.

The gamma function: For Microsoft Excel users, the gamma function is evaluated by first obtaining the natural log of the function since Excel provides an automatic function for the log of the gamma instead of the gamma function itself. For example, to obtain the gamma function evaluated at 1.7, use the formula `"=EXP(GAMMALN(1.7))"`. This yields a value of 0.908639.

For programmers who need the gamma function, there are some good approximations are available. A polynomial approximation taken from [5, p. 155] is

$$\Gamma(1+x) \approx 1 + a_1 x + a_2 x^2 + \cdots + a_5 x^5 \text{ for } 0 \le x \le 1, \tag{1.30}$$

where the constants are $a_1 = -0.5748646$, $a_2 = 0.9512363$, $a_3 = -0.6998588$, $a_4 = 0.4245549$, and $a_5 = -0.1010678$. (Or if you need additional accuracy, an eight term approximation is also available in [5] or [1, p. 257].) If it is necessary to evaluate $\Gamma(x)$ for $x < 1$ then use the relationship

$$\Gamma(x) = \frac{1}{x}\Gamma(1+x)\,. \tag{1.31}$$

If it is necessary to evaluate $\Gamma(n+x)$ for $n > 1$ and $0 \le x \le 1$, then use the relationship:

$$\Gamma(n+x) = (n-1+x)(n-2+x)\cdots(1+x)\Gamma(1+x)\,. \tag{1.32}$$

Example 1.18. Suppose we wish to compute $\Gamma(0.7)$. The approximation given by (1.30), yields a result of $\Gamma(1.7) = 0.9086$. Applying (1.31) yields $\Gamma(0.7) = 0.9086/0.7 = 1.298$. Now suppose that we wish to obtain the gamma function evaluated at 5.7. From (1.32), we have $\Gamma(5.7) = 4.7 \times 3.7 \times 2.7 \times 1.7 \times 0.9086 = 72.52$. □

Weibull parameters: The context for this section is that we know the first two moments of a Weibull distribution (1.20) and would like to determine the shape and scale parameters. Notice that the SCV can be written as $C^2[X] = E[X^2]/(E[X])^2 - 1$; thus, the shape parameter is the value of α that satisfies

$$C^2[X] + 1 = \frac{\Gamma(1+2/\alpha)}{(\Gamma(1+1/\alpha))^2}\,, \tag{1.33}$$

and the scale parameter is then determined by

$$\beta = \frac{E[X]}{\Gamma(1+1/\alpha)} \tag{1.34}$$

Example 1.19. Suppose we would like to find the parameters of the Weibull random variable with mean 100 and standard deviation 25. We first note that $C^2[X]+1 = 1.0625$. We then fill in a spreadsheet with the following values and formulas.

	A	B
1	`mean`	100
2	`st.dev.`	25
3	`alpha-guess`	1
4	`first moment term`	`=EXP(GAMMALN(1+1/B3))`
5	`second moment term`	`=EXP(GAMMALN(1+2/B3))`
6	`ratio, Eq. (1.31)`	`= B5/(B4*B4)`
7	`difference`	`=1+B2*B2/(B1*B1)-B6`
8	`beta-value`	`=B1/B4`

The GoalSeek tool (found under the "Tools" menu in Excel 2003 and under the "What-If" button on the Data Tab for Excel 2007) is ideal for solving (1.33). When GoalSeek is clicked, a dialog box appears with three parameters. For the above spreadsheet, the "Set cell" parameter is set to `B7`, the "To value" parameter is set to `0`, and the "By changing cell" parameter is set to `B3`. The results should be that the `B3` cell is changed to 4.5 and the `B8` cell is changed to 109.6. □

Problems

1.1. A manufacturing company ships (by truckload) its product to three different distribution centers on a weekly basis. Demands vary from week to week ranging over 0, 1, and 2 truckloads needed at each distribution center. Conceptualize an experiment where a week is selected and then the number of truckloads demanded at each of the three centers are recorded.
(a) Describe the sample space, i.e., list all outcomes.
(b) How many possible different events are there?
(c) Write the event that represents a total of three truckloads are needed for the week.
(d) If each event containing a single outcome has the same probability, what is the probability that a total demand for three truckloads will occur?

1.2. A library has classified its books into fiction and nonfiction. Furthermore, all books can also be described as hardback and paperback. As an experiment, we shall pick a book at random and record whether it is fiction or nonfiction and whether it is paperback or hardback.
(a) Describe the sample space, i.e., list all outcomes.
(b) Describe the event space, i.e., list all events.
(c) Define a probability measure such that the probability of picking a nonfiction paperback is 0.15, the probability of picking a nonfiction book is 0.30, and the probability of picking a fiction hardback is 0.65.
(d) Using the probabilities from part (c), find the probability of picking a fiction book given that the book chosen is known to be a paperback.

1.3. Let N be a random variable describing the number of defective items in a box from Example 1.1. Draw the graph for the cumulative distribution function of N and give its pmf.

1.4. Let X be a random variable with cumulative distribution function given by

$$G(a) = \begin{cases} 0 & \text{for } a < 0, \\ a^2 & \text{for } 0 \le a < 1, \\ 1 & \text{for } a \ge 1. \end{cases} .$$

(a) Give the pdf for X.
(b) Find $\Pr\{X \ge 0.5\}$.
(c) Find $\Pr\{0.5 < X \le 0.75\}$.
(d) Let X_1 and X_2 be independent random variables with their *CDF* given by $G(\cdot)$. Find $\Pr\{X_1 + X_2 \le 1\}$.

1.5. Let T be a random variable with pdf given by

$$f(t) = \begin{cases} 0 & \text{for } t < 0.5, \\ k\mathrm{e}^{-2(t-0.5)} & \text{for } t \ge 0.5. \end{cases} .$$

(a) Find k.
(b) Find $\Pr\{0.25 \le T \le 1\}$.
(c) Find $\Pr\{T \le 1.5\}$.
(d) Give the cumulative distribution function for T.
(e) Let the independent random variables T_1 and T_2 have their pdf given by $f(\cdot)$. Find $\Pr\{1 \le T_1 + T_2 \le 2\}$.
(f) Let $Y = X + T$, where X is independent of T and is defined by the previous problem. Give the pdf for Y.

1.6. Let U be a random variable with pdf given by

$$h(u) = \begin{cases} 0 & \text{for } u < 0, \\ u & \text{for } 0 \le u < 1, \\ 2 - u & \text{for } 1 \le u < 2, \\ 0 & \text{for } u \ge 2. \end{cases} .$$

(a) Find $\Pr\{0.5 < U < 1.5\}$.
(b) Find $\Pr\{0.5 \le U \le 1.5\}$.
(c) Find $\Pr\{0 \le U \le 1.5,\ 0.5 \le U \le 2\}$. (A comma acts as an intersection and is read as an "and".)
(d) Give the cumulative distribution function for U and calculate $\Pr\{U \le 1.5\} - \Pr\{U \le 0.5\}$.

1.7. An independent roofing contractor has determined that the number of jobs obtained for the month of September varies. From previous experience, the probabilities of obtaining 0, 1, 2, or 3 jobs have been determined to be 0.1, 0.35, 0.30, and

0.25, respectively. The profit obtained from each job is \$300. What is the expected profit and the standard deviation of profit for September?

1.8. There are three investment plans for your consideration. Each plan calls for an investment of \$25,000 and the return will be one year later. Plan A will return \$27,500. Plan B will return \$27,000 or \$28,000 with probabilities 0.4 and 0.6, respectively. Plan C will return \$24,000, \$27,000, or \$33,000 with probabilities 0.2, 0.5, and 0.3, respectively. If your objective is to maximize the expected return, which plan should you choose? Are there considerations that might be relevant other than simply the expected values?

1.9. Let the random variables A, B, C denote the returns from investment plans A, B, and C, respectively, from the previous problem. What are the mean and standard deviations of the three random variables?

1.10. Let N be a random variable with cumulative distribution function given by

$$F(x) = \begin{cases} 0 & \text{for } x < 1, \\ 0.2 & \text{for } 1 \le x < 2, \\ 0.5 & \text{for } 2 \le x < 3, \\ 0.8 & \text{for } 3 \le x < 4, \\ 1 & \text{for } x \ge 4. \end{cases}$$

Find the mean and standard deviation of N.

1.11. Prove that the $E[(X-\mu)^2] = E[X^2] - \mu^2$ for any random variable X whose mean is μ.

1.12. Find the mean and standard deviation for X as defined in Problem 1.4.

1.13. Show using integration by parts that

$$E[X] = \int_0^b [1 - F(x)]\mathrm{d}x, \;\; \text{for } 0 \le a \le x \le b\,,$$

where F is the CDF of a random variable with support on the interval $[a,b]$ with $a \ge 0$. Note that the lower integration limit is 0 not a. (A random variable is zero outside its interval of support.)

1.14. Find the mean and standard deviation for U as defined in Problem 1.6. Also, find the mean and standard deviation using the last two properties mentioned in Property 1.4.

Use the appropriate distribution from Sect. 1.4 to answer the questions in Problems 1.15–1.19.

1.15. A manufacturing company produces parts, 97% of which are within specifications and 3% are defective (outside specifications). There is apparently no pattern to the production of defective parts; thus, we assume that whether or not a part is

defective is independent of other parts.
(a) What is the probability that there will be no defective parts in a box of 5?
(b) What is the probability that there will be exactly 2 defective parts in a box of 5?
(c) What is the probability that there will be 2 or more defective parts in a box of 5?
(d) Use the Poisson distribution to approximate the probability that there will be 4 or more defective parts in a box of 40.
(e) Use the normal distribution to approximate the probability that there will be 20 or more defective parts in a box of 400.

1.16. A store sells two types of tables: plain and deluxe. When an order for a table arrives, there is an 80% chance that the plain table will be desired.
(a) Out of 5 orders, what is the probability that no deluxe tables will be desired?
(b) Assume that each day 5 orders arrive and that today (Monday) an order came for a deluxe table. What is the probability that the first day in which one or more deluxe tables are again ordered will be in three more days (Thursday)? What is the expected number of days until a deluxe table is desired?
(c) Actually, the number of orders each day is a Poisson random variable with a mean of 5. What is the probability that exactly 5 orders will arrive on a given day?

1.17. A vision system is designed to measure the angle at which the arm of a robot deviates from the vertical; however, the vision system is not totally accurate. The results from observations is a continuous random variable with a uniform distribution. If the measurement indicates that the range of the angle is between 9.7 and 10.5 degrees, what is the probability that the actual angle is between 9.9 and 10.1 degrees?

1.18. The dispatcher at a central fire station has observed that the time between calls is an exponential random variable with a mean of 32 minutes.
(a) A call has just arrived. What is the probability that the next call will arrive within the next half hour.
(b) What is the probability that there will be exactly two calls during the next hour?

1.19. In an automated soldering operation, the location at which the solder is placed is very important. The deviation from the center of the board is a normally distributed random variable with a mean of 0 inches and a standard deviation of 0.01 inches. (A positive deviation indicates a deviation to the right of the center and a negative deviation indicates a deviation to the left of the center.)
(a) What is the probability that on a given board the actual location of the solder deviated by less than 0.005 inches (in absolute value) from the center?
(b) What is the probability that on a given board the actual location of the solder deviated by more than 0.02 inches (in absolute value) from the center?

1.20. The purpose of this problem is to illustrate the dangers of statistics, especially with respect to categorical data and the use of conditional probabilities. In this example, the data may be used to support contradicting claims, depending on the inclinations of the person doing the reporting! The population in which we are interested is made up of males and females, those who are sick and not sick, and those who received treatment prior to becoming sick and who did not receive prior treatment. (In the questions below, assume that the treatment has no adverse side effects.) The population numbers are as follows.

Males		
	sick	not sick
treated	200	300
not treated	50	50

.

Females		
	sick	not sick
treated	50	100
not treated	200	370

.

(a) What is the conditional probability of being sick given that the treatment was received and the patient is a male?
(b) Considering only the population of males, should the treatment be recommended?
(c) Considering only the population of females, should the treatment be recommended?
(d) Considering the entire population, should the treatment be recommended?

1.21. Let X and Y be two discrete random variables where their joint pmf

$$f(a,b) = \Pr\{X = a, Y = b\}$$

is defined by

	0	1	2
10	0.01	0.06	0.03
11	0.02	0.12	0.06
12	0.02	0.18	0.10
13	0.07	0.24	0.09

with the possible values for X being 10 through 13 and the possible values for Y being 0 through 2.
(a) Find the marginal pmf's for X and Y and then find the $\Pr\{X = 11\}$ and $E[X]$.
(b) Find the conditional pmf for X given that $Y = 1$ and then find the $\Pr\{X = 11|Y = 1\}$ and find the $E[X|Y = 1]$.
(c) Are X and Y independent? Why or why not?
(d) Find $\Pr\{X = 13, Y = 2\}$, $\Pr\{X = 13\}$, and $\Pr\{Y = 2\}$. (Now make sure your answer to part (c) was correct.)

1.22. Let S and T be two continuous random variables with joint pdf given by

$$f(s,t) = kst^2 \text{ for } 0 \le s \le 1,\ 0 \le t \le 1\ ,$$

and zero elsewhere.
(a) Find the value of k.

(b) Find the marginal pdf's for S and T and then find the $\Pr\{S \le 0.5\}$ and $E[S]$.
(c) Find the conditional pdf for S given that $T = 0.1$ and then find the $\Pr\{S \le 0.5|T = 0.1\}$ and find the $E[S|T = 0.1]$.
(d) Are S and T independent? Why or why not?

1.23. Let U and V be two continuous random variables with joint pdf given by

$$g(u,v) = \mathrm{e}^{-u-v} \text{ for } u \ge 0,\ v \ge 0\,,$$

and zero elsewhere.
(a) Find the marginal pdf's for U and V and then find the $\Pr\{U \le 0.5\}$ and $E[U]$.
(b) Find the conditional pdf for U given that $V = 0.1$ and then find the $\Pr\{U \le 0.5|V = 0.1\}$ and find the $E[U|V = 0.1]$.
(c) Are U and V independent? Why or why not?

1.24. This problem is to consider the importance of keeping track of history when discussing the reliability of a machine and to emphasize the meaning of Eq. (1.16). Let T be a random variable that indicates the time until failure for the machine. Assume that T has a uniform distribution from zero to two years and answer the question, "What is the probability that the machine will continue to work for at least three more months?"
(a) Assume the machine is new.
(b) Assume the machine is one year old and has not yet failed.
(c) Now assume that T has an exponential distribution with mean one year, and answer parts (a) and (b) again.
(d) Is it important to know how old the machine is in order to answer the question, "What is the probability that the machine will continue to work for at least three more months?"

1.25. Determine the correlation coefficient for the random variables X and Y from Example 1.13.

1.26. A shipment containing 1,000 steel rods has just arrived. Two measurements are of interest: the cross-sectional area and the force that each rod can support. We conceptualize two random variables: A and B. The random variable A is the cross-sectional area, in square centimeters, of the chosen rod, and B is the force, in kilo-Newtons, that causes the rod to break. Both random variables can be approximated by a normal distribution. (A generalization of the normal distribution to two random variables is called a bivariate normal distribution.) The random variable A has a mean of 6.05 cm^2 and a standard deviation of 0.1 cm^2. The random variable B has a mean of 132 kN and a standard deviation of 10 kN. The correlation coefficient for A and B is 0.8.

To answer the questions below use the fact that if X_1 and X_2 are bivariate normal random variables with means μ_1 and μ_2, respectively, variances σ_1 and σ_2, respectively, and a correlation coefficient ρ, the following hold:

- The marginal distribution of X_1 is normal.

- The conditional distribution of X_2 given X_1 is normal.
- The conditional expectation is given by

$$E[X_2|X_1 = x] = \mu_2 + \rho \frac{\sigma_2}{\sigma_1}(x - \mu_1) \, .$$

- the conditional variance is given by

$$V[X_2|X_1 = x] = \sigma_2^2(1 - \rho^2) \, .$$

(a) Specifications call for the rods to have a cross-sectional area of between 5.9 cm^2 and 6.1 cm^2. What is the expected number of rods that will have to be discarded because of size problems?
(b) The rods must support a force of 31 kN, and the engineer in charge has decided to use a safety factor of 4; therefore, design specifications call for each rod to support a force of at least 124 kN. What is the expected number of rods that will have to be discarded because of strength problems?
(c) A rod has been selected, and its cross-sectional area measures 5.94 cm^2. What is the probability that it will not support the force required in the specifications?
(d) A rod has been selected, and its cross-sectional area measures 6.08 cm^2. What is the probability that it will not support the force required in the specifications?

1.27. Using Property 1.8, show the following relationship holds for two dependent random variables, X and Y:

$$V[Y] = E[V[Y|X]] \, + \, V[E[Y|X]] \, .$$

1.28. Let X_1 and X_2 be two independent Bernoulli random variables with $E[X_1] = 0.8$ and $E[X_2] = 0.6$. Let $S = X_1 + X_2$.
(a) Give the joint pmf for S and X_1.
(b) Give the marginal pmf for S.
(c) Give the correlation coefficient for S and X_1.
(d) Give the conditional pmf for S given $X_1 = 0$ and $X_1 = 1$.
(e) Demonstrate that Property 1.8 is true where $Y = S$ and $X = X_1$.
(f) Demonstrate that the property given in Problem 1.27 is true where $Y = S$ and $X = X_1$.

1.29. Derive the expression for the variance in Property 1.9. For this proof, you will need to use the following two equations:

$$E[S^2] = E\left[E\left[\left(\sum_{i=1}^{N} X_i\right)^2 |N\right]\right] \, ,$$

and

$$E\left[\left(\sum_{i=1}^{n} X_i\right)^2\right] = E\left[\sum_{i=1}^{n} X_i^2 + \sum_{i=1}^{n}\sum_{j \neq i} X_i X_j\right] \, .$$

1.30. Consider again the roofing contractor of Problem 1.7. After further analysis, it has been determined that the profit from each job is not exactly \$300, but is random following a normal distribution with a mean of \$300 and a standard deviation of \$50. What is the expected profit and the standard deviation of profit for September?

1.31. Consider again the three investment plans of Problem 1.8. An investor who cannot decide which investment option to use has decided to toss two (fair) coins and pick the investment plan based on the random outcome of the coin toss. If two heads occur, Plan A will be used; if a head and a tail occurs, Plan B will be used; if two tails occur, Plan C will be used. What is the mean and standard deviation of return from the investment plan?

References

1. Abramowitz, M., and Stegun, I.A., editors (1970). *Handbook of Mathematical Functions*, Dover Publications, Inc., New York.
2. Barlow, R.E., and Proschan, F. (1975). *Statistical Theory of Reliability and Life Testing*, Holt, Rinehart and Winston, Inc., New York.
3. Feldman, R.M., and Valdez-Flores, C. (2010). *Applied Probability & Stochastic Processes*, Second Edition, Springer-Verlag, Berlin.
4. Çinlar, E. (1975). *Introduction to Stochastic Processes*, Prentice-Hall, Inc., Englewood Cliffs, NJ.
5. Hastings, Jr., C. (1955). *Approximations for Digital Computers*, Princeton University Press, Princeton, NJ.
6. Law, A.M., and Kelton, W.D. (1991). *Simulation Modeling and Analysis*, 2nd edition, McGraw-Hill Book Company, New York.
7. Rahman, N.A. (1968). *A Course in Theoretical Statistics*, Hafner Publishing Co., New York.

Chapter 2
Introduction to Factory Models

An analytical approach to the modeling and analysis of manufacturing and production systems is the cornerstone of the ability to quickly evaluate alternatives (called rapid scenario analysis) and is the emphasis of the material in this textbook. Pertinent factors must be identified while secondary factors will generally be ignored. Starting with extremely simple models (essentially single machine/resource models), the necessary mechanics and concepts needed to model these situations are developed. Then more complex models are developed by connecting simple models into networks of workstations with the appropriate interconnections. The overall approach is to decompose a system into small components, model these components, and then reintegrate the general system by the appropriate combination of the components' submodels. This decomposition approach is an approximation procedure that has given acceptable results in a wide variety of manufacturing applications. In reality any analytical model, whether exact or approximate, is an approximation of the real world environment. The question that must be answered is whether or not the model yields accurate enough results to be used as an analysis tool in support of design and operational decision making.

2.1 The Basics

The modeling perspective or scope throughout this textbook will start when jobs arrive to the system and end when they are completed. The model scope, depicted in Figure 2.1, will not take into account where or why jobs arrive or how they are transported to customers. Thus, modeling the order creation or completed job delivery systems is not within the scope of our analysis. It is important to observe that we use the term "job" loosely. An arriving job may be a physical entity that must be processed through the various processing steps or an arriving job may be an order to begin the processing of (on-hand) raw material into a newly manufactured entity.

G.L. Curry, R.M. Feldman, *Manufacturing Systems Modeling and Analysis*, 2nd ed.,
DOI 10.1007/978-3-642-16618-1_2, © Springer-Verlag Berlin Heidelberg 2011

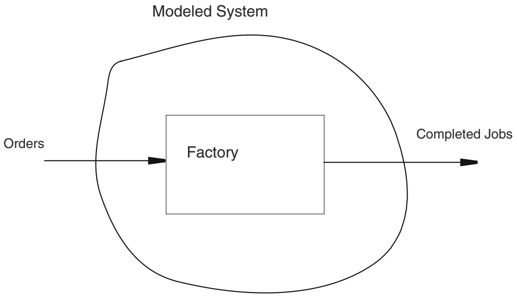


Fig. 2.1 Scope of the system for modeling purposes

To provide the framework necessary for analytical model development, we begin with the basic definitions and notation that will be used throughout the book. In addition, some fundamental relationships involving key factory parameters will be developed. Thus, this section presents terminology and material that will be used for all future factory models.

2.1.1 Notation, Definitions and Diagrams

From our point of view, a factory consists of several machines grouped together by type (called *workstations*) and a series of jobs that are to be produced on these machines. The processing steps for a job generally consists of several processing operations to be performed by different machines in a specified sequence. Thus, one can think of a job as moving through the factory, waiting in line at a machine (workstation) until its turn for processing, being processed on the machine, then proceeding to the next machine location to repeat this sequence until all required operations have been completed. Jobs arrive at the factory either individually or in batches based on some distribution of the time between arrivals, these jobs are processed, and upon completion are shipped to a customer or warehouse.

Possibly the two most important performance measures of a factory are cycle time and work-in-process. These two terms are defined as follows:

Definition 2.1. *Cycle time* is the time that a job spends within a system. The average cycle time is denoted by CT.

Definition 2.2. *Work-in-process* is the number of jobs within a system that are either undergoing processing or waiting in a queue for processing. The average work-in-process is denoted by WIP.

We will need to refer to the cycle time within a workstation as well as the cycle time for the factory as a whole. Thus, a notational distinction must be made between the average factory cycle time denoted as CT_s and the average cycle time at workstation i (the i^{th} grouping of identical machines) denoted as $CT(i)$. Thus,

CT_s is the average time that a job spends within the factory, either being processed at a workstation or waiting in a workstation queue; whereas, $CT(i)$ is the average time jobs spend being processed by workstation *i* plus the average time spend in the workstation *i* queue (or buffer). At times, general properties related to the average cycle time will be developed, in which case CT is used without subscript. At other it will be important to specifically refer to the average cycle time at a workstation or within the entire factory, in which case either $CT(i)$ or CT_s will be used.

To add to the notational confusion, the cycle time at a machine consists of two components, the processing time and the waiting time or queue time at the machine until its processing begins. The processing time at a machine is often known or can be determined without much effort; however, the queue time at a machine is not easily estimated for a given job since it depends on the number and processing times of the various types of jobs that are waiting in the queue ahead of the designated job. Thus, the average cycle time at workstation *i* is given as the sum of two components; namely,

$$CT(i) = CT_q(i) + E[T_s(i)] \, , \tag{2.1}$$

where $CT_q(i)$ denotes the average time a job spends in the queue in front of the workstation and $T_s(i)$ denotes the service time (or processing time) at workstation *i*. (We have just introduced a potential source of confusion in notation, but it should help in future chapters. The "s" subscript usually refers to a "system" characteristic; however, for the random variable T, the subscript refers to "service". The reason for this is that it will become necessary to distinguish among arrival times, departure times, and service times in later chapters.)

Another key system performance measure is the throughput rate.

Definition 2.3. The *throughput rate* for a system is the number of completed jobs leaving the system per unit of time. The throughput rate averaged over many jobs is denoted by th.

For most of the systems that we will consider, the long-run throughput rate of the system must be equal to the input rate of jobs. Given that the throughput rate is known, the main issue will then be the estimation of the total length of time for the manufacturing process (CT_s). Given that there is enough capacity to satisfy the long term average demand, the average cycle time in the factory or system is a function of the factory's capacity relative to the minimum capacity needed. The higher the factory capacity relative to the needs, the faster jobs are completed. Thus, cycle time increases as the factory becomes busier.

As mentioned above, a workstation can be either be a single machine or multiple machines.

Definition 2.4. A *workstation* (or machine group) is a collection of one or more identical machines or resources.

Non-identical machines will not generally be grouped together into a single workstation for purposes of analysis in this text. Also only one type of resource is considered at each workstation. For example, a system that has an operator handling more

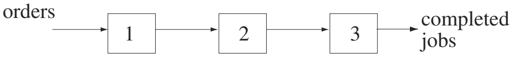


Fig. 2.2 Representation of a factory structure containing workstations and job flow

than one machine at a time is a realistic situation; however, the impact of operator availability on the total system cycle time and throughput should be second-order effects given a reasonable level of operator capacity. For those readers interested in this extension we suggest [1] for further reading. In a general manufacturing context, workstations are sometimes made up of several different machine types called cells where these machines are gathered together for the purpose of performing several distinct processing steps at one physical location. Again, a more restricted definition of this concept is used herein, where the workstation term specifically implies a location consisting of one or more identical machines. In order to model a cell type workstation, one would need to combine several single-machine workstations together.

A *processing step* for a job consists of a specific machine or workstation and the processing time (possibly processing time distribution) for the step. After processing steps have been defined they are organized into routes.

Definition 2.5. The sequence of processing steps for a job is called its *routing*. Jobs with identical routings are said to be of the same *job type*; thus, different job types are jobs with different routings.

The characteristics of all the job routings determine the organization of a manufacturing facility that is used to produce these jobs. If there is a unique routing, then an assembly line could be used within the factory given a high enough throughput rate. When there are only a few routings (a low diversity of job types) with each routing visiting a workstation at most one time, then the factory is referred to as a *flow shop*. When there are a large number of different job routings (a high diversity of jobs types) so that jobs visit workstations with no apparent structure, seemingly random, then the factory is referred to as a *job shop*. In a job shop, a given job type can visit the same workstation several times for different processing operations. In practice, many factories fall somewhere between these two extremes so that there may be characteristics of both flow shops and job shops within one facility. The methodologies that are developed will allow the analysis of all these various configurations. It will seem, due to the sequential manner in which the methodologies are developed, that there is a one-to-one correspondence between workstations and processing steps. However, as the models get more complex, routing steps and workstations will not have a one-to-one correspondence because a given workstation could be visited in several processing steps within the same job routing. This type of routing is called *re-entrant flow*, and requires more careful analysis in that machine loads are developed over job types and multiple processing steps within each routing.

Diagrams used to illustrate the nature of a modeled system will omit the system level structure and emphasize the internal structure of the model itself. The level of

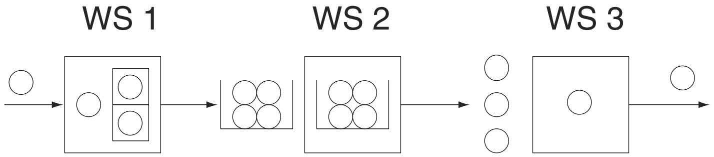


Fig. 2.3 Detailed diagram depicting the two machines in Workstation 1, a batch processing operation at Workstation 2, and individual processing on a single machine at Workstation 3

detail generally needed in diagrams will include workstations and job flow within the factory. So a diagram such as Fig. 2.2 will be used to illustrate the structure of the factory characteristics being analyzed (in this example a single job type arrives and is serially processed through workstations 1, 2 and 3).

The structure within a workstation will frequently be depicted by detailing the machines when a workstation includes more than one machine. Also there can be batch processing where multiple jobs are processed simultaneously by a single machine. Another variation is batch moves where jobs are grouped together for transportation purposes within the factory and then served individually by the machines but kept together for movement purposes. The details of the notation is best described in context where it is needed and developed. However, the general graphical depiction of the system such as the one presented in Fig. 2.3 will be used. Jobs are generally represented by circles and machines by rectangles. For the system depicted in Fig. 2.3, two machines are available for processing in the first workstation, the second workstation requires that four jobs are grouped together for an oven batch processing operation and then jobs are sent on to the third machine individually but with batch processing timing. This causes jobs to arrive at the third station in batches, even though they are not physically grouped together as they were for the oven processing step.

2.1.2 Measured Data and System Parameters

In the modeling and analysis of manufacturing/production systems, some common measures are almost always used. Among these are the number of arrivals and departures to and from the system. Using data collected about these events, system performance measures CT and WIP can be developed. Realistically, one should recognize that the system's characteristics vary with time. The information generally desired about cycle time is the average cycle time for the system calculated for all jobs within the system at a specified time t. This measure is denoted by $CT_s(t)$. Time dependent measures such as $CT_s(t)$ and $WIP_s(t)$ are very difficult to develop. Thus, most often our focus will be restricted to the so called "steady-state" measures that are the limiting value of the time dependent measures. By a property called the ergodic property, steady-state values can also be considered to be time-averaged val-

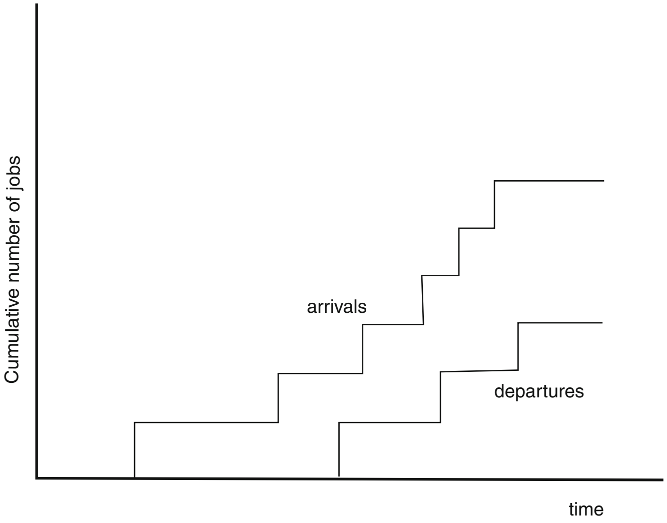


Fig. 2.4 A possible realization for the arrival $A(\cdot)$ and departure $D(\cdot)$ functions

ues as time becomes very large. These steady-state measures are independent of the initial conditions of the system. In the queueing theory that underlines the development of our factory modeling approach, most tractable results are for steady-state system measures. To quote from Gross and Harris [2]: "Fortunately, frequently, in practice, the steady-state characteristics are the main interest anyway."

It is difficult to obtain transient behavior for a system particularly when system behavior has random components. If instead of the transient system behavior, interest is in the long-run average behavior of the system (which in fact is about all the information that can be assimilated anyway) then this information is more easily developed. From a practical point of view, the long-run average system behavior can be obtained from a single realization (or a single simulation run) for most systems. Technically, the system must satisfy certain statistical conditions, called the ergodic conditions, for a steady state to exist. However, intuitively, steady-state conditions are those where the time dependent characteristics of average values vanish. In the following chapters, conditions will be established for which steady states exist based on physical properties and parameter values of the systems under consideration.

The system's performance measures CT and WIP can be estimated from the arrival and departure streams of the system. Define T_i^a as the arrival time of the i^{th} job, and the function $A(t)$ for $t \geq 0$ as the total number of arrivals during the time interval $[0,t]$. Also, define T_i^d as the departure time of the i^{th} job, and the function $D(t)$ for $t \geq 0$ as the total number of departures during the interval $[0,t]$. A realization of these two functions, $A(\cdot)$ and $D(\cdot)$ are displayed in Fig. 2.4 for a system in which arrivals and departures occur one at a time. The left most curve in Fig. 2.4 is the arrival function and the right most curve is the associated departure function.

Consider a time interval (a,b) such that the system starts empty and returns to empty. Let N_{ab} be the number of jobs that arrive to the system during the interval (a,b). We number these jobs from 1 to N, with index i representing specific jobs. Then the average waiting time, $CT(a,b)$, for jobs during this interval is given by

$$CT(a,b) = \frac{1}{N_{ab}} \sum_{i=1}^{N_{ab}} (T_i^d - T_i^a) .$$

Note that the area, AB, between the curves $A(t)$ and $D(t)$ for $a < t < b$ is merely the summation given in the above equation. This is because of the unit nature of the jumps in these functions. This area can also be obtained by standard integration methods as

$$AB = \int_a^b (A(t) - D(t))\mathrm{d}t .$$

Viewed in this manner, the area represents the integral of the number of jobs in the system at time t, since $N(t) = A(t) - D(t)$ is the number of jobs in the system at t. So the time-averaged number of jobs waiting in the system during the time interval (a,b) is given by

$$WIP(a,b) = \frac{1}{b-a} \int_a^b (A(t) - D(t))\mathrm{d}t .$$

Note then that there is a relationship between the average number in the system during the interval (a,b) and the average waiting time or cycle time in the system during this interval. Since the area between $A(\cdot)$ and $D(\cdot)$ (namely AB) is constant regardless of the method used to measure it, we have

$$WIP(a,b) = \frac{1}{b-a} AB \quad \text{and} \quad CT(a,b) = \frac{1}{N_{ab}} AB .$$

Thus, the following relationship is obtained

$$WIP(a,b) = \frac{N}{b-a} CT(a,b) .$$

One final observation is that the mean number of jobs arriving to the system per unit time, normally denoted as λ, is $N_{ab}/(b-a)$. The notation that is used then in this text is

$$WIP(a,b) = \lambda \, CT(a,b) .$$

This result is valid for any interval that starts with an empty system and ends with an empty system. In fact this relationship is the limiting behavior result, or long run average result, for stationary queueing systems, and is known as Little's Law, after the individual who proved the first general version of this relationship [4]. The result holds for individual workstations as well as the system as a whole. This relationship is fundamental and used throughout our analyses.

Property 2.1. Little's Law. *For a system that satisfies steady-state conditions, the following equation holds*

$$WIP = \lambda \times CT \ ,$$

where WIP is the long-run average number of jobs in the system, CT is the long-run average cycle time and λ is the long-run input rate of jobs to the server.

Since the average input rate is usually equal to the average throughput rate, Little's Law can also be written as $WIP = th \times CT$. It should be stressed that the limiting behavior generally estimates mean values and the actual underlying random variables for the systems can be quite variable. For example in most single workstation system models, the average number in the system, WIP, can be easily obtained. However, the behavior of the random variable representing the number in the system at any one point in time can be highly variable as is illustrated in Fig. 2.5 where the number in the system is plotted over time from a simulation. (Note that by our definition, WIP is the steady-state value of the mean of the random variable representing the number in the system.)

Also of importance is the fact that the term steady-state implies that the *mean* reaches a limiting value and thus ceases to change with respect to time. However, steady-state does *not* imply that the system itself ceases to change; the variability as shown in Fig. 2.5 continues forever (i.e., the fluctuations within the system never cease). Steady-state does imply that the entire distribution reaches a limiting value so that not only the mean but also the standard deviation, skewness, and other such measures will have limiting values.

It is often desired that analytical models of these systems describe the steady-state probability distribution. The various measures such as the mean and variance are then computed using the derived distribution. System WIP_s is a good example of one such measure. For a single server system with exponential inter-arrival times (of mean rate λ) and exponential service times (of mean rate μ), the steady-state probability of n jobs in the system is given by

$$\Pr\{N = n\} = \left(1 - \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^n \text{ for } n = 0, 1, \cdots, \infty \ ,$$

where N is the long-run number of jobs within the system. This result is developed in the next chapter.

The mean number of jobs in the system (WIP_s) is the expected value of this discrete probability distribution,

$$WIP_s = E[N] = \sum_{n=0}^{\infty} n p_n, \tag{2.2}$$

which yields

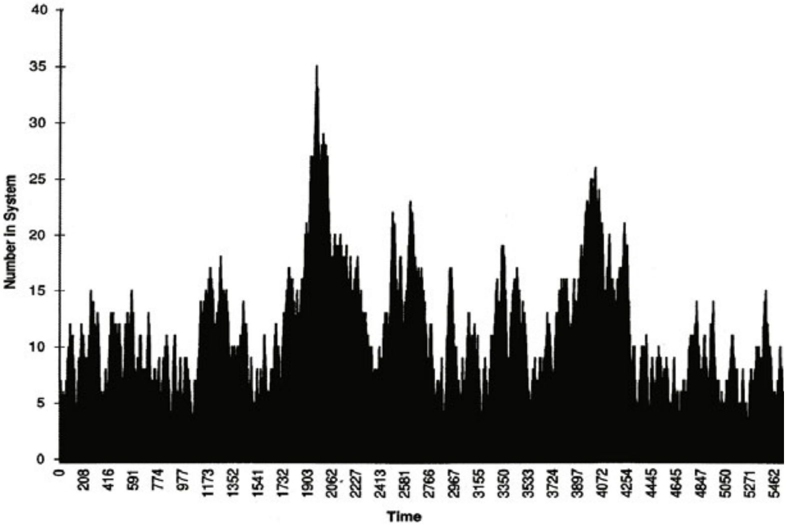


Fig. 2.5 A representation of the number of jobs in a simulated factory

$$WIP_s = \frac{\lambda/\mu}{(1-\lambda/\mu)}, \text{ given that } \lambda < \mu \ .$$

Note that the condition, $\lambda < \mu$, establishes the existence of a steady-state for this system. Using Little's Law (Definition 2.1), the expected time in the system or cycle time, CT_s, becomes

$$CT_s = \frac{1}{\mu - \lambda} \ .$$

The goal of our modeling efforts in future chapters will be to develop equations such as the above. Often, the long-run distribution will be derived and then the mean measures will be obtained from the distributions. The next chapter addresses single workstation models and the associated queueing theory mechanics for their development and approximation.

- *Suggestion: Do Problems 2.1–2.2.*

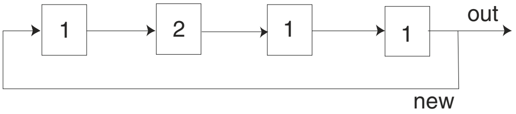


Fig. 2.6 A four machine serial flow production factory with constant service times and a constant WIP_s level

2.2 Introduction to Factory Performance

In this section a factory consisting of four machines in series with deterministic processing times is analyzed. The purpose of the model is to illustrate several issues and properties of manufacturing systems that will be studied in this text. This analysis is patterned after the "penny fab" model in [3]. Through this modeling and analysis exercise, several terms will become more meaningful. In particular, long-term or steady-state performance measures, the validity and robustness of Little's Law for these performance measures, and the impact of a bottleneck or throughput limiting machine will be illustrated.

Consider a factory that makes only one type of product. The processing requirements for this product consists of four processing steps that must be performed in sequence. Each processing operation is performed on a separate machine. These machines can process only one unit of the product at a time (called a job). The processing times for the four operations are constant. These processing times are 1, 2, 1 and 1 hour(s) on each of the four machines, respectively. This idealized factory has no machine downtimes, no product unit losses due to faulty production, and operates continuously. The factory is operated using a constant number of jobs in process (i.e., $WIP_s(t)$ is constant for all t). When a job has completed its four processing steps, it is immediately removed from the factory and a new job is started at Machine 1 to keep the total factory WIP_s at the specified level. This process is depicted in Fig. 2.6.

Since the processing times at each machine are not identical, the factory inventory will not necessarily be the same at each machine. The factory has ample storage space and the factory management policy is to move a job to the next machine area as soon as it completes processing on each machine. Thus, no machine will set idle if there is a part that is ready to be processed on that machine.

This factory is running smoothly at the current time. Management has set a constant WIP_s level at 10 jobs. This accomplishes a throughput rate of $th = 0.5/hr$ jobs (leaving the factory). That is, the factory produces one finished job every two hours on the average. This is the maximum throughput rate for this factory because its slowest processing step (at Machine 2) takes two hours per job. Thus, jobs can be completed no faster than this single machine completes its own processing because of the single unit machines and the serial nature of the production process.

Management is quite pleased with the throughput of the factory since it is at its maximum capacity. However, management is somewhat concerned with the total time that it takes a job from release to finish in the factory (the cycle time). This cycle time is currently running at 20 hours per job. Management feels like this is high since it only takes 5 hours of processing to complete each job. The ratio of the cycle time to the processing time is a standard industry measure that will be called the *x*-factor.

Definition 2.6. The *x*-factor for a factory is the ratio of CT_s to the average total processing time per job.

The average for this industry is currently running at 2.6 as reported in a recent publication by the industry's professional journal. With this factory's *x*-factor being 4, management is worried about their ability to keep customers when the industry on average produces the same product with a considerably shorter lead-time from order placement to receipt.

To address the cycle time problem, management has been considering a large capital outlay to purchase a 25% faster machine (1.5 hours) for processing step two. This purchase would be made expressly for the purpose of reducing the *x*-factor for the factory to be more in line with the industry average. The company selling the machine says that this investment will bring the *x*-factor down to 3.33 and the additional throughput of 0.166 units per hour would pay for the cost of the new machine in three years.

Management has decided that this investment is not worthwhile just based on increased throughput because the funds needed for the large capital outlay to buy the machine are sorely needed in other aspects of the company. The life blood of the company has been its ability to keep pace with the competition in new product development. This level of expenditure would decimate the company's investment in research and development of new products.

In an effort to seek a lower cost solution to the *x*-factor performance measure for the factory, a consulting team from the manufacturing engineering department of a local university was hired to perform a short term factory flow analysis study. The first activity of the consulting team was to devise a method of predicting the long-term factory performance measures of cycle time and throughput.

2.2.1 The Modeling Method

The consulting team accomplished the performance estimation task rather quickly devising a hand simulation procedure of the factory flow. They started with the specified number of 10 jobs in the factory, all placed at Machine 1, and made hourly updates to each job's status. Each job that was on a machine was allocated one hour of processing time and if this completed their requirements on that machine, the job was moved to the next machine. Empty machines were loaded with the first job in the machine queue, for those with a queue, and the next hourly update was started. The jobs soon distributed themselves throughout the factory and after a short period of time a two-hour cyclic pattern emerged. Every cycle of this pattern produced one completed job and the factory returned to the identical state for each machine and associated queue. This set of conditions is referred to as the factory status.

Once the team had the model in the cyclic behavior pattern, they would mark the time that each job entered the factory and again when it exited the factory. The difference in these times is the job cycle time. All of these cycle times had identical values after the system reached the cyclic behavior pattern. Thus, the job cycle time was determined and agreed with the company's actual cycle time of 20 hours.

The consulting team also computed the number of jobs that were completed during the marked job's residence in the model factory. When the marked job emerged this completion total was always 10 (including the marked job). In retrospect this is not surprising since a constant number of jobs is kept in the factory and, thus, when the marked job entered the factory there were 9 other jobs ahead of it in the factory. When the marked job emerged, all 9 of these jobs plus the marked job had been completed. Thus, the total throughput was 10 jobs over the cycle time of 20 hours or 0.5 jobs per hour. This modeling process exactly predicts the long-term factory performance.

The simulation study is detailed in Table 2.1, where the factory status at the start of each hour is displayed. The first entry is the initial factory setup at time 0. Notice that after hour 15 the factory status repeats every two hours; thus, the factory status at the start of hours 15, 17, 19, 21, etc. are identical. Note also that the even hours from time 16 on are also identical. In other words, this factory has reached a cyclic behavior pattern at the start of time 15. Hours 0 through 14 represent the transient phase of the simulation, and after hour 15, the limiting behavior is established.

Consider the system status at beginning with hour 15. There is a new job that has just entered (no processing has occurred) into Machine 1. There are 8 jobs at Machine 2 with no processing completed on the job in the machine. There is one job that just entered Machine 3 and Machine 4 is empty. This factory status is represented by four pairs of numbers, one for each machine. The first number in a machine pair is the number of jobs at the machine, including the job being processed, and the second number is the hours of processing at this machine already completed on the job. The last entry is the cumulative number of completed jobs through this point in time. The hour 15 the factory status entry in the table is

$$15: \ (1,0),(8,0),(1,0),(0,0) \ : 6$$

After an additional hour of processing the factory status is

$$16: \ (0,0),(9,1),(0,0),(1,0) \ : 6$$

which shows that the job in Machine 1 was completed and moved to Machine 2. The job processing in Machine 2 needs an additional hour before being completed since it requires a total of two hours for processing, and the job in Machine 3 was completed and moved to Machine 4 to begin processing.

After one more hour of processing, the job in Machine 4 is completed and removed from the factory and a new job is, therefore, entered into Machine 1. The job processing on Machine 2 is completed and moved to Machine 3. Thus, the system status at the end of time 17 is identical to that of time 15, except that one additional job is completed.

Table 2.1 Factory simulation with WIP = 10, four single-machine workstations, and processing times of (1,2,1,1) for one 24-hour day using a time step of one hour; data pairs under each workstation are the number of jobs at the workstation and the elapsed processing time for the job being processed

Time	WS #1	WS #2	WS #3	WS #4	Cum. Thru.
0	(10,0)	(0,0)	(0,0)	(0,0)	0
1	(9,0)	(1,0)	(0,0)	(0,0)	0
2	(8,0)	(2,1)	(0,0)	(0,0)	0
3	(7,0)	(2,0)	(1,0)	(0,0)	0
4	(6,0)	(3,1)	(0,0)	(1,0)	0
5	(6,0)	(3,0)	(1,0)	(0,0)	1
6	(5,0)	(4,1)	(0,0)	(1,0)	1
7	(5,0)	(4,0)	(1,0)	(0,0)	2
8	(4,0)	(5,1)	(0,0)	(1,0)	2
9	(4,0)	(5,0)	(1,0)	(0,0)	3
10	(3,0)	(6,1)	(0,0)	(1,0)	3
11	(3,0)	(6,0)	(1,0)	(0,0)	4
12	(2,0)	(7,1)	(0,0)	(1,0)	4
13	(2,0)	(7,0)	(1,0)	(0,0)	5
14	(1,0)	(8,1)	(0,0)	(1,0)	5
15	(1,0)	(8,0)	(1,0)	(0,0)	6
16	(0,0)	(9,1)	(0,0)	(1,0)	6
17	(1,0)	(8,0)	(1,0)	(0,0)	7
18	(0,0)	(9,1)	(0,0)	(1,0)	7
19	(1,0)	(8,0)	(1,0)	(0,0)	8
20	(0,0)	(9,1)	(0,0)	(1,0)	8
21	(1,0)	(8,0)	(1,0)	(0,0)	9
22	(0,0)	(9,1)	(0,0)	(1,0)	9
23	(1,0)	(8,0)	(1,0)	(0,0)	10
24	(0,0)	(9,1)	(0,0)	(1,0)	10

Computing the cycle time for a job consists of starting with a new job release into the factory and following through 10 subsequent job completions. This release occurs at the end of the given period that coincides with the start of the next time period. It is convenient to place the new job into its location in the Machine 1 list before recording the factory status so that the system maintains the required 10 jobs. Consider the job that just enters the factory at the end of time period 15 (the beginning of time period 16), this job leaves the factory at the end of time period 35 (that is, actually equal to time 36). The time in the system for this job is 36-16 = 20 hours.

The consulting team also modeled the factory under the assumption of a new Machine 2 with a constant processing time of 1.5 hours. To model this situation, the consulting team used 1/2 hour time increments for the model time step and, thus, the associated processing time requirements at the machines were $(2,3,2,2)$ in terms of the number of time steps needed to complete a job. Again the results obtained for this situation agreed with those proposed by the company trying to sell the new machine. These results were a cycle time of 30 time increments (15 hours) and a throughput rate of 2/3 jobs per hour (10 jobs every 15 hours).

2.2.2 Model Usage

The consulting team, recognizing that their modeling approach was general and being familiar with Little's Law (WIP equals throughput multiplied by CT), decided to estimate the x-factor, the ratio of cycle time to total processing time, for various numbers of jobs in the system (WIP). Letting CT represent cycle time and th represent throughput, then Little's Law (Property 2.1) yields

$$CT = WIP/th.$$

Using a throughput rate of 1/2 jobs per hour, then cycle time is given by

$$CT = 2 \times WIP\ .$$

Since the total processing time is 5 time units, then the ratio of the cycle time to the processing time called the x-factor for the factory would be

$$x = \frac{CT}{5} = \frac{WIP}{2.5}\ .$$

Notice that as long as the processing speeds of the machines do not change, the maximum throughput rate for this factor is 1/2 per hour due to the speed of the second workstation. Thus, the above formula gives shows the relationship between the x-factor and WIP provided that WIP does not get too small so as to "starve" Machine 2. Therefore, using the above formula, notice that if $WIP = 6.5$, the x-factor will equal the desired level of 2.6.

Since 6.5 is a non-integer, the constant WIP level should be set to 6 or 7. A fixed WIP level of 6 should yield an x-factor lower than 2.6 and a fixed WIP level of 7 should yield an x-factor slightly higher than 2.6 as long as the throughput rate of 1/2 per hour can be maintained.

The consulting team recognized that Little's Law is a relationship between three factory performance measures and that two of these measures must be known before the third can be obtained. The issue of concern is whether or not the throughput would stay at 1/2 when the total factory WIP was reduced below 10 jobs. For instance if only one job is allowed in the factory, the throughput rate is one job every 5 hours or 1/5. It certainly is obvious that at some job level, the factory throughput would drop below 1/2. So the consulting team decided to perform a study of the factory performance for all fixed job levels from 1 to 10 using their performance analysis modeling approach. These results are displayed in Table 2.2. They found that the WIP level in the factory can be reduced all the way down to 3 jobs while maintaining the factory throughput rate of 1/2. The cycle time reduces to $2 \times WIP = 6$ hours with an x-factor of 1.2. Thus at no expense, the factory can maintain its current throughput rate and reduce its cycle time from 20 to 6 hours. The cycle time and throughput performance measures for this factory as a function of the fixed factory WIP level are displayed in Figs. 2.7 and 2.8, respectively.

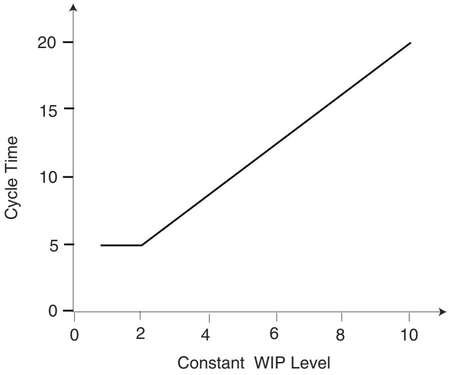


Fig. 2.7 Average cycle time for the simple factory model as a function of the constant *WIP* level

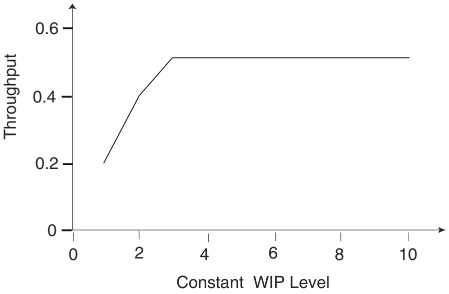


Fig. 2.8 Average throughput rate for the simple factory model as a function of the constant *WIP* level

2.2.3 Model Conclusions

Detailed consideration has been given to the factory performance measures of throughput, cycle time and work-in-process for a simple factory model. Little's Law for long-term system behavior is valid for both deterministic and stochastic factory models. Little's Law applies to individual workstations and to the system as

Table 2.2 Factory performance measures as a function of the *WIP* level

WIP	**Throughput**	**Cycle Time**	***x*-factor**
1	0.2	5	1.0
2	0.4	5	1.0
3	0.5	6	1.2
4	0.5	8	1.6
5	0.5	10	2.0
6	0.5	12	2.4
7	0.5	14	2.8
8	0.5	16	3.2
9	0.5	18	3.6
10	0.5	20	4.0

a whole. For serial systems, the factory performance is controlled by the bottleneck workstation (herein, the slowest machine). When there is enough WIP in the system the maximum throughput rate is reached and is equal to the bottleneck workstation (machine). As WIP increases beyond the minimum needed to reach the maximal throughput rate, factory cycle time performance degrades proportionally.

- *Suggestion: Do Problems 2.3–2.14.*

2.3 Deterministic vs Stochastic Models

The simple throughput analysis of a serial factory with deterministic processing times of the last section was used to illustrate several system performance measures and their inter-relationships (i.e., Little's Law). The modeling approach was developed specifically for deterministic processing times. This approach does not necessarily yield accurate results when processing times are random. If the mean processing time for a stochastic system is used in the above deterministic modeling approach, the results can be misleading and the wrong decisions can be drawn. This problem is illustrated below with a system similar to the above example. The key point to be made here is that for the evaluation of stochastic systems, stochastic methodologies should be employed. How one models stochastic production and manufacturing systems is the purpose of this book.

Consider the four-step production system represented by Fig. 2.6. Now instead of the constant processing time of two hours at workstation 2, let us assume that this time actually varies between two values: 1 hour and 3 hours. If these times occur with equal probability, then the system has a mean processing time of 2 hours and using this time one would draw the conclusions of the previous section. Recall that the principle problem was to determine the constant WIP level that yields a maximal throughput rate while maintaining a cycle time that is as small as possible. The decision arrived at using the deterministic analysis was that a WIP level of 3 jobs in the system at all times yields the maximum throughput rate of 0.5 jobs per hour with the minimal cycle time of 6 hours.

This stochastic system is now analyzed more thoroughly. One (incorrect) approach would be to develop the system performance measures using the deterministic model but recognizing that the processing times at Machine 2 are not 2 hours but 1 hour and 3 hours, with equal frequency. Thus, one approach would be to model the system using constant processing times of 1 hour and 3 hours and then average these results since these times occur with equal frequencies. This leads to the results in Table 2.3. These average results indicate that the decision to limit the constant WIP level to 3 jobs is incorrect and that 4 jobs would yield the maximum throughput of 2/3 jobs per hour. Thus, this slightly more involved (but still not proper) methodology, would indicate that the throughput level is up by 33% over the previous estimate of 0.5 jobs per hour.

Table 2.3 Weighed average throughput rate results for the factory of Fig. 2.6 with Workstation 2 processing times of 1 and 3 hours, and constant WIP levels of 3, 4 and 5

	Processing Times		
WIP	**1 hour**	**3 hours**	**Average**
3	3/4	1/3	13/24
4	1	1/3	2/3
5	1	1/3	2/3

The reason for the averaged deterministic results not yielding the correct stochastic result is that the factory throughput is not an instantaneous function of the processing rate of Machine 2. This processing rate has an impact on the number of jobs allowed into downstream machines and, hence, there is a longer term impact on system performance. The length of this impact is also such that the system might re-enter this rate status more than once while a job is in the system. Hence, complex and longer term impacts cannot be properly estimated by merely performing a weighted average of the constant processing time results. To illustrate this idea, the throughput gain for the average results is obtained from the system when the processing time is only one hour. This situation corresponds to a throughput rate of 1 job per hour (for a WIP level of at least 4 jobs). This high level of throughput is balanced by the lower throughput rate (1/3 jobs per hour) when the system has a 3 hours processing time at Machine 2. These situations occur at the machine with equal probability for a given job. However, the proportion of the time that the system is operating in the slow state is 75%. Thus, one would expect a more accurate throughput rate estimate to be

$$\frac{3}{4}\left(\frac{1}{3}\right)+\frac{1}{4}(1)=\frac{1}{2}.$$

This is the expected throughput rate for the stochastic system if the WIP level is at least the minimum of 4 jobs. If there are only 3 jobs allowed in the system simultaneously, then the throughput rate reduces to around the 0.47 jobs per hour level. Notice the detrimental effect of the variability in the processing time; namely, a necessary increase in WIP and CT to maintain the same throughput rate. In general, variability in workplace parameters always is detrimental in that it increases average work-in-process and cycle times!

The calculation of throughput rates in our stochastic system can be obtained by simulation or by the analytical decomposition method of Chap. 8. The bottom line is that stochastic systems are much more difficult to evaluate than deterministic systems and the purpose of this textbook is to expose the reader to some of the analytical approaches available for stochastic modeling of manufacturing systems.

Appendix

In this appendix, Microsoft Excel will be used to present a discrete simulation model of the factory given in Fig. 2.6 with a generalization that the processing time at Workstation 2 is random as discussed in Sect. 2.3. In the next chapter, we will present a more general simulation methodology (an event driven simulation) that can better handle continuous time. For now, we shall limit ourselves to discrete time. (For practice in developing similar models, see Problem 2.15.) We also suggest that the understanding of this material is best accomplished by reading the appendix while Excel is available so that the reader can build the spreadsheet as it is presented below.

Simulation is a very important tool, especially for testing the validity of the models and approximations developed in these chapters. Simulation modeling is generally robust with respect to modeling distributional assumptions and allows for more realistic modeling of system interactions. The price that one pays with simulation is the time requirement for obtaining accurate estimates of system performance parameters. With analytical models, the system response can often be characterized by studying the mathematical structure; while this must be accomplished in the simulation environment by experimentation that again adds another dimension to the already time consuming computational burden.

Before building the spreadsheet simulation model, it is important to understand five Excel functions. The Excel function

`RAND()`

generates random numbers that are uniformly distributed between 0.0 and 1.0. Note that the `RAND` function has no parameter, although the parentheses are used. The Excel function

`IF(`*boolean_expression, true_value, false_value*`)`

evaluates the boolean expression and returns the value contained in the second parameter if the boolean expression is true and returns the value contained in the third parameter if the boolean expression is false. The Excel `IF()` function can act similar to an `If` — `ElseIf` structure by replacing either the *true_value* or *false_value* with another `IF()` function. The above two functions can be used together to create the random law mentioned previously; namely, the function `IF(RAND()<0.5,1,3)` will yield a value of one 50% of the time and a value of three 50% of the time. The

`OFFSET(`*cell_ref, number_rows_offset, number_cols_offset*`)`

function allows for the referencing of a cell relative to another cell. For example, `OFFSET(A1,3,0)` references the `A4` cell. The function

`MATCH(`*value, array_reference, 0*`)`

will return an integer equal to the first location within the array that contains *value*. For example, if the array `B4:B8` contains the elements `1,1,2,2,3`, then the function `MATCH(2,B4:B8,0)` will return a value of 3. (There are actually three different options in the use of the `MATCH` function, and the option we need is to match by equality which is designated by the final parameter being set to zero.) The final function that will be needed is

`INDIRECT(`*string*`)`

which converts *string* to an address. For example, suppose that the cell `B5` contains the number 7, then `INDIRECT("A"&(B5+1))` refers to cell `A8`. In order to understand this evaluation, first observe that the ampersand (&) concatenates (or adds) two strings, before the concatenation occurs, the numerical value of `B5+1` is converted to a string; thus, the two strings `"A"` and `"8"` are combined to form the address `A8`.

An Excel simulation usually involves building a table similar to Table 2.1; thus, we start our spreadsheet with the following two rows.

	A	**B**	**C**	**D**	**E**
			Time-1		Time-2
1	Hour	# at WS 1	Remaining	# at WS 2	Remaining
2	0	5	1	0	0

	F	**G**	**H**	**I**	**J**
		Time-3		Time-4	Cumulative
1	# at WS 3	Remaining	# at WS 4	Remaining	Completed
2	0	0	0	0	0

	K	**L**	**M**	**N**
		Finish	Start	Cycle
1	Entity #	Time	Time	Time
2	0	0	0	0

The key difference between the Excel table and Table 2.1 is the meaning of Columns C, E, G, and I. The spreadsheet will maintain the time remaining for processing instead of the time that has already been used. In order to build the future rows, we use the following formulas in row 3.

Column A	`=A2+1`
Column B	`=B2-(C2=1)+(I2=1)`
Column C	`=IF(B3=0,0, IF(C2<=1,1,C2-1))`
Column D	`=D2-(E2=1)+(C2=1)`
Column E	`=IF(D3=0,0, IF(E2<=1, IF(RAND()<0.5,1,3),E2-1))`
Column F	`=F2-(G2=1)+(E2=1)`
Column G	`=IF(F3=0,0, IF(G2<=1,1,G2-1))`
Column H	`=H2-(I2=1)+(G2=1)`
Column I	`=IF(H3=0,0, IF(I2<=1,1,I2-1))`
Column J	`=J2+(I2=1)`
Column K	`=K2+1`
Column L	`=OFFSET($A$1,MATCH(K3,$J$2:$J$1000,0),0)`

Column M	`=OFFSET($A$1,MATCH(K3-$B$2,$J$2:$J$1000,0),0)`
Column N	`=L3-M3`

Once the formulas are entered, the range of cells `A3:N1000` should be highlighted and then the "copy down" feature (or `<ctrl>-D`) used to extend the table down. Do not be concerned that several entries in the L, M, and N columns contain number errors (i.e., `#N/A`); these are expected and should be ignored. One of the keys to understanding the above formulas is to recognize that a job undergoing processing will leave the work station whenever the time remaining at that workstation equals 1. We also use the fact that when a boolean expression is used within a mathematical expression, it will return the value 1 when `true` and return 0 when it evaluates to `false`. Because the `RAND` function is a "volatile" function, it is recomputed whenever the F9 key is pressed, so if you would like to see different realizations of the simulation, press F9.

The final step in the simulation is to report the average throughput rate (*th*) and the average cycle time *CT*. To do this, place the word `Throughput` in cell `P1`, and put `=J1000/A1000` in the `P2` cell. Remember, the row 3 formulas were copied down to row 1000; thus, the value in cell `A1000` represents the total time for the simulation and the value in cell `J1000` is the total number of jobs processed through the simulation. In other words, the `P2` cell equals the total output divided by the total time, which is the average throughput rate. In cell `P3`, place the word `CycleTime` and in the `P4` cell place

```
=AVERAGE(INDIRECT("N" & (B2+2) & ":N" & (J1000+2)))
```

which yields the average of the individual cycle times. To understand this formula, remember that the value of cell `B2` is equal to 5 and is the initial work-in-process. The value in cell `J1000` varies depending on the random outcome of the simulation. To illustrate the formula, suppose that 425 entities were processed (i.e., the value of `J1000` is 425), then the `INDIRECT` function will reference `"N7:N427"` which contain cycle times. (Other cells within column `N` will likely contain number errors.) The reason for using the `INDIRECT` function is so that when the *WIP* level is changed, the *CT* formula will be changed to include or exclude the appropriate cells.

A final suggestion can be made with respect to the throughput rate. The rate is biased towards the low side because the initial few hours are not representative of steady-state conditions. Therefore, the formula `=(J1000-J48)/(A1000-A48)` would give a better estimate of the long-run average value. The choice of considering the first two days as comprising the transient period of operation is somewhat arbitrary and can be studied further by developing graphs of the average values if so desired.

- *Suggestion: Do Problem 2.15.*

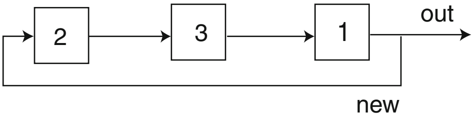
Problems

2.1. A workstation with a single machine for processing has a long-run average inventory level (WIP) of 25 jobs. The average rate at which jobs enter the workstation is 4 per hour, and the average processing time is 14.5 minutes per job. What is the average time that a job spends in the queue?

2.2. Consider a factory operating 24 hours per day consisting of two workstations. Arrivals to the first station occur at a rate of 10 per day. The long-run average time that a job spends at the first workstation is 4.2 hours. After processing at the first workstation, a job is sent directly to the second workstation where it spends an average of 5.3 hours. After processing at the second workstation, the job leaves the system. What is the average work-in-process within the factory?

2.3. Why can the example factory of Section 2.2 maintain its maximum throughput level of 1/2 job per hour even when there are less jobs in the system than there are machines?

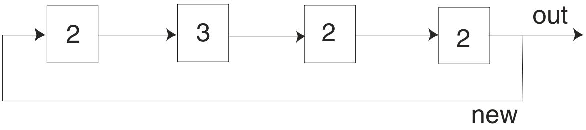
2.4. Develop a table of the factory status at the beginning of each one-hour time interval for the following serial system under the condition that the system maintains a total work-in-process of 5 jobs. Develop this table for the system status for 15 hours of operation. The workstation processing times (in hours) are listed in the squares representing the workstations. The initial (time 0) starting work-in-process distribution is $(5,0,0)$. That is, 5 jobs in the first workstation and none elsewhere and assume that the first job begins processing at time 0. Compute the cycle times (time in the system) for the first 4 completed jobs assuming that all 5 initial jobs entered the system at time 0.



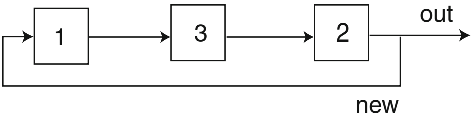
2.5. Reconsider Problem 2.4 starting with the initial conditions: work-in-process is $(2,3,0)$. Assume further that the first job in line at workstations one and two have already completed one hour of processing. Compute the cycle times for all jobs that are completed during the 15 hours of operation, assuming that all 5 initial jobs entered the system at time 0. Explain why the cycle times for the first 4 completed jobs are not valid as the long-run average.

2.6. For the factory and initial conditions of Problem 2.4, compute the long-run average factory throughput, cycle time and x-factors for various constant work-in-process levels of 1 through 5.

2.7. Compute the long-term average throughput, cycle time and x-factors for this factory for fixed work-in-process levels of 5 and 10 for the four machine serial flow factory model below where the constant processing times are listed on the machines. Argue that the results of the WIP level of 10 are the measures reported for the example factory with the new Machine 2 with a processing time of 1.5 hours.

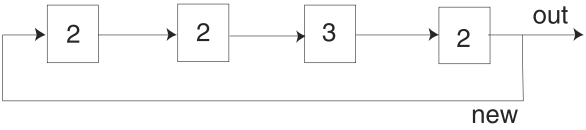


2.8. Develop a table of the factory status at the beginning of each one-hour time interval for the following serial system under the condition that the system maintains a total work-in-process of 4 jobs. Develop this table for the system status for 15 hours of operation. The workstation processing times (in hours) are listed in the squares representing the workstations. The initial (time 0) starting work-in-process distribution is $(4,0,0)$. That is, 4 jobs in the first workstation and none elsewhere and assume the first job has yet to begin processing. Compute the cycle times (time in the system) for the first 4 completed jobs assuming that all 4 initial jobs entered the system at time 0.



2.9. For the factory and initial conditions of Problem 2.8, compute the long-term average factory throughput, cycle time and x-factors for various constant WIP levels 1 through 5.

2.10. Compute the long-term average throughput, cycle time and x-factors for this factory for fixed WIP levels of 1 through 5 for the four machine serial flow factory model below where the constant processing times are listed on the machines. Assume that the factory starts with the configuration $(N,0,0,0)$ for fixed WIP level N and no processing has occurred on any of the active jobs.



2.11. Develop a spreadsheet model to solve Problem 5.

2.12. Develop a spreadsheet model to solve Problem 6.

2.13. Develop a spreadsheet model to solve Problem 7.

2.14. Develop a spreadsheet model to solve Problem 8.

2.15. Develop a spreadsheet model of the factory in Sect. 2.2 (Fig. 2.6) except that Workstation 1, 2, and 4 have random processing times that are distributed according to a discrete uniform distribution between 1 and 3, and Workstation 3 has a random processing time distributed according to a discrete uniform distribution between 1 and 4. To generate random integers uniformly between *a* and *b*, use `"=a+FLOOR((b+1-a)*RAND(),1)"`. Version 2007 or Analysis Tool Pack with an earlier version of Excel provides a function to generate the discrete uniform variates directly ; namely, `RANDBETWEEN(a,b)`. (This problem is based on material contained in the Appendix.)

References

1. Deuermeyer, B.L., Curry, G.L., and Feldman, R.M. (1993). An Automatic Modeling Approach for the Strategic Analysis of Manufacturing Facilities. *Production and Operations Management*, **2**:195–220.
2. Gross, D., and Harris, C.M. (1974). *Fundamentals of Queueing Theory*, John Wiley & Sons, New York.
3. Hopp, W.J., and Spearman, M.L. (1996). *Factory Physics: Foundations of Manufacturing Management*, Irwin, Chicago.
4. Little, J.D.C. (1961). A Proof for the Queuing Formula: $L = \lambda W$. *Operations Research*, **9**:383–387.

Chapter 3
Single Workstation Factory Models

Throughout the analyses given in this textbook, emphasis is on the development of steady-state system measures such as the expected number of jobs in the system (WIP) and their mean cycle times (CT). For these analyses, it is often useful to obtain the probability mass function (pmf) of the steady-state number of jobs in the system. From these pmf's, the measures of system effectiveness can often be developed. For notational purposes, define the random variable N as the number of jobs in the system and define p_n as the probability that the number of jobs in the system is n; namely, $p_n = \Pr\{N = n\}$. In the first section, a method is developed for deriving equations that determine the steady-state probabilities p_n for $n = 0, 1, \cdots$. The initial models will include probabilistic behavior for the arrival process and processing times, and the early models will restrict these two probability laws to the exponential distribution.

Important assumptions on the operating characteristics of the system are also made. It is assumed that job inter-arrival times are independent of the status of the system. Another operating assumption is that the server will never be idle when there is a job in the system that can be served. That is, if it is allowed for the processor to serve a job, then no delay occurs between the time that one job leaves the server and the next job begins processing on the server. Here the assumption is made that the server is always busy processing jobs when there are jobs available for service. Thus, the server will only be idle when there are no jobs available. In later models, nonproductive times will be incorporated into the model. For example, in order to have realistic models for many systems, machine breakdowns will need to be incorporated.

3.1 First Model

Consider a single server with a limited waiting area for $n_{\max} - 1$ jobs and one in the server position for a maximum of $n_{\max}$ jobs in the system. Jobs arrive to the system one at a time with exponentially distributed inter-arrival times. Denoting the mean

G.L. Curry, R.M. Feldman, *Manufacturing Systems Modeling and Analysis*, 2nd ed.,
DOI 10.1007/978-3-642-16618-1_3, © Springer-Verlag Berlin Heidelberg 2011

arrival rate as λ, the mean inter-arrival time is then $1/\lambda$. If the system is full, the arriving job is rejected (and lost to another factory). If there is room in the waiting area, the arriving job is accepted and processed in a first-come-first-serve order (this sequence is denoted by FIFO which stands for first-in first-out). The processing time is also assumed to be exponentially distributed, with mean rate μ (the mean service time is $1/\mu$).

Since this system can have at most $n_{\max}$ jobs, there are $n_{\max}+1$ possible states, $\{0,1,\cdots,n_{\max}\}$, representing the number of jobs in the system. Interest is in developing the steady-state distribution of the number of jobs in the system. Assuming that a steady-state exists, then the flow into and out of each state must balance. This balance is the key property used to establish the steady-state probability of being in each possible system state.

Let p_n denote the steady-state probability of n jobs in the system for $n = 0,\cdots,n_{\max}$. The flow *into* an intermediate state n $(0 < n < n_{\max})$ is made up of two components: (1) the arrival of a new job to the system when the system has exactly $n-1$ jobs, and (2) the completion of a job's service when the system has exactly $n+1$ jobs. The steady-state flow *out* of an intermediate state n $(0 < n < n_{\max})$ is also made up of two components: (1) the completion of a job's service when the system has exactly n jobs, and (2) the arrival of a new job to the system when there are exactly n jobs in the system prior to the arrival event.

The resulting flow balance equation for state n is made up of the above four components. The mean arrival rate of jobs into the system is λ and the mean service rate of jobs when there is at least one job in the system is μ. The flow into state n occurs at the rate λ times the probability that the system is in state $n-1$ plus the rate μ times the probability that the system is in state $n+1$. Similarly, the flow out of state n occurs with rate $(\lambda+\mu)$ times the probability that the system is in state n. Thus, the steady-state flow-balance equation for an intermediate state n is

$$\lambda p_{n-1} + \mu p_{n+1} = (\lambda+\mu)p_n \text{ for } n = 1,\cdots,n_{\max} , \tag{3.1}$$

where the left-hand-side is the inflow and the right-hand-side is the outflow.

States 0 and $n_{\max}$ have different equations since some of the terms of the intermediate states equation are not valid for these boundary states. For example, the service rate is zero if there are no jobs in the system (state 0) nor can the system reside in state -1 so that an arrival event will put it into state 0. Also if the system is full (state $n_{\max}$), then no service from state $n_{\max}+1$ can occur and no new jobs are allowed to enter the system. The two special flow-balance equations (for states 0 and $n_{\max}$) are

$$\mu p_1 = \lambda p_0 \tag{3.2}$$

and

$$\lambda p_{n_{\max}-1} = \mu p_{n_{\max}} . \tag{3.3}$$

These three equations (namely, 3.1, 3.2, and 3.3) specify $n_{\max}+1$ equations connecting the state probabilities p_n. In addition, it is also known that the sum of these probabilities must add to one. Thus, there exists the additional equation, called the *norming equation*, written as

$$\sum_{n=0}^{n_{\max}} p_n = 1 \,. \tag{3.4}$$

It turns out that the system is over-specified; that is, Eqs. (3.1–3.4) contain more equations than unknowns. To solve the system, any one of the equations can be omitted *except* for the norming equation. (The reader is asked to consider this point further in Problem 3.6.) After (arbitrarily) eliminating one equation from the system comprised of (3.1–3.3), there will be a total of $n_{\max}+1$ linear equations in $n_{\max}+1$ unknowns from the system defined by (3.1–3.4).

Given the mean arrival rate λ, the mean service rate μ and a system limit of $n_{\max}$, the resulting $n_{\max}+1$ linear equations can be solved by standard numerical methods. If $n_{\max}$ is not large, the equations can be written explicitly and solved for the specified values of λ and μ. However, because the system (3.1–3.4) has a fairly simple structure, it can be also be solved in general by a recursive substitution scheme and a closed form solution obtained. Not all systems that we develop in this text will have a structure leading to a general solution, but when this can be accomplished, it is the preferred method since the values of the parameters λ, μ and $n_{\max}$ need not be specified and a parametric solution for all values (or acceptable ranges of these parameter values) is obtained when solving the general system. For illustrative purposes, the system (3.1–3.4) is solved by both methods.

Example 3.1. **Specific Solution.** Consider a facility with a single machine that is used to service only one type of job. The company policy is to limit the number of orders accepted at any one time to 3. The mean arrival rate of orders, λ, is 5 jobs per day, and the mean processing time for a job is 1/4 day (thus, the processing rate is $\mu = 4/\text{day}$). Both the processing and inter-arrival times are assumed to be exponentially distributed. These assumptions lead to the system of equations

$$\begin{aligned} 4p_1 - 5p_0 &= 0 \\ 5p_0 + 4p_2 - (5+4)p_1 &= 0 \\ 5p_1 + 4p_3 - (5+4)p_2 &= 0 \\ 5p_2 - 4p_3 &= 0 \\ p_0 + p_1 + p_2 + p_3 &= 1 \,. \end{aligned}$$

We ignore the fourth equation and only use the first three equations plus the fifth (norming) equation to obtain

$$(p_0, p_1, p_2, p_3) = (0.173, 0.217, 0.271, 0.339) \,.$$

(See the appendix for using Excel to solve linear systems of equations.) The number of lost jobs per hour (i.e., those arriving to a full system) is given by $\lambda p_3 = 5 \times 0.339 = 1.695$. The server is idle when the system is empty, so the percentage of server idle time is 17.3%. Because the system is at steady-state, the throughput is equal to the number of jobs that enter the system per unit time (those jobs that actually get into the system, called the effective arrival rate). Thus, throughput rate equals the arrival rate minus the loss rate; namely, 5 - 1.695 = 3.305 jobs/day. Note that

$$\begin{aligned} WIP &= E[N] = \sum np_n = 1 \times 0.217 + 2 \times 0.271 + 3 \times 0.339 = 1.776 \text{ jobs} , \\ CT &= WIP/th = WIP/(\lambda(1-p_3)) = 1.776/3.305 = 0.537 \text{ days} . \end{aligned}$$

□

Example 3.2. **General Solution.** To illustrate the more general solution approach, this system of equations is solved using the parameters rather than their actual values. The system to be solved is

$$\begin{aligned} \mu p_1 - \lambda p_0 &= 0 \\ \lambda p_0 + \mu p_2 - (\lambda + \mu) p_1 &= 0 \\ \lambda p_1 + \mu p_3 - (\lambda + \mu) p_2 &= 0 \\ \lambda p_2 - \mu p_3 &= 0 \\ p_0 + p_1 + p_2 + p_3 &= 1 . \end{aligned}$$

As before, the first three equations and the fifth equation will be used. The solution procedure is a two-step process. First, all variables are expressed in terms of p_0 by use of the first three equations. This is accomplished through a series of successive substitutions. Second, the value of p_0 is obtained by the use of the norming equation. Specifically, the first equation yields p_1 in terms of p_0 by

$$\begin{aligned} \mu p_1 &= \lambda p_0 \\ p_1 &= \frac{\lambda}{\mu} p_0 . \end{aligned}$$

The variable p_2 is obtained as a function of p_0 by substituting the expression for p_1 into the second equation as

$$\begin{aligned} \lambda p_0 + \mu p_2 &= (\lambda + \mu) p_1 \\ \mu p_2 &= (\lambda + \mu) p_1 - \lambda p_0 \\ p_2 &= (\lambda + \mu) \frac{\lambda}{\mu^2} p_0 - \frac{\lambda}{\mu} p_0 \\ p_2 &= \left(\frac{\lambda}{\mu} \right)^2 p_0 . \end{aligned}$$

Similarly, the third equation is used to obtain p_3 as a function of p_0 by substituting the expressions for the previously obtained p_1 and p_2; namely,

$$\begin{aligned} \lambda p_1 + \mu p_3 &= (\lambda + \mu) p_2 \\ p_3 &= (\lambda + \mu) \frac{\lambda^2}{\mu^3} p_0 - \left(\frac{\lambda}{\mu} \right)^2 p_0 \end{aligned}$$

$$p_3 = \left(\frac{\lambda}{\mu}\right)^3 p_0 \,.$$

The conclusion from the first step is that all probabilities are now in terms of p_0; namely,

$$p_1 = \left(\frac{\lambda}{\mu}\right) p_0, \qquad p_2 = \left(\frac{\lambda}{\mu}\right)^2 p_0, \qquad p_3 = \left(\frac{\lambda}{\mu}\right)^3 p_0 \,. \tag{3.5}$$

The final step is to substitute these expressions into the norming equation as follows:

$$\begin{aligned} 1 &= p_0 + p_1 + p_2 + p_3 \\ &= \left[1 + \frac{\lambda}{\mu} + \left(\frac{\lambda}{\mu}\right)^2 + \left(\frac{\lambda}{\mu}\right)^3\right] p_0 = 1 \end{aligned}$$

thus

$$p_0 = \left[1 + \frac{\lambda}{\mu} + \left(\frac{\lambda}{\mu}\right)^2 + \left(\frac{\lambda}{\mu}\right)^3\right]^{-1} \,. \tag{3.6}$$

From here we can develop the measures of $WIP = p_1 + 2p_2 + 3p_3$, $th = \lambda(p_0 + p_1 + p_2)$, and $CT = WIP/th$. □

Before moving to the remainder of the chapter, it is beneficial to formally define the effective arrival rate and comment on Little's Law. Whenever the system is finite, there is the possibility that the system will be full and arriving jobs will be lost; hence, the actual rate of jobs that enter the system, λ_e may not be the same as the arrival rate, λ.

Definition 3.1. The *effective arrival rate* for a system is the rate at which jobs enter the system. For a workstation with constant arrival rate, λ, and with a maximum number of jobs at the workstation limited to $n_{\max}$, the effective arrival rate is given by

$$\lambda_e = \lambda(1 - p_{n_{\max}})$$

where $p_{n_{\max}}$ is the probability that the workstation is full.

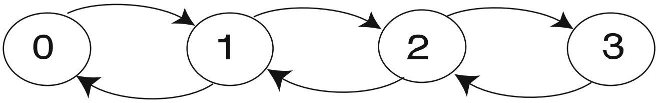
A system at steady-state will have its system throughput rate equal to the effective arrival rate; that is, $th = \lambda_e$, and the use of Little's Law (Property 2.1) must always use λ_e and not λ for the throughput.

- *Suggestion: Do Problem 3.1.*

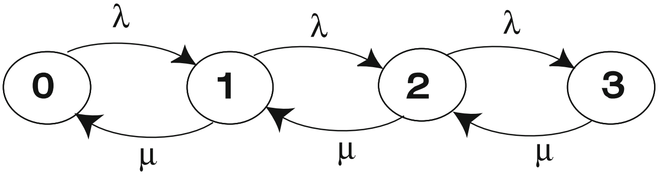
3.2 Diagram Method for Developing the Balance Equations

There is a relatively straightforward method for developing the balance equations for essentially any system in steady-state whose inter-arrival and service times are exponentially distributed. The approach is to start by listing all of the states as nodes in a network. For the single-server problem, a sequential listing is the best. As one develops an understanding of this approach, a suitable layout will be apparent. The node listing is

Now directional arcs are added to the network to represent possible flows between nodes (states). For instance, node 0 is connected to node 1 to represent the flow from state 0 to 1 when an arrival occurs and the system is in state 0. Similarly, node 1 is connected to node 0 to represent the flow when a service occurs with the system in state 1 (a service results in an empty system or state 0). States 1 and 2 are connected, with a directed arc from 1 to 2, by an arrival event while in state 1. Conversely, states 1 and 2 are connected by a service event while in state 2; thus, the directed arc is from 2 to 1. The same logic connects states 2 and 3. So the following directed network is obtained. Note that an arrival into the system cannot occur when the system is in state 3 (i.e., when the system is full).

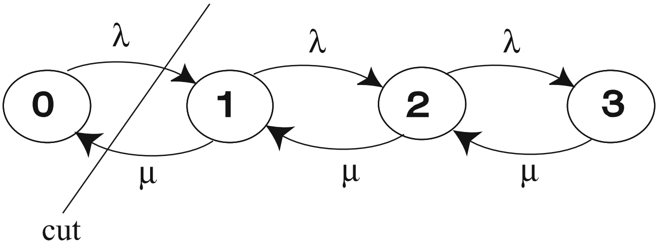


Now that the appropriately directed arc network of the system being modeled has been developed, the actual flow rates can be displayed on theses arcs. These rates are relatively straightforward to determine. Since the system has an arrival process that does not depend on the state of the system (excluding when it is full and so no arrivals can occur), the upward movements among the states all occur at a rate λ times the probability of being in that state, p_n. That is, the conditional arrival rate given that the system is in state n is λ and the net upward rate from state n is λp_n. The downward movements all occur when a service has been completed and these have rates that are μ times the probability of being in the particular state, p_n. Thus, the conditional service rate given that there is a job in the system to be serviced is μ. The resulting downward rates from state n is μp_n. The similarity of the service rates is again due to the assumption about the system. There is a single server and the service rate is independent of the state of the system. That is, the server works at the same rate without regard to the number of jobs in the queue. The standard method of graphically depicting the flow between states is to label the flow (arrows) with the conditional rates for that state.



This completed directed network can now be used to derive the steady-state balance equations previously analyzed. The logic goes as follows. Partition the nodes into two subsets of nodes, then establish values for the appropriate steady-state probabilities to balance the flow between the two subsets. Partitions are redrawn at $n-1$ different locations to obtain $n-1$ equations. These balance equations are then combined with the norming equations to yield a system of equations similar to the system of (3.1–3.4).

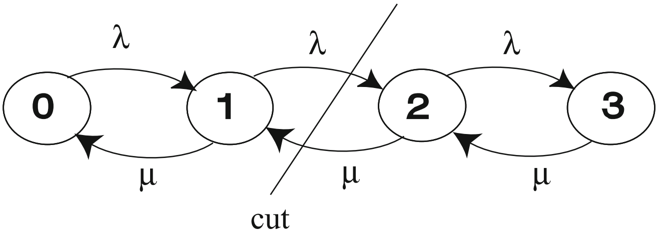
Consider the two subsets of nodes formed when a cut is made between nodes 0 and 1 as is illustrated below.



The balance equation associated with this initial cut is

$$\lambda p_0 = \mu p_1 \ .$$

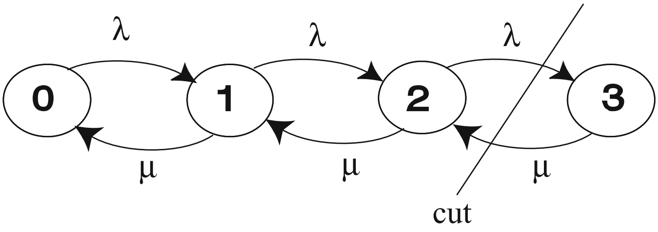
The second cut is between states 1 and 2.



The resulting balance equation associated with this cut is

$$\lambda p_1 = \mu p_2 \ .$$

The final cut is between states 2 and 3 as depicted below.



Thus the third balance equation is

$$\lambda p_2 = \mu p_3 \, .$$

These three-balance equations and the norming equation yield another representation for our modeled system as

$$\begin{aligned} \lambda p_0 &= \mu p_1 \\ \lambda p_1 &= \mu p_2 \\ \lambda p_2 &= \mu p_3 \\ \sum_{n=0}^{3} p_n &= 1. \end{aligned} \tag{3.7}$$

The system (3.7) obviously has the same relationships between the probabilities as (3.1–3.4); however, there is usually less work in obtaining this system using the flow balance approach. Successive substitution can then be used with (3.7) to obtain (3.5) and the norming equation yields the value for p_0 as was accomplished with (3.6).

Another subset partition that leads to the same system of equations is obtained by separating each node into its own singleton subset. The other subset contains all the other nodes of the network. The associated balance equations for each node arise when considering the input arcs to the node and balancing those rates with the outflow arcs. The development of this set of balance equations parallels the discussion in Sect. 3.1 and is left as an exercise for the reader (Problem 3.2).

The labeled directed arc network and partitioning method is a powerful methodology for deriving balance equations for queueing systems with exponentially distributed inter-arrival and service times. It is a useful method that helps one visualize the relationships in the system and keep track of the associated derived balance equations as they are being developed. Extensive use is made in this textbook of the labeled-directed arc-diagram approach for studying factory models.

3.3 Model Shorthand Notation

The models studied to this point all assumed exponentially distributed inter-arrival and service mechanisms. There is a notational shorthand due to Kendall [6] for characterizing queueing models that is quite useful. With essentially one word, the model assumptions and system behavior can be summarized. This notation, or variants of it, frequently appear in the queueing theory literature, particularly in paper titles. This system does not encompass all model variations imaginable, but it does present a great deal of information about the system in concise notation. The Kendall notation for queues is a list of characters each separated by a "/". The first element in the list specifies the inter-arrival time distribution assumption. The symbol M (for Markovian) depicts exponentially distributed times. The second element in the list denotes the service time distribution assumption. The third element in the list specifies the number of servers and the fourth element is the maximum number of jobs allowed in the system at one time. An optional fifth element specifies the assumption for the queueing discipline. The general form for Kendall's notation is

$$\left(\begin{matrix}\text{arrival}\\ \text{process}\end{matrix} \Big/ \begin{matrix}\text{service}\\ \text{process}\end{matrix} \Big/ \begin{matrix}\text{number}\\ \text{of servers}\end{matrix} \Big/ \begin{matrix}\text{maximum}\\ \text{possible}\\ \text{in system}\end{matrix} \Big/ \begin{matrix}\text{queue}\\ \text{discipline}\end{matrix} \right)$$

with Table 3.1 providing a summary of the commonly used abbreviations. Thus, the example queueing system just studied is denoted as an $M/M/1/3$ system. The two server model of Problem 3.3 is denoted by $M/M/2/3$. If the system has no effective limit on the number of jobs allowed, then the fourth parameter would be infinity. Most often the fourth parameter is omitted when it is not finite, so that such a model would often be written as $M/M/1$ instead of $M/M/1/\infty$.

Table 3.1 Queueing symbols used with Kendall's notation

Symbols	Explanation
M	Exponential (Markov) inter-arrival or service time
D	Deterministic inter-arrival or service time
E_k	Erlang type k inter-arrival or service time
G	General inter-arrival or service time
$1,2,\cdots,\infty$	Number of parallel servers or capacity
FIFO	First in, first out queue discipline
LIFO	Last in, first out queue discipline
SIRO	Service in random order
PRI	Priority queue discipline
GD	General queue discipline

As the need arises, other parameter designations will be defined such as D for a deterministic time and G for a general distribution. To illustrate this notation, some of the most fundamental results needed for studying factory performance are the $G/G/1$ model approximations that are taken up at the end of this chapter.

- *Suggestion: Do Problems 3.2–3.6.*

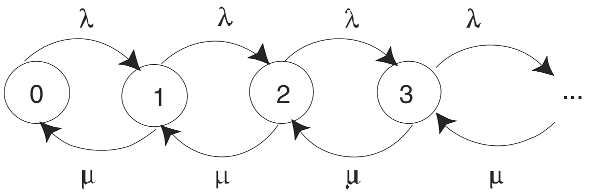
3.4 An Infinite Capacity Model ($M/M/1$)

The finite capacity limitation on the $M/M/1/3$ model just studied is easily dropped, and the removal of this limitation has some interesting consequences. First note that the system of equations derived above (i.e., with a finite capacity) has a solution regardless of the relationship between the arrival rate and the system service rate. If the arrival rate of jobs to the system is larger than the system service capacity, the system is full a relatively high proportion of the time. This in turn leads to more jobs being turned away because of the full system. In fact, the effective arrival rate (those jobs getting into the system) will necessarily be less than the system's service capacity. Let's consider a few cases for the above example that illustrate this point. Suppose that the mean arrival rate is equal to the mean service rate, $\lambda = \mu$ for the $M/M/1/3$ system. With $\lambda = \mu$, each probability is equal so that $p_0 = \cdots = p_3 = 1/4$. The effective arrival rate is, thus, given by $\lambda_e = \lambda(1 - p_3) = (3/4)\lambda < \mu$. If the mean arrival rate is twice the mean service rate, $\lambda = 2\mu$, then the effective arrival rate becomes $\lambda_e = (7/15)\lambda < \mu$. For a mean arrival rate that is three times the mean service rate, $\lambda = 3\mu$, the effective arrival rate becomes $\lambda_e = (13/40)\lambda < \mu$. Note that as the ratio of λ/μ becomes larger, the effective arrival rate approaches the inverse of this ratio but never reaches it. The reader is asked to compute these effective rates in Problem 3.5.

One of the lessons to be learned from the finite capacity model is that these systems have a built-in mechanism to adjust the arrival rate (called the effective arrival rate) to a level that can be handled by the system service capacity. If a system that has no realistic limit on the number of jobs allowed is considered, then mathematically, these systems can be put in a situation where the mean arrival rate exceeds the mean service rate and no steady-state exists. It is unreasonable to assume that jobs continue to arrive when there is essentially an infinite queue and the expected cycle time is also infinite. Of course, one would like to operate well below the blowup point with respect to the arrival and service capacity ratio. The analyses of the unlimited queueing models result in conditions that establish the existence of the steady-state behavior for these models.

The formulation of the unlimited-jobs system is very analogous to the finite capacity model formulation. The solution procedure is considerably different in that an infinite number of states exist and, correspondingly, an infinite number of descriptive equations result. Thus, standard numerical solutions for linear equations cannot be used. One is forced to solve these systems in a fashion analogous to the parametric solution approach illustrated for the finite capacity systems. This method is essentially substitution and formulation of a recursive relationship for the general solution structure.

The set of equations for the $M/M/1$ system is the same as the equations for the finite system capacity case except that the system does not have a final equation. Thus, an infinite system of equations exists. The diagram for this system is depicted below.



Using the cut partitioning method for obtaining the system of equations needed in defining the steady-state probabilities, the following is obtained:

$$\begin{aligned}
\lambda p_0 &= \mu p_1 \\
\lambda p_1 &= \mu p_2 \\
\lambda p_2 &= \mu p_3 \\
&\vdots \\
\lambda p_n &= \mu p_{n+1} \\
&\vdots \\
\sum_{n=0}^{\infty} p_n &= 1\,.
\end{aligned}$$

The above system can be rewritten to obtain the following equivalent system.

$$\begin{aligned}
p_1 &= \tfrac{\lambda}{\mu} p_0 \\
p_2 &= \tfrac{\lambda}{\mu} p_1 \\
p_3 &= \tfrac{\lambda}{\mu} p_2 \\
&\vdots \\
p_n &= \tfrac{\lambda}{\mu} p_{n-1} \\
&\vdots
\end{aligned}$$

Using a successive substitution procedure, each p_n term can be written as a function of p_0 to obtain

$$p_n = \left(\frac{\lambda}{\mu}\right)^n p_0 \ \text{ for } n = 0, 1, \cdots\,. \tag{3.8}$$

The final step is to substitute (3.8) into the norming equation yielding

$$p_0 + \left(\frac{\lambda}{\mu}\right) p_0 + \left(\frac{\lambda}{\mu}\right)^2 p_0 + \cdots + \left(\frac{\lambda}{\mu}\right)^n p_0 + \cdots = 1\,,$$

which can be solved to obtain an expression for p_0 as

$$p_0 = \frac{1}{\left(1 + \frac{\lambda}{\mu} + \left(\frac{\lambda}{\mu}\right)^2 + \cdots + \left(\frac{\lambda}{\mu}\right)^n + \cdots\right)}\,.$$

The denominator is a geometric series[1] that has a finite value if $\lambda/\mu < 1$. Under the condition that $\lambda < \mu$, this series sums to

$$p_0 = 1 - \frac{\lambda}{\mu}\,, \tag{3.9}$$

[1] The geometric series is $\sum_{n=0}^{\infty} r^n = 1/(1-r)$ for $|r| < 1$. Taking the derivative of both sides of the geometric series yields another useful result, $\sum_{n=1}^{\infty} n r^{n-1} = 1/(1-r)^2$ for $|r| < 1$.

and the general solution to the steady-state probabilities is (given that $\lambda/\mu < 1$)

$$p_n = \left(1 - \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^n \text{ for } n = 0, 1, \cdots . \tag{3.10}$$

The throughput rate per unit time for this system is λ. (The reader is asked to develop this result in Problem 3.10.) The utilization factor u for the server is obtained from

$$u = 0p_0 + 1\left(\sum_{n=1}^{\infty} p_n\right) = 1 - p_0 = 1 - \left(1 - \frac{\lambda}{\mu}\right) = \frac{\lambda}{\mu} .$$

The expected number of jobs in the system in steady-state is obtained by using the derivative of the geometric series as follows:

$$\begin{aligned} WIP_s = E[N] &= \sum_{n=0}^{\infty} np_n = \sum_{n=0}^{\infty} n\left(1 - \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^n \\ &= \left(1 - \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)\sum_{n=1}^{\infty} n\left(\frac{\lambda}{\mu}\right)^{n-1} \\ &= \left(1 - \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)\left(\frac{1}{1 - \frac{\lambda}{\mu}}\right)^2 \\ &= \frac{\left(1 - \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)}{\left(1 - \frac{\lambda}{\mu}\right)^2} = \frac{\frac{\lambda}{\mu}}{\left(1 - \frac{\lambda}{\mu}\right)} = \frac{u}{1-u} \end{aligned} \tag{3.11}$$

where N is a random variable denoting the number of jobs in the system. Using Little's Law (Property 2.1), the expected time in system (the cycle time) CT_s is given by

$$CT_s = \frac{WIP_s}{\lambda} = \frac{1}{\lambda}\frac{\frac{\lambda}{\mu}}{(1 - \frac{\lambda}{\mu})} = \frac{1}{\mu - \lambda} . \tag{3.12}$$

Example 3.3. Consider a single server system with exponentially-distributed inter-arrival times and exponentially-distributed service times (thus, this is an $M/M/1$ system). If 4 jobs per hour arrive for service ($\lambda = 4$) and the mean service time is 1/5 hour ($\mu = 5$), then the utilization factor u ($u = \lambda/\mu$) equals 0.8. The expected number of jobs in the system, WIP_s from (3.11) is

$$WIP_s = \frac{0.8}{(1 - 0.8)} = 4 .$$

The cycle time in the system, CT_s, is given by (3.12) and is

$$CT_s = \frac{1}{5-4} = 1 \text{ hr} .$$

The cycle time in the system is the sum of the cycle time in the queue plus the service time. Hence, $CT_q = 1 - 0.2 = 0.8$ hr. The probability that the server is idle, of course, equals the probability that the system is empty, p_0. This probability is

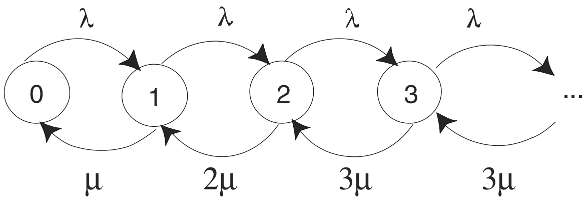
$$p_0 = 1 - \frac{\lambda}{\mu} = 0.2 \,.$$

The steady-state probability that there are n jobs in the system is given by

$$p_n = 0.2 \times 0.8^n \quad \text{for } n = 0, 1, \cdots \,.$$

□

A workstation may consist of multiple machines; however, in most models, server or machine distinctions are not usually made. That is, if there are two machines available, then for ease of modeling it is usually assumed that these are identical machines and that jobs are not split, but processed completely on one machine. Under the assumption of identical machines, if one machine operates at a rate of μ, then n machines operate at a rate of $n\mu$, and the state diagram must be adjusted accordingly. For example, suppose a workstation has three machines, then the service rate when two machines are busy is 2μ and whenever all machines are busy the service rate is 3μ; thus, the rate diagram is as below.



- *Suggestion: Do Problems 3.7–3.14.*

3.5 Multiple Server Systems with Non-identical Service Rates

The assumptions of identical machines may not be accurate, and if there is a significant difference in the operating characteristics of the machines associated with a single workstation, more complex models will result. To provide some exposure to the complexity involved in modeling non-identical machines within a single workstation, a simple non-identical servers model is considered and the associated defining equations for the steady-state probabilities are developed. The structure of this system is that it has two non-identical servers and a limit of four jobs in the system at one time. Inter-arrival and service times are all assumed to be exponentially distributed with a mean arrival rate of λ and mean service rates of μ and γ for the two distinct machines. Let $\gamma < \mu$, so that the μ machine is faster and, therefore,

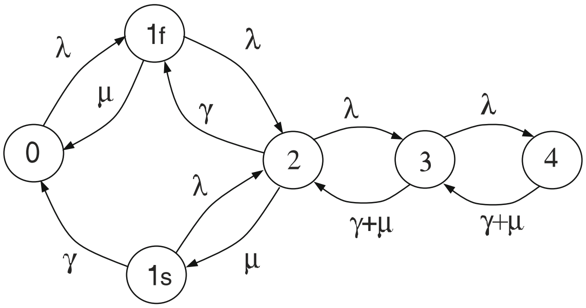


Fig. 3.1 State diagram for an $M/M/2/4$ system with non-identical servers, where μ denotes the rate of the faster machine and γ is the rate of the slower machine

preferred. The system operating policy is such that when the system is empty, an arriving job is always assigned to the faster machine. If a job arrives to the system and finds that only one machine is busy, the job is assigned to the idle machine immediately regardless of the speed of the machine or how long the other machine has been busy. This same logic is applied when a machine completes service and there is a queue of waiting jobs. The next job in line is immediately allocated to the idle machine; thus, machines can never be idle when there is a queue of waiting jobs. A final assumption is that once a job is assigned to a machine for processing, it remains on that machine until its processing is complete. Hence, jobs cannot be split and processed on both machines nor can a job be moved from the slower to the faster machine.

As before, $n_{\max}$ is the maximum number of jobs allowed in the system (here $n_{\max} = 4$) so that there will be a total of $n_{\max} + 2$ possible states for this model. In the identical server model, there were $n_{\max} + 1$ possible states. The extra state arises because we must know which machine is busy when there is only one job at the workstation in order to know the service rate associated with the job in process. When there are two or more jobs in the system, both machines are busy and no distinction about the state needs to be made. Denoting the state (i.e., the number of jobs at the workstation) by n, one possible state space is the set $\{0, 1\text{f}, 1\text{s}, 2, 3, 4\}$, where $n = 1\text{f}$ indicates that one job is in the system and that job is being processed on the fast machine and $n = 1\text{s}$ indicates that one job is in the system and is being processed on the slow machine. Given these operational rules and notation, the state diagram of this system is displayed in Fig. 3.1.

The transition rates shown in the diagram of Fig. 3.1 are explained as follows. In any state (other than the maximum), the arrival of a job takes the system to the next higher state number. Both states 1f and 1s move to state 2 with a job arrival. An arrival to an empty system moves the state from 0 to state 1f because of the assumption that the faster machine is preferred. From state 2, the next state depends on which machine finishes first. If the faster machine finishes before the slower machine, the system has one job remaining and this job continues being processed on the slower machine; thus, the system ends up in state 1s. This occurs with rate

μp_2. With similar reasoning, it should be clear that if the slower machine completes its processing first, the system transitions to state 1f. The transition from 2 to 1f occurs at a rate of γp_2. Notice that the downward movement from state 2 occurs with rate $(\mu+\gamma)p_2$. Downward movement from state 3 to state 2 occurs with rate $(\mu+\gamma)p_3$ and, similarly, from state 4 to state 3 with rate $(\mu+\gamma)p_4$.

The defining equations for the steady-state probabilities are determined by taking cuts between states. A slight problem exists with defining a cut between states due to the multiplicity of state 1 (i.e., 1f and 1s). The general idea of a cut is to isolate a set of states from the remaining states. In a serial system this cut process is easily defined and leads to the number of equations necessary for uniquely defining the probabilities when combined with the norming equation. The diagram (Fig. 3.1) for this non-identical server system is non-serial and thus there are several more possibilities for the cuts. The actual cuts that are used in the final analysis must be chosen wisely so that all probabilities are defined. For our set, we shall establish five cuts such that a cut is placed immediately to the right of each node subset contained within the following set:

$$\{\ \{0\},\{0,1\mathrm{f}\},\{0,1\mathrm{f},1\mathrm{s}\},\{0,1\mathrm{f},1\mathrm{s},2\},\{0,1\mathrm{f},1\mathrm{s},2,3\}\ \}$$

thus producing the following five equations:

$$\begin{aligned}\lambda p_0 &= \mu p_{1\mathrm{f}} + \gamma p_{1\mathrm{s}}\\ \lambda p_{1\mathrm{f}} &= \gamma p_2 + \gamma p_{1\mathrm{s}}\\ \lambda p_{1\mathrm{f}} + \lambda p_{1\mathrm{s}} &= (\gamma+\mu)p_2\\ \lambda p_2 &= (\gamma+\mu)p_3\\ \lambda p_3 &= (\gamma+\mu)p_4\,.\end{aligned}\tag{3.13}$$

These equations, plus the norming equation,

$$p_0 + p_{1\mathrm{f}} + p_{1\mathrm{s}} + p_2 + p_3 + p_4 = 1$$

are six equations that can be solved to obtain the steady-state probabilities for this system.

Example 3.4. An overhaul facility for helicopters is open 24 hours a day, seven days a week and helicopters arrive to the facility at an average rate of 3 per day according to a Poisson process (i.e., exponential inter-arrival times). One of the areas within the facility is for degreasing one of the major components. There is only room in the facility for 4 jobs at any one time and there are two machines that do the degreasing. The newer of the two degreasing machines takes an average of 8 hours to complete the degreasing and the older machine takes 12 hours for the degreasing operation. Because of the large variability in helicopter conditions, all times are exponentially distributed. Thus, we have $\lambda = 3$ per day, $\mu = 3$ per day, and $\gamma = 2$ per day. The system of equations given by (3.13) become

$$\begin{aligned}
3p_0 - 3p_{1\text{f}} - 2p_{1\text{s}} &= 0\\
3p_{1\text{f}} - 2p_2 - 2p_{1\text{s}} &= 0\\
3p_{1\text{f}} + 3p_{1\text{s}} - 5p_2 &= 0\\
3p_2 - 5p_3 &= 0\\
3p_3 - 5p_4 &= 0\\
p_0 + p_{1\text{f}} + p_{1\text{s}} + p_2 + p_3 + p_4 &= 1 \,.
\end{aligned}$$

The solution to this system of equations is

$$p_0 = 0.288,\ p_{1\text{f}} = 0.209,\ p_{1\text{s}} = 0.118,\ p_2 = 0.196,\ p_3 = 0.118,\ p_4 = 0.071 \,.$$

The average number in the system is obtained by using the definition of an expected value; namely,

$$WIP_s = p_{1\text{f}} + p_{1\text{s}} + 2p_2 + 3p_3 + 4p_4 = 1.356$$

and the average number in the queue is obtained similarly,

$$WIP_q = p_3 + 2p_4 = 0.259 \,.$$

Note that for the average number in the queue, p_3 is multiplied by 1 because when there are 3 in the system, there is only 1 in the queue. Also, p_4 is multiplied by 2 because when there are 4 in the system, there are 2 in the queue. Average cycle times are obtained through Little's Law as

$$\begin{aligned}
CT_s &= \frac{WIP_s}{\lambda_e} = \frac{1.356}{3\times(1-0.071)} = 0.486 \text{ day}\\
CT_q &= \frac{WIP_q}{\lambda_e} = \frac{0.259}{3\times(1-0.071)} = 0.093 \text{ day} \,.
\end{aligned}$$

A couple of other measures that are sometimes desired by management are the number of busy processors (i.e., degreasers) and their utilization. The expected number of busy servers, $E[BS]$, is 1.097, and is obtained as

$$E[BS] = 1p_{1\text{f}} + 1p_{1\text{s}} + 2p_2 + 2P_3 + 2p_4 = 1.097 \,.$$

The system utilization factor u is the expected number of busy servers divided by the number of machines available

$$u = \frac{E[BS]}{2} = 0.5485 = 54.85\% \,.$$

Our final calculation is to obtain the average time needed for degreasing. Because of the preference given to using the faster machine, we would expect the average time to be closer to 8 hours than to 12 hours. To get an exact value, we take advantage of the fact that the time in the system equals the time in the queue plus service time (Eq. 2.1); thus

$$E[T] = CT_s - CT_q = 0.486 - 0.093 = 0.393 \text{ days} = 9.4 \text{ hr} .$$

□

- *Suggestion: Do Problems 3.15–3.20.*

3.6 Using Exponentials to Approximate General Times

The exponential distribution is an extremely powerful modeling tool because of its lack of memory (Eq. 1.16 and Problem 1.24). That is, the rate of completion of the process does not change with elapsed time. So for systems with exponential times, it is not necessary to keep track of the elapsed inter-arrival time nor the elapsed service time. This allows the steady-state modeling approach to be used. To model more general systems, one fruitful approach is to approximate the general times by combinations of exponentials. Then the exponential rate modeling approach can still be applied by developing more complex state representations of the system.

The Erlang-k distribution (see p. 18 for a review of the Erlang) provides an excellent distribution to use for the expanded state modeling approach. The Erlang-k distribution is the sum of k independent and identical exponential distributions, so that it can be modeled as a serial k-node system, with each node referring to identical exponentials. Since the Erlang-k has a squared coefficient of variation given by $C^2 = 1/k$, it also allows modeling of processes that have less variation than the exponential distribution.

3.6.1 Erlang Processing Times

To illustrate the expanded state modeling approach, consider a single server system with exponential inter-arrival times having a mean rate λ and a processing time that is described by an Erlang-2 distribution with mean rate μ and thus mean time $1/\mu$. This Erlang-2 distribution will be modeled using two exponential nodes (or *phases*), where each node has a mean rate of 2μ. Since rates and times are reciprocals, the mean time spent in each node is $1/(2\mu)$. This gives the total time spent in the two nodes as $1/\mu$ (i.e., the sum of the two means) which is equal to the average time of the Erlang-2 processing time distribution. To further simplify this example, the number of jobs allowed into the system will be limited to three. Thus, we are interested in analyzing an $M/E_2/1/3$ system.

The idea of the expanded state space approach is to represent the non-exponential process by more than one node, where each individual node is exponential. Therefore, the service process will have two nodes representing the two phases of the Erlang-2 distribution. When a job begins its processing, it enters the node represent-

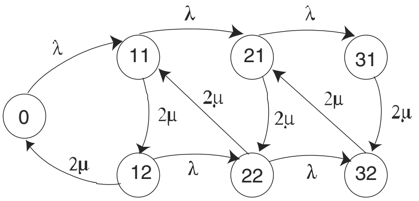


Fig. 3.2 Diagram for an $M/E_2/1/3$ model where the state (n,i) indicates that there are n jobs in the system with the i^{th} service phase busy

ing phase 1 and stays in phase 1 for an exponential length of time. When the job has been completed its phase 1 service, the job moves to the node representing phase 2. As long as the job is in either phase, it is considered to be continuing its processing and a new job is not allowed into service. When the job is finished with phase 2, it is considered to be finished with its processing and it leaves the system, and at this point in time, a new job can enter phase 1 to begin its service. A convenient representation for the state space is to use ordered pairs. In other words, (n,i) denotes a state of the system, where n is the number of jobs in the system and i is the service phase being occupied by the job being processed. The $M/E_2/1/3$ state diagram is displayed in Fig. 3.2.

There are $2n_{\max}+1$ states, where $n_{\max}$ is the maximum number of jobs allowed into the system (here $n_{\max}=3$). To obtain the steady-state probabilities for this system, six cuts are placed so that the following node sets are isolated on one side of the cut

$$\{\ \{0\},\{0,(1,2)\},\{0,(1,1)\},\{0,(1,1),(1,2)\},\{(3,1),(3,2)\},\{(3,2)\}\ \}$$

which together with the norming equation yields the following system of equations,

$$\begin{aligned}
\lambda p_0 - 2\mu p_{12} &= 0\\
\lambda p_0 + \lambda p_{12} - 2\mu p_{11} &= 0\\
(\lambda+2\mu)p_{11} - 2\mu p_{12} - 2\mu p_{22} &= 0\\
\lambda p_{11} + \lambda p_{12} - 2\mu p_{22} &= 0\\
\lambda p_{21} + \lambda p_{22} - 2\mu p_{32} &= 0\\
\lambda p_{22} + 2\mu p_{31} - 2\mu p_{32} &= 0\\
p_0 + p_{11} + p_{12} + p_{21} + p_{22} + p_{31} + p_{32} &= 1\ .
\end{aligned}$$

The performance measures of work-in-process, cycle time and throughput are computed from

$$\begin{aligned}
WIP_s &= \sum_{n=1}^{4} n(p_{n1}+p_{n2})\\
th &= \lambda_e = \lambda(1-p_{31}-p_{32})\\
CT_s &= WIP_s/\lambda_e\ .
\end{aligned}$$

3.6.2 Erlang Inter-Arrival Times

If the inter-arrival process is an Erlang distribution then the state-space scheme is slightly different from that used for Erlang service. The same concept of breaking the service process into phases is used for the arrival process; however, the state space will be slightly different. We illustrate the expanded state space process applied to arrivals by assuming an Erlang-2 inter-arrival time process. The arrivals will be processed one-at-a-time at a single workstation with exponentially distributed service times with a limit of three jobs in the system, in other words, we consider an $E_2/M/1/3$ system.

Conceptually, an arriving job is always in one of two phases, and each phase has a mean rate of 2λ or a mean sojourn time of $1/(2\lambda)$. As long as a job is in one of the arrival phases, it is not yet considered part of the system. The arriving job begins in phase 1. After an exponentially distributed length of time, the job transitions to phase 2. After another exponential length of time, two events occur simultaneously: the job leaves phase 2 and enters the system and another jobs enters phase 1. (Note that for a model of phased arrivals, one of the arrival phases is always occupied and the other phases are empty.)

The slight difference in the state space for the Erlang inter-arrival time model versus the Erlang service time model occurs due to the situation that the arrival process has two phases regardless of the number of jobs in the system. So when the system is empty, there are still two phases that the arriving job must complete before it becomes an active job attempting to enter the system. The state-space notation used is (i,n) where as before i is the phase and n is the number of jobs in the system. Note that the order has been reversed from the Erlang service model to help keep in mind that the phases are for the arrival process. The states needed to model the $E_2/M/1/3$ system are: {(1,0), (2,0), (1,1), (2,1), (1,2), (2,2), (1,3), (2,3)}. The diagram of this model is given in Fig. 3.3. Note also that there is a different situation for blocked jobs for this model. A job is not blocked until it arrives to a full system which occurs from state (2,3) with rate 2λ. Then the arrival process starts over in state (1,3) rather than staying at state (2,3). That is, the arriving job is rejected and the arrival process starts over at state (1,3) for the next job creation. Thus, there is an arc between (2,3) and (1,3) with rate 2λ in Fig. 3.3 to represent this transition.

Instead of using cuts to derive the equations of state, we use the single-node isolation method for generating the equations that define the steady-state probabilities. The following system of equations (all eight equations are given but only seven are used since the norming equation is also required) are generated for the states in the order that they appear in the above state list.

$$\begin{aligned}
2\lambda p_{10} &= \mu p_{11} \\
2\lambda p_{20} &= 2\lambda p_{10} + \mu p_{21} \\
(2\lambda + \mu) p_{11} &= 2\lambda p_{20} + \mu p_{12} \\
(2\lambda + \mu) p_{21} &= 2\lambda p_{11} + \mu p_{22}
\end{aligned}$$

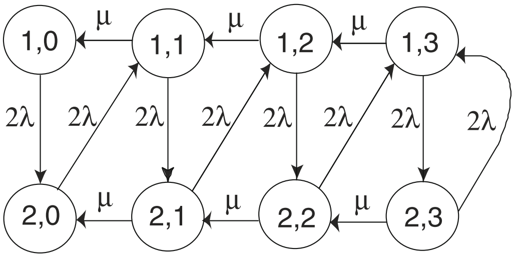


Fig. 3.3 Diagram for an $E_2/M/1/3$ model where the state (i,n) indicates that the arrival process is in phase i and there are n total jobs in the system

$$\begin{aligned}(2\lambda+\mu)p_{12} &= 2\lambda p_{21}+\mu p_{13}\\(2\lambda+\mu)p_{22} &= 2\lambda p_{12}+\mu p_{23}\\(2\lambda+\mu)p_{13} &= 2\lambda p_{22}+2\lambda p_{23}\\(2\lambda+\mu)p_{23} &= 2\lambda p_{13}\end{aligned}$$

and

$$p_{10}+p_{20}+p_{11}+p_{21}+p_{12}+p_{22}+p_{13}+p_{23}=1\ .$$

Example 3.5. Since this system consists of only 8 unknowns, it is easily solved using the matrix formulas in Excel (see the appendix to this chapter). Let $\lambda=5$ jobs/hr and $\mu=5$ jobs/hr, and the solution to the $E_2/M/1/3$ system of equations is

$$\begin{aligned}p_{10} &= 0.0687\ , & p_{20} &= 0.1358\ ,\\p_{11} &= 0.1374\ , & p_{21} &= 0.1342\ ,\\p_{12} &= 0.1406\ , & p_{22} &= 0.1278\ ,\\p_{13} &= 0.1534, & p_{23} &= 0.1022\ .\end{aligned}$$

Some of the system performance measures are

$$\begin{aligned}WIP_s &= 0(p_{10}+p_{20})+1(p_{11}+p_{21})+2(p_{12}+p_{22})+3(p_{13}+p_{23})=1.5751\\u &= p_{11}+p_{21}+p_{12}+p_{22}+p_{13}+p_{23}=1-(p_{10}+p_{20})=79.55\%\\th &= \lambda_e=\lambda-2\lambda p_{23}=\mu\times u=3.978\text{ jobs/hr}\\CT_s &= WIP_s/th=0.3960\text{ hr}\ .\end{aligned}$$

Notice that the throughput can be calculated in a couple of different but equivalent ways. The expression $\lambda-2\lambda p_{23}$ arrises by observing that arrivals are blocked from entering the system whenever the system is in the $(2,3)$ state and then the rate at which jobs leave state $(2,3)$ and try to enter the system is 2λ. Alternately, the throughput can be determined by multiplying the service rate μ times the probability that the server is busy, i.e., the utilization. □

- *Suggestion: Do Problems 3.21–3.24, and 3.33–3.36.*

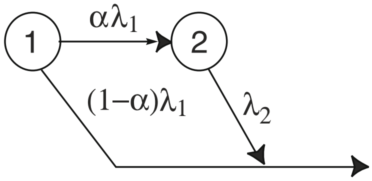


Fig. 3.4 A generalized Erlang with two phases, where the first phase always occurs and has a mean rate λ_1 and the second phase occurs with probability α and has a mean rate λ_2

3.6.3 Phased Inter-arrival and Processing Times

The improved modeling generality gained from the phased-service time model is frequently worth the notational inconvenience. For a phased-service time model, the state space is expanded essentially by a multiple of the number of phases. The state space for an $M/M/1/3$ system has four states $(n_{\max}+1)$, while its extension to the $M/E_2/1/3$ system has seven states $(2n_{\max}+1)$. The inter-arrival time process can also be broken into phases at the same time that the service times have phases to allow for even greater modeling flexibility, and the phases can be structured so as to be more general than the standard Erlang model. To illustrate the approach, the previous $M/E_2/1/3$ model is extended in this section to have a generalized Erlang-2 arrival process. There are two generalizations in the Erlang process that allow for a broader range of squared coefficients of variation, C^2, values while maintaining the essential exponential nature of individual nodes. The first generalization is to allow for non-identical phases and second is to give a probability that the process is complete at the end of each phase. Such a phased process is called a Generalized Erlang, GE, or a Coxian distribution. A GE with two phases is diagramed in Fig. 3.4.

A two-phase GE will be denoted by GE_2. Thus, the system of interest is an $GE_2/E_2/1/3$ model. The purpose of illustrating this generalization is to develop modeling skills that have more flexibility in the range of inter-arrival and service time distributions that can be studied. The distribution resulting from the GE_2 process illustrated in Fig. 3.4 can result in a squared coefficient of variation C^2 in the range $[0.5,\infty)$. Thus, the parameters of an GE_2 distribution can be selected to fit any finite mean and C^2 values needed, given that $C^2 \geq 1/2$. Notice that we have three parameters for the GE_2 distribution; namely, λ_1, λ_2, and α. It is possible to fix those three parameters to match a given mean, variance, and skewness for a distribution provided the skewness coefficient is not too large [2, p. 53]. However, it is more common to have only the mean and variance for a distribution. Parametric values for the GE_2 distribution have been suggested by Altiok [2, p. 54–56] when fitting the parameters to two moments. These are

$$\lambda_1 = \frac{2}{E[X]}\ , \quad \lambda_2 = \frac{1}{E[X]C^2[X]}\ , \quad \alpha = \frac{1}{2C^2[X]} \quad \text{for } C^2[X] > 1\ ; \tag{3.14}$$

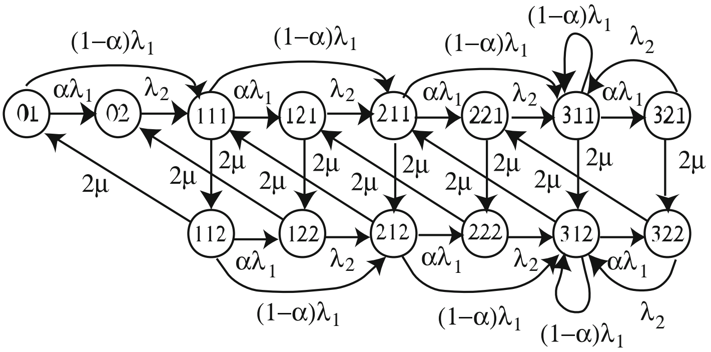


Fig. 3.5 State diagram for an $GE_2/E_2/1/3$ model, where a (n,i,j) indicates that there are n jobs in the system with one job in arrival phase i and one job is service phase j

$$\lambda_1 = \frac{1}{E[X]C^2[X]}\,,\quad \lambda_2 = \frac{2}{E[X]}\,,\quad \alpha = 2(1 - C^2[X]) \ \text{ for } \frac{1}{2} \le C^2[X] \le 1\,. \tag{3.15}$$

Note that matching two parameters of a distribution does not always characterize the distribution. Some distributions require three or more parameters for proper characterization, while the exponential distribution only requires one parameter (the mean rate λ or mean time $1/\lambda$).

Modeling with the GE_2 distribution causes these systems to quickly become quite complex. The $GE_2/E_2/1/3$ model, illustrated in Fig. 3.5, has 14 states, two states for each of the proceeding $M/E_2/1/3$ system states including the 0 state. The system empty state, state 0, now must be expanded so that the phase of the arriving job is represented. As one can readily see from the state diagram (Fig. 3.5) for this system, exponential-based generalizations for system times can be accomplished; however, these generalizations yield complex, and often intractable, models. The next section develops another approach for approximating general system time distributions (inter-arrival and service times).

- *Suggestion: Do Problems 3.25–3.29.*

3.7 Single Server Model Approximations

There are a variety of single facility generalizations that are standard in the queueing literature. Our concern is mainly with the assumptions regarding the inter-arrival and service time distributions. To use these models in a factory setting, more general assumptions on these distributions are needed. Rather than giving the general $G/G/1$ approximation model directly, a more circumspect route is taken that, hopefully, illuminates why and where the approximation arose. The model considered next is the exact result for the $M/G/1$ queue, that in a proper form, suggests the structure of the general approximation result.

3.7.1 General Service Distributions

Consider a single-server system with exponential inter-arrival times, with mean rate λ, and a general service time distribution having mean time $1/\mu$ and variance σ_s^2. The state-diagram approach can no longer be used to develop equations that define the steady-state probabilities since these diagrams are tied to the exponential distribution or Markovian property. Variations such as Erlang service times can be developed using the state-diagram approach because the Erlang continues with the exponential assumption for the individual phases. The point of view taken for a general service process is to observe the system only at service completion times. This allows us to model, using the Markovian properties of the arrival process, the steady-state system size probabilities at departure points. It turn out that for this $M/G/1$ system, the steady-state probabilities at departure points are the same as the steady-state probabilities at an arbitrary point in time [4, p. 221]. The derivation of these probabilities is beyond the scope of this text and involves developing the generating function transform for the departure point probabilities. The development of the mean values for the number of jobs in the system was initially obtained independently in 1932 by Pollaczek and Khintchine and is now considered a standard property for general service time queueing systems.

Property 3.1. *The Pollaczek and Khintchine, or "P-K", formula for WIP in an $M/G/1$ queueing system is given by*

$$WIP_s = E[N] = \frac{\lambda}{\mu} + \frac{\left(\frac{\lambda}{\mu}\right)^2 + \lambda^2\sigma_s^2}{2\left(1 - \frac{\lambda}{\mu}\right)}$$

where N is the number of jobs in the system, λ is the mean arrival rate, and the service distribution has mean and variance given by $1/\mu$ and σ_s^2, respectively.

The notation used in the above property is common throughout this text. The subscript s used with WIP is to emphasize that the mean work-in-process is over the entire system; the subscript s used with the variance is to emphasize that the parameter refers to the service time distribution and is frequently used to differentiate the service distribution parameters from the inter-arrival parameters.

One implication of Little's Law is that for workstations that have one-at-a-time processing, the relationship between the average number in the system and the average number in the queue is given by $WIP_s - WIP_q = \lambda_e/\mu$. Since $\lambda_e = \lambda$ for $M/G/1$ systems, the expected number of jobs waiting for the processing, $E[N_q]$, is

$$WIP_q = E[N_q] = \frac{\left(\frac{\lambda}{\mu}\right)^2 + \lambda^2\sigma_s^2}{2\left(1-\frac{\lambda}{\mu}\right)} \,.$$

Using Little's Law one more time, the following important property is obtained, and this property will be used to develop approximations for more complicated systems.

Property 3.2. *The P-K formula for the queue cycle time in an $M/G/1$ system is given by*

$$CT_q = E[T_q] = \frac{WIP_q}{\lambda} = \frac{\left(\frac{\lambda}{\mu}\right)^2 + \lambda^2\sigma_s^2}{2\lambda\left(1-\frac{\lambda}{\mu}\right)}$$

where T_q is a random variable denoting the time a job spends in the queue, λ is the mean arrival rate, and the service distribution has mean and variance given by $1/\mu$ and σ_s^2, respectively.

The goal is now to rearrange this formula into a form that will be utilized a great deal in the development of more realistic factory models. First recall from (1.11) that the squared coefficient of variation is defined by

$$C^2[T] = \frac{V[T]}{E[T]^2}$$

so that in terms of service time distribution parameters, we can write

$$C_s^2 = \mu^2\sigma_s^2 \,.$$

Recall from (3.11) and (3.12) that the results for the $M/M/1$ model are

$$\begin{aligned} WIP_s(M/M/1) &= \frac{u}{1-u}\,, \text{ and} \\ CT_s(M/M/1) &= \frac{1}{\mu-\lambda} \end{aligned}$$

where u is the server utilization factor and is equal to λ/μ. Here we have introduced a notational convention of writing the model assumptions (i.e., $M/M/1$) explicitly in the formula. This convention will be used whenever the context does not make the model clear. It should not be difficult to show (hint: use (2.1)) the following:

$$\begin{aligned} WIP_q(M/M/1) &= \frac{u^2}{1-u}\,, \text{ and} \\ CT_q(M/M/1) &= \frac{u}{1-u}E[T_s] \end{aligned} \tag{3.16}$$

where T_s is a random variable denoting the time a job spends in the server.

The P-K formula for cycle time in the queue (Property 3.2) can be rewritten as

$$\begin{aligned} CT_q &= \frac{\left(\frac{\lambda}{\mu}\right)^2 + \lambda^2 \sigma_s^2}{2\lambda\left(1-\frac{\lambda}{\mu}\right)} \\ &= \frac{\left(\frac{\lambda}{\mu}\right)^2 + \lambda^2 \frac{C_s^2}{\mu^2}}{2\lambda\left(1-\frac{\lambda}{\mu}\right)} \\ &= \left(\frac{1+C_s^2}{2}\right)\left(\frac{u}{1-u}\right) E[T_s]\,. \end{aligned}$$

Thus, we have an extremely important (exact) relationship between the $M/G/1$ and the $M/M/1$ models; namely,

$$CT_q(M/G/1) = \left(\frac{1+C_s^2}{2}\right) CT_q(M/M/1)\,. \tag{3.17}$$

3.7.2 Approximations for $G/G/1$ Systems

The P-K mean queue cycle time result (3.17) is based on the assumption of exponential inter-arrival times. Since the coefficient of variation for the exponential distribution is one, the P-K result could just as accurately have been written as

$$CT_q(M/G/1) = \left(\frac{C_a^2+C_s^2}{2}\right) CT_q(M/M/1)\,,$$

where C_a^2 refers to the squared coefficient of variation for the inter-arrival times. This form suggests that the relationship might be a reasonable approximation for the general $G/G/1$ system. In fact, Kingman [7] looked at various approximations in heavy-traffic conditions (i.e., for utilization factors close to 1) and obtained a similar result. Therefore, our first approximation is named after Kingman.

Property 3.3. *The Kingman diffusion approximation for the $G/G/1$ queueing system is*

$$CT_q(G/G/1) \approx \left(\frac{C_a^2+C_s^2}{2}\right) CT_q(M/M/1)\,,$$

where C_a^2 and C_s^2 are the squared coefficients of variation for the inter-arrival distribution and the service time distribution, respectively.

There have been extensive studies using the Kingman diffusion approximation and it has been shown to be an upper bound on the actual mean queue cycle time.

An improved approximation was developed by Kraemer and Langenbach-Belz [8] and studies by Whitt [10] have shown that it is good when the inter-arrival time variability is less than the exponential distribution. Whitt's conclusion is to extend the approximation by adding another multiplicative term resulting in the following:

$$CT_q(G/G/1) \approx g(u, C_a^2, C_s^2) \times \left(\frac{C_a^2 + C_s^2}{2}\right) CT_q(M/M/1)\,, \tag{3.18}$$

where g is a function of server utilization and the two squared coefficients of variation defined as

$$g(u, C_a^2, C_s^2) = \begin{cases} \exp\{-\frac{2(1-u)}{3u}\frac{(1-C_a^2)^2}{C_a^2+C_s^2}\} & \text{for } C_a^2 < 1\,, \\ 1 & \text{for } C_a^2 \geq 1\,. \end{cases}$$

For the remainder of this textbook, the simple form of Kingman's diffusion approximation (Property 3.3) is used with the understanding that improvements are possible using Whitt's extension (3.18). Since the time in the system equals the time in the queue plus the processing time, we also have a good approximation for the system mean cycle time as

$$CT_s(G/G/1) \approx \left(\frac{C_a^2 + C_s^2}{2}\right)\left(\frac{u}{1-u}\right) E[T_s] + E[T_s]\,. \tag{3.19}$$

Example 3.6. Consider again Example 3.3 illustrating an $M/M/1$ system. For this model, $\lambda = 4$/hr and $\mu = 5$/hr yielding a utilization factor $u = 0.8$. Since this was an exponential system, we had $C_a^2 = C_s^2 = 1$ and $E[T_s] = 0.2$ hr. Thus, the $G/G/1$ approximation is

$$CT_q(G/G/1) = \left(\frac{C_a^2 + C_s^2}{2}\right)\left(\frac{u}{1-u}\right) E[T_s] = \left(\frac{1+1}{2}\right)\left(\frac{0.8}{0.2}\right) 0.2 = 0.8 \text{ hr}\,.$$

Whenever the Kingman approximation (Property 3.3) is applied to an $M/M/1$ or $M/G/1$ system, it is exact and not an approximation. We observe that the above result of 0.8 hr for the waiting time agrees exactly with CT_q as calculated in Example 3.3. (It is always nice to have consistency in mathematics!) □

Example 3.7. Consider a $G/G/1$ system with inter-arrival times distributed according to a gamma distribution with mean 15 minutes and standard deviation 30 minutes, and with service times distributed according to an Erlang-4 distribution with mean 12 minutes. Since the distribution of service times is Erlang, the initial temptation may be to use the methodology of Sect. 3.6.1; however, because the arrival times are not exponential, we are left with the $G/G/1$ results. The given data yields the following parameters: $\lambda = 4$/hr, $\mu = 5$/hr, $C_a^2 = 4$, and $C_s^2 = 0.25$. Thus, this example has the same mean characteristics of Example 3.6 yielding a utilization of $u = 0.8$, but the arrival process has more variability and the processing times are less variable. Using the Kingman diffusion approximation (Property 3.3), we have

$$CT_q(G/G/1) \approx \left(\frac{C_a^2 + C_s^2}{2}\right)\left(\frac{u}{1-u}\right)E[T_s] = \left(\frac{4+0.25}{2}\right)\left(\frac{0.8}{0.2}\right)0.2 = 1.7 \text{ hr} .$$

This cycle time is over twice a large as the exponentially distributed system result; thus, the variability associated with non-exponential distributions can have a significant impact on the expected cycle time.

The queue waiting times for single-server queueing systems can be easily simulated with a spreadsheet model (see the Appendix); thus to check the accuracy of the approximation, we simulated the $G/G/1$ system using Excel as discussed in the appendix. (Also refer to the appendix for the importance of reporting confidence intervals along with simulation results.) The simulation yielded a mean waiting time of 1.89 hours with a half-width of ±2 minutes for the 95% confidence interval. It is interesting that when a Weibull distribution with the same mean and variance was used instead of the Gamma distribution, the simulated mean waiting time was 1.71 hours with a half width of ±1.5 minutes for the 95% confidence interval. □

3.7.3 Approximations for $G/G/c$ Systems

There are many generalizations of the $G/G/1$ approximations to account for multiple server systems in the literature. Allen and Cunneen [1] have one of the first commonly used approximation based on the Kingman diffusion approximation. Their approximation was later adjusted by Hall [3] to be a simple extension of Property 3.3 and is given as

$$CT_q(G/G/c) \approx \left(\frac{C_a^2 + C_s^2}{2}\right) CT_q(M/M/c) . \tag{3.20}$$

This form of the multiple server approximation is particularly appealing and will be used herein since it reduces to the form of the single-server approximation when $c = 1$. In addition, it is not too difficult to obtain WIP and CT for an $M/M/2$ system (see Problem 3.9) and the $M/M/3$ system; thus, we have the following two properties.

Property 3.4. *The Kingman diffusion approximation extended for a two-server system is*

$$CT_q(G/G/2) \approx \left(\frac{C_a^2 + C_s^2}{2}\right)\left(\frac{u}{1-u}\right)\left(\frac{u}{1+u}\right)E[T_s] ,$$

where $u = \lambda E[T_s]/2$ is server utilization. This approximation is exact for the $M/M/2$ system.

Property 3.5. *The Kingman diffusion approximation extended for a three-server system is*

$$CT_q(G/G/3) \approx \left(\frac{C_a^2 + C_s^2}{2}\right)\left(\frac{u}{1-u}\right)\left(\frac{3u^2}{2+4u+3u^2}\right)E[T_s]\ ,$$

where $u = \lambda E[T_s]/3$ is server utilization. This approximation is exact for the $M/M/3$ system.

An approximation proposed in Hopp and Spearman [5] uses the following approximation for a Markovian multiple server system from [9]

$$CT_q(M/M/c) = \left(\frac{u^{\sqrt{2c+2}-2}}{c}\right)CT_q(M/M/1)\ .$$

The resulting approximation of Hopp and Spearman yields a general extension as:

Property 3.6. *The Kingman diffusion approximation extended for a multi-server system is*

$$CT_q(G/G/c) \approx \left(\frac{C_a^2 + C_s^2}{2}\right)\left(\frac{u^{\sqrt{2c+2}-1}}{c(1-u)}\right)E[T_s]\ ,$$

where $u = \lambda E[T_s]/c$ is server utilization.

Finally, we repeat the obvious rule for system cycle time (3.19) extended to a multiple-server system that holds whenever service is one-at-a-time:

$$CT_s(G/G/c) = CT_q(G/G/c) + E[T_s]\ . \tag{3.21}$$

Example 3.8. Consider again the system of Example 3.7 except for a two-server system and with a mean service time of 24 minutes. Thus, server utilization stays the same (namely, $u = 0.8$) and the squared coefficients of variation are still given as $C_a^2 = 4$ and $C_s^2 = 0.25$. Then the expected system cycle time using the approximation of Property 3.6 is

$$\begin{aligned} CT_q(G/G/2) &\approx \left(\frac{4+0.25}{2}\right)\left(\frac{(0.8)^{\sqrt{6}-1}}{2(1-0.8)}\right)0.4 \\ &= 1.54 \text{ hr}\ . \end{aligned}$$

If we use Property 3.4, the approximation becomes

$$
\begin{aligned}
CT_q(G/G/2) &\approx \left(\frac{4+0.25}{2}\right)\left(\frac{0.8}{1-0.8}\right)\left(\frac{0.8}{1+0.8}\right)0.4 \\
&= 1.51 \text{ hr} .
\end{aligned}
$$

A simulation of this system yielded a mean cycle time in the queue of 1.63 hr with a half-width of ±0.01 hr for the 95% confidence interval. □

A comparison of the analytical result and the simulation result in the above example illustrates that these approximations are adequate but certainly not exact. Throughout the next four chapters, we will utilize these approximations extensively as we build approximations for more general factory models.

- *Suggestion: Do Problems 3.30–3.32.*

Appendix

In this appendix, we discuss using Excel to solve linear systems of equations and the use of confidence intervals within a simulation. We also present a very simple method for simulating a single-server queueing system with a FIFO queueing discipline.

Solutions to Linear Systems of Equations. Linear systems can always be written in matrix form as

$$A\mathbf{x} = \mathbf{b} ,$$

where A is an $m \times n$ matrix of the coefficients, $\mathbf{x}$ is a vector of n unknowns, and $\mathbf{b}$ is an m dimensioned vector of the right-hand-side constants. If the system has the same number of equations as unknowns (namely, $m = n$) and if the matrix A has an inverse, the solution to this system is

$$\mathbf{x} = A^{-1}\mathbf{b} ,$$

where A^{-1} denotes the inverse of the matrix. Excel has functions for both the matrix inverse and for matrix multiplication. The key to using an Excel function that has an array for the answer, is to highlight the area of the answer and use `<ctrl-shift-enter>` when executing the function. For example, suppose we wish to solve the following system:

$$
\begin{aligned}
3x_1 + 4x_2 + 5x_3 &= 4 \\
2x_1 + 2x_2 + 5x_3 &= 3 \\
1x_1 + 6x_2 - 2x_3 &= 1 .
\end{aligned}
$$

Using Excel, type the coefficient matrix, A, in the square block of cells `A2:C4` and the right-hand-side vector in a single column block of cells `E2:E4` as shown below.

	A	**B**	**C**	**D**	**E**
1	Coefficient Matrix				RHS
2	3	4	5		4
3	2	2	5		3
4	1	6	-2		1

The solution to the system, namely $A^{-1}\mathbf{b}$ is a 3×1 array; therefore, a column of three cells for storing the answer must be selected (highlighted). Choosing the cells `G2:G4` for the answer, select those three cells by placing the mouse in cell `G2` and dragging the mouse down three cells. While the three cells are highlighted, type the following (the typing will be appear in cell `G2` since that is where the selection started)

```
=MMULT(MINVERSE(A2:C4),E2:E4)
```

but do not hit the `<enter>` key. Note that the `MMULT()` function multiplies two arrays, and the `MINVERSE()` function produces the inverse of an array. In Excel, matrix functions always begin with the letter `M`. When finished typing, hold down the `<ctrl>` and `<shift>` keys and while holding these two key down, hit the `<enter>` key. The answer (0.75, 0.125, 0.25) should appear in the highlighted cells `G2:G4`.

Simulation of Waiting Times in a Single-Server Workstation. Consider a G/G/1 queueing system in which each job is numbered sequentially as it arrives. Let the service time of the n^{th} job be denoted by the random variable S_n, the delay time (time spent in the queue) by the random variable D_n, and the inter-arrival time between the n-1^{st} and n^{th} job by the random variable A_n. The delay time of the n^{th} job must equal the delay time of the previous job, plus the previous job's service time, minus the inter-arrival time; however, if inter-arrival time is larger than the previous job's delay time plus service time, then the queueing delay will be zero. In other words, the following must hold

$$D_n = \max\{0,\, D_{n-1} + S_{n-1} - A_n\}\,. \tag{3.22}$$

If we can generate observations of the random variables A_n and S_n for $n = 1, \cdots, n_{\max}$ we will have simulated the arrival and service times for $n_{\max}$ jobs and thus be able to simulate their delays using (3.22). In the Appendix of Chap. 2, the Excel function `RAND()` was used to generate random numbers which are defined as a sequence of numbers appearing to have a continuous uniform distribution between 0 and 1. General random variates can be obtained by the following property that is used to relate random numbers to any other random variable.

Property 3.7. *Let R be a random variable with a continuous uniform distribution between zero and one, and let F be an arbitrary CDF. If the inverse of the function F exists, denote it by F^{-1}; otherwise, let $F^{-1}(a) = \min\{t|F(t) \geq a\}$. Then the random variable X defined by*

$$X = F^{-1}(R),$$

has a distribution function given by F; that is,

$$P\{X \leq a\} = F(a) \quad \text{for } -\infty < a < \infty.$$

To illustrate the use of this property, consider the Excel function

`GAMMAINV(`*probability, shape_parameter, scale_parameter*`)`

that yields the inverse of the gamma CDF evaluated at the specified probability with the given shape, α, and scale, β, parameters (review p. 19); thus,

```
=GAMMAINV(RAND(),4,3)
```

will generate gamma random variates with mean 12 and standard deviation 6 (because the mean is the shape times scale and the variance is shape times scale squared).

To begin a simulation of Example 3.7, type the following in the first three rows of an Excel spreadsheet.

	A	B	C
1	InterArrive	Service	Delay
2	0	`=GAMMAINV(RAND(),4,3)`	0
3	`=GAMMAINV(RAND(),0.25,60)`	`=GAMMAINV(RAND(),4,3)`	`=MAX(0,C2+B2-A3)`

Notice that the references in the `C3` cell are relative references and that two of the references are to the previous row, but the third reference (`A3`) is to the same row. Also, remember that the Erlang distribution is a gamma distribution whose shape parameter is an integer. Now copy the third row down for several thousands of rows and obtain an average of the values in the `C` column. This average is an estimate for the mean cycle time. However, because of the large variablity in the inter-arrival times, the simulation needs to be repeated several times to obtain a good estimate. Reporting the simulation results together with an estimate of its variability is briefly discussed in the next few paragraphs.

Confidence Intervals. Simulations are statistical experiments; therefore, results should never be reported without giving some idea of the accuracy or variability of the statistical information. Assume there is a data set $\{x_1, \cdots, x_n\}$ containing n data points from independent and identically distributed observations. Our goal is to estimate the underlying true (but unknown) mean of the distribution that produced the data. For any data set, the sample mean is given by

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.23}$$

and the sample variance is given by

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^2 - n\overline{x}^2\right). \tag{3.24}$$

Since an estimate for the true mean is desired, the temptation may be to report the sample mean only; however, a single value will provide an estimate but it gives no information on the variability of the estimate. To include information about variability, a *confidence interval* is often used. For example, a 95% confidence interval for the mean implies that if the same experiment were repeated 100 times, approximately 95 of those confidence intervals would contain the true mean; that is, we expect to be correct approximately 19 out of 20 times.

Under the assumption of normally distributed data and unknown variance, the $1-\alpha$ confidence interval for the mean is given by

$$(\overline{x}_n - t_{n-1,\frac{\alpha}{2}} \frac{s_n}{\sqrt{n}} \; , \; \overline{x}_n + t_{n-1,\frac{\alpha}{2}} \frac{s_n}{\sqrt{n}}) \tag{3.25}$$

where $t_{n-1,\alpha/2}$ is a critical value based on the Student-t distribution. Statistical tests are usually better as the degrees-of-freedom increases. (As a rule of thumb, a statistical test loses a degree-of-freedom whenever a parameter must be estimated by the data set; thus, the t-test has only $n-1$ degrees-of-freedom instead of n because we use the data to estimate the variance.)

If using Excel, the function `=TINV(0.05, 24)` would yield the critical value for a 95% t-statistic for a sample of 25 data points. Notice that Excel automatically splits the error into a right-hand error and a left-hand error; thus, if it were desired to obtain the critical value for a 90% confidence interval of a sample of 100 points, the function `=TINV(0.10, 99)` would be used. (As an historical note: when statistical tables were primarily used to obtain the critical value for the statistics, the rule of thumb was to use the z-statistic for large sample sizes; however, with Excel, there is no reason to switch to the z-statistic since Excel does not have a problem with large sample sizes.)

When applying confidence intervals to simulations, care must be taken not to violate the independence assumption. Because sequential output from a simulation are usually correlated, it is best to form a random sample by performing several replicates of the same simulation, where each replicate starts with a different random number seed. The random sample for the confidence interval then comes from the summary statistics of each replicate.

Problems

3.1. Consider a facility open 24 hours per day with a single machine that is used to service only one type of job. The company policy is to limit the number of jobs within the facility at any one time to 4. The mean arrival rate of jobs is 120 jobs per day, and the mean processing time for a job is 15 minutes. Both the processing and inter-arrival times are assumed to be exponentially distributed. Answer the following questions regarding the long-run behavior of the facility.
(a) What is the average number of jobs that arrive to the facility (but not necessarily get in) per hour?

(b) What is the probability that there are no jobs at the facility?
(c) What is the average number of jobs within the facility?
(d) What is the average number of jobs lost per day due to the limited capacity of the facility?
(e) What is the average throughput rate per hour?
(f) What is the average amount of time, in minutes, that a job spends within the facility?

3.2. Consider a single server system with a limit of 3 jobs (an $M/M/1/3$ system). Let λ be the mean arrival rate and μ be the mean service rate.
(a) Use the singleton subset partition method to derive a system of balance equations (note the last equation is the probability norming equation):

$$\begin{aligned}
\lambda p_0 - \mu p_1 &= 0\\
\lambda p_0 + \mu p_2 - (\lambda+\mu) p_1 &= 0\\
\lambda p_1 + \mu p_3 - (\lambda+\mu) p_2 &= 0\\
\lambda p_2 - \mu p_3 &= 0\\
p_0 + p_1 + p_2 + p_3 &= 1.
\end{aligned}$$

(b) Use the subset partition between successive nodes to derive a system of balance equations.
(c) Solve for each p_i in terms of p_0 for each set of balance equations (a and b) to establish that they yield the same solution.

3.3. Consider a two-server system with exponentially distributed inter-arrival and service times. Let λ be the mean arrival rate and μ be the mean service rate of each server. The system has a limit of 3 jobs at any time. The servers work on jobs independently (only one server is working when there is only one job in the system).
(a) Develop the labeled directed arc network for this system.
(b) Write a system of equations, balance and norming equations, for this system.
(c) Solve this system for the general form of the steady-state probabilities.
(d) Write the equation for server utilization in terms of the steady-state probabilities.
(e) What is the mean number of jobs lost per unit time due to the limited system capacity?
(f) What is the system throughput rate? Note that throughput means completed jobs.

3.4. Consider a single-server system with two types of jobs. The system has a limited capacity of three total jobs in the system at any time. The job classes have different mean arrival and service rates, but all are assumed to be exponentially distributed. Let λ_1 be the mean arrival rate and μ_1 be the mean service rate of job type 1, and let λ_2 be the mean arrival rate and μ_2 be the mean service rate of job type 2. Job class 1 are high priority items and, as such, they have preemptive priority over jobs of type 2 on the server. Space within the system limit of three jobs is on a first-come first-service basis; thus, once a low-priority job is in the system, it cannot be replaced by a high-priority job. Although all low-priority jobs must wait until all high-priority jobs have been processed, even if they arrive when a low-priority job is being serviced. Develop the labeled directed arc network for this system. Hint: there are ten different states and the number of each job type must be accounted for separately.

3.5. Consider the $M/M/1/3$ system of Problem 3.2 with an effective arrival rate given by the equation

$$\lambda_e = \lambda(1 - p_3).$$

Compute the effective arrival rate as a function of μ for the following situations:

λ	$\lambda = \mu$	$\lambda = 2\mu$	$\lambda = 3\mu$	$\lambda = 4\mu$
λ_e	?	?	?	?

3.6. Consider solving the set of steady-state equations for a system with a limit on the number of jobs allowed (example $M/M/1/3$). Suppose there are $n_{\max}$ steady-state equations derived from the flow-in equals flow-out approach. Show that if only these equations (omitting the norming equation) are used and if they are linearly independent, then the solution for p_n cannot satisfy the conditions for a pmf. This result leads to the conclusion that this set of equations must be dependent and, therefore, the norming equation must be used in place of one of the other equations.

3.7. Jobs arrive at a single machine for processing. Jobs arrive in groups of two (always) with an exponentially distributed time between groups with mean rate λ. The single server works on individual jobs. The service time is exponentially distributed with a mean rate μ. Let p_n be the probability that there are n jobs in the system in steady-state. Note that there is no limit to the number of jobs allowed into this system. Draw the state diagram with labeled arcs and write the steady-state equations for states 0, 1, 2, 3, 4, and 5. What is the relationship between λ and μ that guarantees that a steady-state exists?

3.8. Redo Problem 3.7 under the assumption that the group size is one with probability 1/2 and two with probability 1/2.

3.9. Consider a factory with a two-identical servers where jobs can be run on either of the two servers. All jobs have the mean-arrival rate of λ and the same mean-service rate μ, and both distributions are assumed to be exponential. Assume that there is no limit on the number of jobs allowed in the system. Thus, the system is an $M/M/2/\infty$ queue.
(a) Develop the steady-state diagram connecting the states of the system.
(b) Develop the system of equations that the steady-state probabilities must satisfy.
(c) Develop the general probability relationship for p_n in terms of p_0.
(d) Develop a formula for p_0. Hint: the appropriate service rate when both servers are busy is 2μ.

3.10. For the $M/M/1/\infty$ model, show that the expected output rate of jobs is equal to the mean input rate λ.

3.11. For the $M/M/1/\infty$ model derive, from the p_n's, an expression for the queue work-in-process WIP_q.

3.12. Using Little's Law, obtain the cycle time in the queue, CT_q, from the result of Problem 3.11.

3.13. The cycle time in the system is logically the cycle time in the queue plus the expected service time

$$CT_s = CT_q + E[T_s].$$

For the $M/M/1/\infty$ model derive an expression for CT_q using the CT_s result of Eq. (3.12).

3.14. Consider an $M/M/1/\infty$ system with a mean arrival rate of $\lambda = 5$ jobs per hour. Compute the system performance measures (WIP_s, CT_s, th_s, u) for several different service rates $\mu \in \{5.5, 6, 7, 8, 9, 10\}$. Graph the WIP_s and CT_s as a function of the system utilization factor u.

3.15. Determine the impact of an arrival rate of 5 per day in Example 3.4 ($\lambda = 5, \mu = 3, \gamma = 2$ in Eq. 3.13) as it reflects on the system parameters.
(a) Write the system of equations for the steady-state probabilities.
(b) Obtain the system performance measures: CT_s, CT_q, WIP_s, WIP_q, utilization u, mean service time $E[T_s]$, and throughput λ_e.

3.16. For a system with non-identical service rates (see Sect. 3.5) and a limit of N jobs in the system (Eq. 3.13), obtain an expression for the mean service time per job, $E[T_s]$, as a function of the mean throughput rate λ_e, the steady-state probabilities p_n and the mean-service rates μ and γ.

3.17. Solve Problem 3.16 for the probabilities given the parameters: $n_{\max} = 4$, $\lambda = 3$, $\mu = 3$, and $\gamma = 2$.

3.18. Consider a two-server system with non-identical machines, exponentially distributed inter-arrival and service times, and a limit of four jobs. The mean inter-arrival rate is λ. The mean service rates are $\gamma < \mu$. Jobs cannot be split across machines. When there is not a queue of waiting jobs and the faster machine completes processing first, the job on the slower machine is immediately moved to the faster machine to complete processing.
(a) Develop the steady-state diagram of the number of jobs in the system and the flow rates between states.
(b) Develop the system of equations describing the steady-state probabilities of being in each state.
(c) Solve this system of equations.

3.19. For Problem 3.18, obtain the system parameters: CT_s, CT_q, WIP_s, WIP_q, u, mean service time $E[T_s]$, the expected number of busy servers (EBS), and throughput th_s.

3.20. A workstation has two different machines for performing two distinct processing tasks. The workstation has one operator that performs all work done in the

workstation on all jobs. That is, the operator stays with a job and moves it from machine to machine to accomplish the necessary processing. Jobs arrive to the workstation at a mean rate λ (exponentially distributed inter-arrival times). Each job is first processed by the operator on Machine 1 which takes an exponentially distributed length of time with mean rate μ. Then the job and operator go to Machine 2 for further processing. The processing time on the second machine is also exponentially distributed but with a mean rate γ. The operator works on one job at a time and completes it before starting on a new job. The company limits the jobs in this workstation to 3.
(a) Define an appropriate state space representation for this model.
(b) Using your state space, develop a state diagram to model this situation.
(c) Write the utilization equation for machine one, using the state probabilities.
(d) Write the operator utilization equation, using the state probabilities.
(e) Write the workstation work-in-process equation, using the state probabilities.
(f) Write the throughput equation, using the state probabilities.

3.21. A company has a special purpose processing area that makes parts used throughout the company. A variety of different parts are made on a single machine and transported to various locations within the company for storage until they are needed in that area. The company has a very experienced employee who does the analysis of the parts currently available throughout the company and then decides what part type is to be made next at this machine. The part-needs analysis and release for processing is performed by this employee in two steps. The needs-analysis step takes 1/2 hour on average, but with the variety of parts to be analyzed, this time is exponentially distributed. Historical data indicates that 7 of every 9 parts analyses results in a standard part-type release and, since the part processing information is already on file, the part order is then released to the machine immediately.

Two of every nine analyses, however, results in the need for a special-purpose part for which the processing data are not available. Thus, this employee then develops a complete processing plan for the part. This processing plan development time averages an additional 2.5 hours. Due to the variety of the special purpose parts, it has been observed that this extra preparation time also is exponentially distributed. The order development employee is additionally charged with keeping the flow of jobs within the machine area reasonably smooth and timely. Towards this objective, the employee has developed the following release strategy. If there are 3 part orders already in the machining area, the employee holds the current completed order at her desk until a part has been completed and shipped. Then the "ready" order is given to the machine area personnel. If there is a completed (but blocked) order on the analyses employee's desk, no new order analysis is started until the blocked order has been cleared and been released to the machining area.

The machining area has only one machine and the average time for processing an order is 70 minutes. Due to the variety of part types, this processing time is exponentially distributed.

Develop a model of the special parts processing workstation (order analyses through processing). This encompasses the analyses employee and the machine (there is an operator for the machine and it is not necessary to keep track of this operator). First draw a diagram of every possible configuration that this workstation can encounter. From this set of configurations, develop a state-space representation for these configurations. Then draw a rate-connected state diagram relating all of these configurations. Develop the steady-state equations for the rate-state diagram. Solve these equations for the steady-state probabilities. And finally, develop the workstation performance measures for this problem (machine utilization, order-development employee utilization, and throughput).

3.22. Consider an $E_2/M/1/3$ model with the arrival rate of 3 jobs per hour and a service rate of 4 jobs per hour. Compute the steady state probabilities and the system performance measures of utilization, CT_s, WIP_s, and throughput. Note that this system has a capacity of 3 jobs.

3.23. Consider an $E_2/M/1/4$ model with the arrival rate of 3 jobs per hour and a service rate of 4 jobs per hour. Compute the steady state probabilities and the system performance measures of utilization, CT_s, WIP_s, and throughput. Note that this system has a capacity of 4 jobs.

3.24. Solve Problem 3.21 using a spreadsheet such as Excel.

3.25. Find the parameters of a GE_2 approximation for a random variable X with specified mean and squared coefficient of variation:

Case	$E[X]$	$C^2[X]$	λ_1	α	λ_2
i	1	5/4			
ii	4/3	3/2			
iii	5	2			
iv	5/8	5/2			

3.26. Develop a model of an $M/GE_2/1/3$ system and compute the system performance measures given the mean arrival rate is 0.2/hr and the service distribution has parameters $E[S] = 5$ hr and $C^2[S] = 2$.

3.27. Develop a model of an $M/GE_2/1/3$ system and compute the system performance measures given the mean arrival rate is 3/hr and the service distribution has parameters $\mu = 3$/hr, $\alpha = 0.5$, and $\gamma = 4$/hr.

3.28. Solve Problems 3.25 and 3.26 using a spreadsheet such as Excel.

3.29. Develop the node-arc diagram for an $M/GE_2/2/3$ system (identical machines).

3.30. Using the approximation of Eq. 3.19, compute the cycle time in an $M/G/1$ system for three systems with the same arrival rates of $\lambda = 4$ and service times $E[T_s] = 0.2$, but different squared coefficients of variation $(C^2[T_s] = 1/2, 1, 2)$.

3.31. Using the data from Problem 3.30, except for λ, develop a graph of the system WIP_s over the utilization from 0.1 to 0.95 in steps of 0.05. Insert three curves into the graph, based on the squared coefficients of variation $(C^2[T_s] = 0.5, 1, 2)$.

3.32. Using the approximation of Property 3.6 and Eq. (3.21), compute the cycle time in the system for three systems with the same mean arrival rates of $\lambda = 4$ and mean service times of $E[T_s] = 0.4$, but different squared coefficients of variation ($C^2[T_s] = 1/2, 1, 2$). Note here that one machine is not adequate since $u > 1$, so assume that there are two-identical machines available, i.e., use an $M/G/2$ system.

3.33. Consider a single-server system with two types of jobs. The system has a limited capacity of three total jobs in the system at any time. The job classes have different mean arrival and service rates, but all are assumed to be exponentially distributed. Let λ_1 be the mean arrival rate and μ_1 be the mean service rate of Job Type 1, and let λ_2 be the mean arrival rate and μ_2 be the mean service rate of Job Type 2. Jobs are served on a first-come first-serve basis (denoted as FCFS or FIFO).
(a) Develop the labeled directed arc network for this system. Hint: there are fifteen different states and the sequence of job types in the queue must be maintained.
(b) Write the equations linking the steady-state probabilities.
(c) Write a formula for computing (in terms of the p_i's) the total WIP_s, WIP_s by product type, throughput, throughput by product type, the system CT_s, CT_s by product type.

3.34. Consider a single-server system with two types of jobs. The system has a limited capacity of three total jobs in the system at any time. The job classes have different mean arrival and service rates, but all are assumed to be exponentially distributed. Let λ_1 be the mean arrival rate and μ_1 be the mean service rate of Job Type 1, and let λ_2 be the mean arrival rate and μ_2 be the mean service rate of Job Type 2. Jobs are served on a non-preemptive priority basis with job type 1 given preference; that is, once a job starts it can not be displaced from the machine.
(a) Develop the labeled directed arc network for this system. Hint: there are thirteen different states and the sequence of job types in the queue will always be Type 1's in front of Type 2's.
(b) Write the equations linking the steady-state probabilities.
(c) Write a formula for computing (in terms of the p_i's) the total WIP_s, WIP_s by product type, throughput, throughput by product type, the system CT_s, and CT_s by product type.

3.35. Team Computer Project. Consider a situation (factory) where there is a limit of 4 jobs allowed at any time; arrivals to a full system are lost. Assume that all interarrival and processing times are exponentially distributed with mean rates specified. Job processing has two steps (Step 1 uses Machine 1 and Step 2 uses Machine 2). That is, there are two independent processing steps that must be done in the sequence: Machine 1 then Machine 2. The system is automated with-respect-to job movement between the queue and machines and between machines and then from the last machine to shipping (not part of this problem). There currently is no space for a job to wait for processing at Machine 2 after it has completed processing at Machine 1. Therefore, the completed job is left on Machine 1 until Machine 2 becomes available.

Management would like to improve the factory throughput and they are want to know what throughput improvement could be gained if they would invest in a

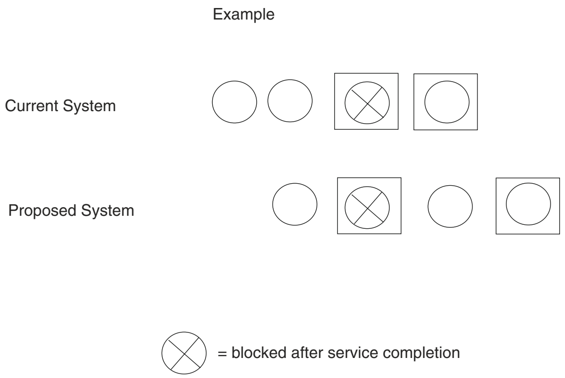


Fig. 3.6 Two configurations for Problem 3.35

conveyor between the machines. Develop a model and obtain the throughput for this system under the following two parameter sets: $\lambda = 6$, $\mu_1 = 8$, $\mu_2 = 7$ and $\lambda = 9$, $\mu_1 = 6$, $\mu_2 = 6$. Contrast the system throughput with and without a single buffer (job holding station) between the two machines for both configurations (see Fig. 3.6).

Develop a computer code to solve these two problems and evaluate the system throughput. Make it general in that the rate parameters are input or specified values within the spreadsheet that can be changed (such as merely changing parameter values between the data sets).

3.36. Model an $E_2/M/1/3$ system with a dependent arrival process in that once the system is full, the arrival process is shutoff until space is available in the system.

References

1. Allen, A.O. (1978). *Probability, Statistics, and Queueing Theory: With Computer Science Applications*, Academic Press, New York.
2. Altiok, T. (1996). *Performance Analysis of Manufacturing Systems*, Springer-Verlag, New York.
3. Hall, R.W. (1991). *Queueing Methods: For Services and Manufacturing*, Prentice-Hall, Englewood Cliffs, N. J.
4. Gross, D., and Harris, C.M. (1998). *Fundamentals of Queueing Theory*, Third Edition, John Wiley & Sons, New York.
5. Hopp, W.J. and Spearman M.L. (1996). *Factory Physics: Foundations of Manufacturing Management*, Irwin, Chicago.
6. Kendall, D.G. (1953). Stochastic Processes Occurring in the Theory of Queues and Their Analysis by the Method of Imbedded Markov Chains. *Annals of Mathematical Statistics*, **24**:338–354.
7. Kingman, J.F.C. (1962). On queues in heavy traffic. *J. Royal Statist. Soc. Ser. B*, **32**:102–110.
8. Kraemer, W. and Langenbach-Belz, M. (1976). Approximate Formulae for the Delay in the Queueing System $GI/G/1$. *Congressbook, Eighth Int. Teletraffic Cong.*, Melbourne.

9. Sakasegawa, H. (1977). An Approximation Formula $L_q = \alpha\beta^{\rho}(1-\rho)$. *Annuals of the Institute of Statistical Mathematics*, **29**:67–75.
10. Whitt, W. (1983). The Queueing Network Analyzer. *The Bell System Technical Journal*, **62**:2779–2814.

Chapter 4
Processing Time Variability

In the previous chapter, an approximation for the cycle time in a system queue was developed (or waiting time in the queue for a machine). The relationship consists of four parameters. These are the squared coefficient of variation of the inter-arrival time process (C_a^2), the squared coefficient of variation of the service time process (C_s^2), the machine utilization (u), and the mean service time ($E[T_s]$). This relationship is

$$CT_q(G/G/1) = \frac{(C_a^2 + C_s^2)}{2}\left(\frac{u}{1-u}\right)E[T_s]\,. \tag{4.1}$$

From this relationship, it is clear that reducing one of the variability components, C_a^2 or C_s^2, will reduce the cycle time in the queue. What might be overlooked is that reducing variability is equivalent to reducing the machine utilization by some factor with respect to the mean cycle time measure. In more direct terms, reducing process variability is equivalent to finding extra capacity in the system since a reduction of utilization with a constant arrival rate implies an increase in the mean processing rate.

To illustrate the equivalence between reducing variability and utilization, consider a single machine system with the following parameter values:

$$\begin{aligned} C_a^2 &= 1 \\ C_s^2 &= 1 \\ u &= 0.8 \\ E[T_s] &= 2 \text{ hr}\,. \end{aligned}$$

Thus the cycle time in the queue CT_q is thus

$$CT_q = \frac{(1+1)}{2}\left(\frac{0.8}{1-0.8}\right)2 \text{ hr} = 8 \text{ hr}\,.$$

Now if C_s^2 is reduced by 10% to 0.9, the resulting cycle time is 7.6 hours, a reduction of 5%. It would take a reduction in machine utilization from 80% to 79.17%

G.L. Curry, R.M. Feldman, *Manufacturing Systems Modeling and Analysis*, 2nd ed.,
DOI 10.1007/978-3-642-16618-1_4, © Springer-Verlag Berlin Heidelberg 2011

to accomplish this same cycle time decrease if C_s^2 was not changed. Thus, reducing service time variability (or inter-arrival time variability) has the same effect as obtaining additional machine capacity. The equivalent utilization factor u is found by solving the equation

$$\begin{aligned}\frac{(1+1)}{2}\left(\frac{u}{1-u}\right)2 &= 7.6,\\ 4.6u &= 3.6,\\ u &= 0.7917.\end{aligned}$$

Now a 50% reduction in the service time variability for this example data would reduce the cycle time measure to 6 hours. The equivalent machine utilization factor for 6 hours given the original system parameters is 0.75. This is a reduction in utilization, or the mean service time, of 6.25%. Either of these changes would result in a cycle time in the queue of 6 hours which is a 25% reduction from the original 8 hours.

The conclusion that can be drawn from this analysis is that reducing component variability is equivalent to increasing system capacity when measured by cycle time response. So it is very important to concentrate on reducing variability for the inter-arrival and service time processes since these reductions are like finding “free” machine capacity.

There are many factors that contribute to the variability of the length of time that a job spends in processing. The term “in processing” indicates that the job has control of the machine and other jobs cannot be processed until this job is completed. Job residence time includes the actual time that the machine is processing the job (herein called the natural processing time to distinguish it from the total time on the machine), any setup needed to place the job on the machine and prepare the machine for the particular job type, any delay due to the unavailability of an operator once the machine is available for allocation to that specific job, and delays due to machine breakdowns and repairs. Scheduled maintenance is normally accounted for in the available machine time rather than accounting for this lost time as part of a specific job’s residence time. The principle contributors to job residence time variability are:

- Natural processing time variability — the variability evident in the time it takes to actually process a specific job type.
- Random breakdowns and repairs during processing — the variability of the time between breakdowns and the variability of the time to repair a broken machine.
- Operator unavailability can induce random delays in the time a job spends “in control of” a machine. This time delay occurs when a machine and job are available with the operator being needed to setup the machine and start processing, but the operator is busy serving another machine/job combination.
- Job class setup and take-down times — the time caused by a job-type change on a machine. This change-over time generally occurs at the end of processing of one job type and the starting of a different job class.

The variability associated with job class setup times is generally for a group of jobs and, in Chap. 7, this delay and associated variability component is modeled as an aspect of batch-type processing. Operator availability and their impacts on system performance is also a complex interaction between the number of operators servicing a set of machines. For the most part, this level of detail will be omitted when complex factory models are developed. However, in the last section of this chapter, a model of this type of multiple resource interaction is illustrated. Concise approximations for quantifying this factor are not available at this time. Thus, the main objective of this chapter is the analysis of the impacts of “natural” processing time variability and breakdown/repair induced variability on workstation performance measures such as cycle time and work-in-process.

4.1 Natural Processing Time Variability

Consider a job with processing time random variable, T, with known mean and variance parameters $E[T]$ and $V[T]$, respectively. If the processing time is made up of several separate tasks, then there is a good opportunity to reduce the total processing time variability by reducing the variability of the individual tasks. Of course, one can directly attempt to reduce the total processing time variability. This is more consistently accomplished when there are sub-tasks that can be studied separately or possibly assigned to different workers for manual task operations. To illustrate this point, consider that the natural processing time random variable T is made up of three separate (independent) sub-tasks. Hence,

$$\begin{aligned} E[T] &= E[T_1] + E[T_2] + E[T_3] \\ V[T] &= V[T_1] + V[T_2] + V[T_3] \\ C^2[T] &= \frac{V[T]}{E[T]^2} \,. \end{aligned}$$

Additionally consider that these three sub-processes times are independent and identically distributed random variables so that

$$\begin{aligned} E[T] &= 3E[T_1] \\ V[T] &= 3V[T_1] \,. \end{aligned}$$

Hence, the individual processing time random variables T_i, for $i = 1,2,3$, have distributional parameters

$$\begin{aligned} E[T_i] &= \frac{E[T]}{3} \\ V[T_i] &= \frac{V[T]}{3} \,. \end{aligned}$$

Furthermore, the squared coefficient of variation of the individual tasks are

$$C^2[T_i] = \frac{V[T_i]}{E[T_i]^2} = \frac{V[T]/3}{E[T]^2/3^2} = 3C^2[T], \text{ for } i = 1,2,3 .$$

So if the total processing time is made up of three identical sub-tasks, then the squared coefficient of variation of the individual tasks is actually three times that of the total time squared coefficient of variation. Now suppose in the analyses of the individual tasks it is found that their variability, as measured by $C^2[T_i]$, can be reduced to that of the total processing time variability. Then the overall processing time squared coefficient of variation $C^2[T]$ would be reduced by 1/3.

Example 4.1. Consider a natural processing time that is exponentially distributed with a mean time of 3 hours. Thus, the squared coefficient of variation $C^2[T]$ is equal to one. Now further assume that this job consists of three distinct but identically distributed sub-tasks. Then these sub-tasks have processing times random variables T_i that have distributional parameters $E[T_i] = 1$ and $V[T_i] = 3$, for each i, by the above analysis.

After further study of the three sub-tasks, it is found that the variability of each task can be substantially reduced and the resulting times are *i.i.d.* exponentially distributed times each with a mean of one hour. (It is assumed that these variabilities can be reduced while the mean processing times remain unchanged.) Thus, $C^2[T_i] = 1$, for each sub-task i. The impact on the variability of the total processing time random variable T is significant. The parameters are now

$$\begin{aligned} E[T_i] &= 1 \\ C^2[T_i] &= 1 \\ V[T_i] &= 1 . \end{aligned}$$

Thus, the total processing time random variable now has parameters

$$\begin{aligned} E[T] &= \sum_{i=1}^{3} E[T_i] = 3 \\ V[T] &= \sum_{i=1}^{3} V[T_i] = 3 \\ C^2[T] &= \frac{3}{3^2} = 1/3 . \end{aligned}$$

For this example, the total processing time variability was reduced to $\frac{1}{3}$ of its original value. This reduction in processing time variability will in turn reduce the associated workstation cycle time in the queue by $\frac{1}{6}$ (why?). So in essence extra processing capability has been found (that is, this new system is equivalent in cycle time response to a system with a faster processing time). □

- *Suggestion: Do Problems 4.1–4.3.*

4.2 Random Breakdowns and Repairs During Processing

A major source of processing time variability is due to the breakdown of an operating machine and the subsequent delay while the machine is being repaired. Several courses of action might result from the breakdown of a machine. The job undergoing processing at the time of the breakdown might not be recoverable (i.e., lost), the job might require additional processing before resumption of "normal" processing, or the job might not be effected by the breakdown and normal processing can resume immediately after the repair is complete (as if the breakdown never occurred). Only the latter case is considered herein, although for the second case, the additional processing time needed to resume service can be included in the machine repair time so that the second and third situations become equivalent.

The assumption being made is that once a machine has been repaired after a breakdown, the job that was processing at the time of the breakdown is continued as if the breakdown never occurred. Thus, a breakdown merely extends the job processing time (actually job residence time with the ”normal” processing time being unaffected). When breakdowns occur, they obviously impact the job residence time on the machine and the resulting job residence time distribution needs to be developed. This is called the effective processing time to distinguish it from the normal processing time.

Definition 4.1. The *effective processing time*, T_e, refers to the time that a job first has control of the processor until the time at which the job releases the processor so that it is available to begin work on another job.

Only the mean and variance parameters of the effective processing time random variable are needed and not the distribution itself. For this development, a given job has several possibilities. The job can complete processing without a breakdown interruption, the machine could breakdown once during service, the machine could breakdown twice during service, etc. So the effective processing time is also a random variable given by

$$T_e = T + \sum_{i=1}^{N} R_i \, , \tag{4.2}$$

where T is the normal (uninterrupted) processing time random variable, the R_i's are the ($i.i.d.$) repair time random variables, and N is the random number of failures during the service time T. The number of failures N is a function of the time between failures random variables, F_i, for the machine in question and is assumed independent of the actual time that it takes to do the repairs, R_i.

A key parameter needed for expressing the effect of failures and repairs on service times is the availability of the processor.

Definition 4.2. The *availability*, a, of a processor that is subject to failures is the long-run average fraction of time that the processor is available for processing jobs.

Using the notation above, it is not difficult to express availability in terms of the mean time to failure and the mean repair time.

Property 4.1. *Processor availability is determined by*

$$a = \frac{E[F_1]}{E[F_1] + E[R_1]}$$

where $F_1, F_2, \cdots$ and $R_1, R_2, \cdots$ are i.i.d. random variables representing successive failure times and successive repair times, respectively, for the processor.

Hopp and Spearman [2] developed an expression for the mean and variance of the effective service time for processors that are less than 100% reliable under the assumption that failures are exponentially distributed:

$$E[T_e] = \frac{E[T_s]}{a} \text{ , and} \tag{4.3}$$

$$C_e^2 = C^2[T_e] = C_s^2 + \frac{(1 + C^2[R_1])a(1-a)E[R_1]}{E[T_s]} \text{ .} \tag{4.4}$$

They show that when T_e and C_e^2 are used in place of T_s and C_s^2 in (4.1) the formula gives an exact expression for the mean waiting time in the queue for a workstation described by an $M/G/1$ system subject to exponential failures. (Notice that when T_e replaces T_s, the utilization factor must be adjusted as well.) For other $G/G/c$ systems, it serves as an approximation.

Example 4.2. Consider a single workstation with jobs arriving according to a Poisson process (i.e., exponential inter-arrival times) with an average time between arrivals of 75 minutes. Initially we ignore the fact that the machine at the workstation is not 100% reliable and observe that the normal processing time is described by an Erlang type-3 distribution with mean of 58 minutes; thus, $C_a = 1$, $E[T_s] = 58$ min, $C_s = 1/3$, and $u = 58/75 = 0.7733$. These parameters used in (4.1) yield $CT_q = 132$ min.

After presenting these results, we are told that the processing machine is not completely reliable. The time between machine breakdowns is exponentially distributed with a mean time of 3 hours measured according to machine processing time and does not include idle time. The repair time is distributed according to a lognormal distribution with a mean time of 30 min and a standard deviation of 15 min yielding a squared coefficient of variation of 0.25 for the repair time. The availability is thus given by

$$\begin{aligned} a &= \frac{E[F_1]}{E[F_1] + E[R_1]} \\ &= \frac{3}{3 + 1/2} = 0.85714 \text{ .} \end{aligned}$$

The mean of the effective processing time (4.3) is

$$E[T_e] = \frac{E[T]}{a} = 67.67 \text{ min} ,$$

and the squared coefficient of variation for the effective processing time (4.4) is

$$C^2[T_e] = \frac{1}{3} + \frac{(1+0.25)(0.85714)(1-0.85714)(30)}{58} = 0.4125 .$$

The effective mean service time yields an effective utilization of $u = 67.67/75 = 0.9023$ so that the application of (4.1) results in a revised value for the expected waiting time in the queue

$$CT_q = \frac{(1+0.4125)}{2}\left(\frac{0.9023}{1-0.9023}\right) 67.67 \text{ min} = 441 \text{ min} .$$

Notice that the inclusion of machine failure in the model results in over a three-fold increase in the mean waiting time; thus, to ignore failures can create significant errors in performance measures. This increase is due to two factors: (1) machine failures cause an increase the effective utilization factor and (2) machine failures cause an increase in the service variability. As the utilization factor approaches one, small changes in the factor will have major changes in waiting times, and in this case, the majority of the increase in waiting times is due to the utilization factor increase; only about 5%–6% of the increase is due to the increase in service variability. □

- *Suggestion: Do Problems 4.4–4.9 and 4.13–4.14.*

4.3 Operator-Machine Interactions

Operators are frequently required to setup a machine for each job. Machine preparation time usually takes significantly more time than the job unloading operation. If the machining operation requires a dedicated operator, then most likely the model of that situation would require only one resource (either the machine or the operator). When an operator is used only part time during processing and the operator is then free to perform other tasks, operators and machines can no longer be modeled as one. In this situation, an operator is frequently assigned control of more than one machine and, thus, is responsible for setting up jobs on several machines. If an operator is assigned to cover too many machines then system performance can be significantly degraded because of delays resulting from waiting for the operator to become available to perform the necessary job setups. Even when the operator is assigned to cover only two machines, some delays will be encountered due to the

[1] Section 4.3 can be omitted without affecting the continuity of the remainder of the text.

timing of job completions. If a system has reasonable capacity, then the operator machine interaction problem does not significantly impact system performance. Thus, this level of detail is frequently omitted in system models. This interaction can, however, degrade system performance significantly if overlooked. The operator-machine interaction problem also offers an opportunity to illustrate how multiple resource interactions can be quantified and evaluated.

In the modeling assumptions, only one job class is treated with two identical machines and one operator. In addition, to simplify the analysis as much as possible, exponentially distributed times are assumed for job inter-arrival times, job setup times, and job processing times. Since we need to keep track of two resources, namely the operator and the machines, a state space that only keeps a record of the number of jobs in the system does not carry enough information to appropriately establish the true system state. Specifically, in addition to the number of jobs in the system, the status of each machine-job combination must be known; that is, the state of the system must include whether the job is "in setup" or "in processing". If two jobs are in the "in setup" status, then only one of them can be actually proceeding with setup because there there is only one operator.

There is often more than one way to define a state space, so that the particular definition chosen is up to the modeler. It is good practice to choose a state space definition that is descriptive so that the individual defining equations for the steady-state probabilities will be easy to read. One descriptive state definition is to use a three-tuple for the states. Each state is represented as (n,i,j), where n denotes the number of jobs in the system and i and j indicate the status of the two machines. There are three possible values for i and j: 0 indicates a machine has no job associated with it, s indicates that a machine has a job "in setup", and p indicates a machine has a job "in process". For example, the state $(1,s,0)$ indicates that there is one job in the system and the operator is setting it up on a machine, state $(5,s,s)$ indicates that there are 5 jobs in the system with one job being set-up on a machine, another job waiting at a machine for the operator, and 3 jobs waiting in the queue for a machine, and state $(7,p,p)$ indicates 7 jobs in the system with both machines busy processing, 5 jobs queued, and the operator idle. Because the machines are identical, it is not necessary to know which machine is processing and which machine is begin setup.

The state space representation for $n \geq 2$ is made up of three individual states: (n,s,s), (n,s,p), and (n,p,p). For $n = 0$, there is no need for all three indices, but for consistency this state is denoted as $(0,0,0)$. For $n = 1$, the possible states are $(1,s,0)$ and $(1,p,0)$. The states of the system, grouped by number of jobs in the system, are

$$\{(0,0,0),(1,s,0),(1,p,0),(2,s,s),(2,s,p),(2,p,p),(3,s,s),(3,s,p),(3,p,p),\cdots\}.$$

The inter-arrival time, setup time, and service time distributions are all assumed to be exponentially distributed. The mean rates for these three processes are denoted by λ, γ, and μ, respectively. Note that if both machines are processing (independently), the mean output rate for the system is 2μ. If both machines are being setup, the mean setup rate is γ, not 2γ, because there is only one operator. The equations relating the steady-state probabilities for this system are:

$$
\begin{aligned}
\lambda p_{(0,0,0)} &= \mu p_{(1,p,0)} \\
(\lambda+\gamma)p_{(1,s,0)} &= \lambda p_{(0,0,0)} + \mu p_{(2,s,p)} \\
(\lambda+\mu)p_{(1,p,0)} &= \gamma p_{(1,s,0)} + 2\mu p_{(2,p,p)} \\
(\lambda+2\mu)p_{(2,p,p)} &= \gamma p_{(2,s,p)} \\
(\lambda+\gamma)p_{(2,s,s)} &= \lambda p_{(1,s,0)} + \mu p_{(3,s,p)} \\
(\lambda+\gamma+\mu)p_{(2,s,p)} &= \lambda p_{(1,p,0)} + \gamma p_{(2,s,s)} + 2\mu p_{(3,p,p)} \\
(\lambda+2\mu)p_{(3,p,p)} &= \lambda p_{(2,p,p)} + \gamma p_{(3,s,p)} \\
(\lambda+\gamma)p_{(3,s,s)} &= \lambda p_{(2,s,s)} + \mu p_{(4,s,p)} \qquad (4.5) \\
(\lambda+\gamma+\mu)p_{(3,s,p)} &= \lambda p_{(2,s,p)} + \gamma p_{(3,s,s)} + 2\mu p_{(4,p,p)} \qquad (4.6) \\
(\lambda+2\mu)p_{(4,p,p)} &= \lambda p_{(3,p,p)} + \gamma p_{(4,s,p)} \qquad (4.7) \\
&\vdots \\
(\lambda+\gamma)p_{(n,s,s)} &= \lambda p_{(n-1,s,s)} + \mu p_{(n+1,s,p)} \\
(\lambda+\gamma+\mu)p_{(n,s,p)} &= \lambda p_{(n-1,s,p)} + \gamma p_{(n,s,s)} + 2\mu p_{(n+1,p,p)} \\
(\lambda+2\mu)p_{(n+1,p,p)} &= \lambda p_{(n,p,p)} + \gamma p_{(n+1,s,p)} \\
&\vdots
\end{aligned}
$$

plus the norming equation, which is the sum of all probabilities equal to one. The set of three numbered equations (4.5–4.7) are repeated with increasing indices. The last three listed equations represent these equations for the index n (where $n \geq 3$). So the system has an infinite number of defining equations with the first seven being special and all others being one of three possible general forms.

The numerical solution scheme employed is rather straightforward, but unfortunately, the solution cannot be represented nicely in closed form. For specified values of the parameter set (λ, γ, μ), the system is solved in the following fashion. The unknown $p_{(0,0,0)}$ is set 1.0, then all other probabilities can be solved recursively for numerical values according to the procedure described in the next paragraph. Since we have an infinite system, it will be truncated to a finite set of probabilities at the point that the probabilities become very small. Thus, we continue to find probabilities until the individual probability terms become very small. At that point, we stop and determine the sum of all probabilities that have been calculated. The final answer then becomes the individual terms divided by this sum.

The process for evaluating the individual probabilities is actually rather straightforward. First given $p_{(0,0,0)}$ the first equation is used to obtain $p_{(1,p,0)}$. Then, using three equations at a time, the probabilities groups solved in turn are: the group $(p_{(1,s,0)}, p_{(2,s,p)}, p_{(2,p,p)})$, then the group $(p_{(2,s,s)}, p_{(3,s,p)}, p_{(3,p,p)})$ and finally $(p_{(3,s,s)}, p_{(4,s,p)}, p_{(4,p,p)})$. Each of these sets is found from the solution of three linear equations. This last set of three equations is repeated solved for $(p_{(n-1,s,s)}, p_{(n,s,p)},$

$p_{(n,p,p)}$) for increasing values of n until the sum of the three values are less that some limit. The solution of each of these three distinct sets of equations needs only values for previously obtained probabilities. Thus, starting with one assumed value, $p_{(0,0,0)}$, as many probabilities as necessary can be obtained; after the relative values for these probabilities have been determined, they are normed to sum to one. One further observation is that only one 3×3 matrix inverse is needed since the same matrix reoccurs as the coefficients of the unknowns for all three forms of the three equations groups.

To be more specific, we first observe that $p_{(1,p,0)} = (\lambda/\mu)p_{(0,0,0)}$. Then the second through fourth equations can be rewritten in matrix form as

$$\begin{bmatrix} -(\lambda+\gamma) & \mu & 0 \\ \gamma & 0 & 2\mu \\ 0 & \gamma & -(\lambda+2\mu) \end{bmatrix} \begin{bmatrix} p_{(1,s,0)} \\ p_{(2,s,p)} \\ p_{(2,p,p)} \end{bmatrix} = \begin{bmatrix} -\lambda p_{(0,0,0)} \\ (\lambda+\mu)p_{(1,p,0)} \\ 0 \end{bmatrix} ,$$

with its solution given by

$$\begin{bmatrix} p_{(1,s,0)} \\ p_{(2,s,p)} \\ p_{(2,p,p)} \end{bmatrix} = \begin{bmatrix} -(\lambda+\gamma) & \mu & 0 \\ \gamma & 0 & 2\mu \\ 0 & \gamma & -(\lambda+2\mu) \end{bmatrix}^{-1} \begin{bmatrix} -\lambda p_{(0,0,0)} \\ (\lambda+\mu)p_{(1,p,0)} \\ 0 \end{bmatrix} . \tag{4.8}$$

Once the values of the probabilities $(p_{(1,s,0)}, p_{(2,s,p)}, p_{(2,p,p)})$ have been obtained, the vector $(p_{(2,s,s)}, p_{(3,s,p)}, p_{(3,p,p)})$ is solved similarly using the fifth through seventh equations in the system. This solution is written as

$$\begin{bmatrix} p_{(2,s,s)} \\ p_{(3,s,p)} \\ p_{(3,p,p)} \end{bmatrix} = \begin{bmatrix} -(\lambda+\gamma) & \mu & 0 \\ \gamma & 0 & 2\mu \\ 0 & \gamma & -(\lambda+2\mu) \end{bmatrix}^{-1} \begin{bmatrix} -\lambda p_{(1,s,0)} \\ -\lambda p_{(1,p,0)} + (\lambda+\mu+\gamma)p_{(2,s,p)} \\ -\lambda p_{(2,p,p)} \end{bmatrix} . \tag{4.9}$$

Equations (4.5–4.7) can now be used to yield the general form of the solution for $n \geq 3$; namely,

$$\begin{bmatrix} p_{(n,s,s)} \\ p_{(n+1,s,p)} \\ p_{(n+1,p,p)} \end{bmatrix} = \begin{bmatrix} -(\lambda+\gamma) & \mu & 0 \\ \gamma & 0 & 2\mu \\ 0 & \gamma & -(\lambda+2\mu) \end{bmatrix}^{-1} \begin{bmatrix} -\lambda p_{(n-1,s,s)} \\ -\lambda p_{(n-1,s,p)} + (\lambda+\mu+\gamma)p_{(n,s,p)} \\ -\lambda p_{(n,p,p)} \end{bmatrix} . \tag{4.10}$$

Notice that the solution to each system always involves the same inverse which greatly simplifies the computational burden of the process.

Not all values for the three parameters will yield a system that can be solved. If the operator sets up too slowly or if the arrival rates are too fast for the processing times, the queues will build up continually and no steady-state is possible. Although developing the steady-state conditions is outside the scope of this text, they are given in [1] and, for completeness, we state them below. Steady-state probabilities will exist if and only if the three parameter values are such that

$$\frac{2(\mu+\gamma)\mu\gamma}{2\mu^2+2\mu\gamma+\gamma^2} < \lambda .$$

Example 4.3. To illustrate the methodology and computations, consider a two-machine system with one server. Let the mean arrival rate of jobs be 1 per hour, the mean time to perform a setup by 15 minutes, and let the mean processing time be 90 minutes. Recall that all the times are exponentially distributed. Thus, $\lambda = 1$, $\gamma = 4$, and $\mu = 2/3$. The matrix that needs to inverted, and its inverse, are

$$\begin{bmatrix} -(\lambda+\gamma) & \mu & 0 \\ \gamma & 0 & 2\mu \\ 0 & \gamma & -(\lambda+2\mu) \end{bmatrix}^{-1} = \begin{bmatrix} -0.1622 & 0.0473 & 0.0270 \\ 0.2838 & 0.3547 & 0.2027 \\ 0.4865 & 0.6081 & -0.0811 \end{bmatrix} .$$

Now setting $p_{(0,0,0)}$ to 1.0 yields $p_{(1,p,0)} = 1.5$. Using (4.8), the first set of three probabilities are

$$\left(p_{(1,s,0)}, p_{(2,s,p)}, p_{(2,p,p)}\right) = (0.2804, 0.6030, 1.0338) .$$

From these values, (4.9) is used to evaluate the next three probabilities

$$\left(p_{(2,s,s)}, p_{(3,s,p)}, p_{(3,p,p)}\right) = (0.1082, 0.3910, 1.1133) .$$

The probabilities $\left(p_{(3,s,s)}, p_{(4,s,p)}, p_{(4,p,p)}\right)$ are obtained based on these previous values using (4.10) to yield

$$\left(p_{(3,s,s)}, p_{(4,s,p)}, p_{(4,p,p)}\right) = (0.0637, 0.3156, 1.0182) .$$

Repeating the use of (4.10), we obtain

$$\begin{aligned} \left(p_{(4,s,s)}, p_{(5,s,p)}, p_{(5,p,p)}\right) &= (0.0489, 0.2713, 0.9014) , \\ \left(p_{(5,s,s)}, p_{(6,s,p)}, p_{(6,p,p)}\right) &= (0.0413, 0.2367, 0.7921) , \\ &\vdots \\ \left(p_{(14,s,s)}, p_{(15,s,p)}, p_{(15,p,p)}\right) &= (0.0110, 0.0635, 0.2129) . \end{aligned}$$

Stopping at this point, these probabilities sum to 15.288. Dividing all of these probabilities by 15.288 yields an approximate solution to this system. It is obvious that since the probability $p_{(15,p,p)}$ is not very close to zero, that this truncated solution will not be very close to the unlimited system solution. In fact using these probability values, the estimate for the mean number of jobs, N_s, in the system is

$$WIP = E[N_s] = 5.606 .$$

As the number of probabilities obtained is increased, the expected system WIP, converges. These iteration results are displayed below where n denotes the number of probabilities obtained. Note that it is not much work to increase the number of probabilities obtained since they are found iteratively three at a time using (4.10) repeatedly:

$$\begin{aligned}
n &= 20, & WIP &= 6.399,\\
n &= 30, & WIP &= 7.263,\\
n &= 40, & WIP &= 7.603,\\
n &= 50, & WIP &= 7.725,\\
n &= 60, & WIP &= 7.658,\\
n &= 70, & WIP &= 7.779,\\
n &= 80, & WIP &= 7.783,\\
n &= 90, & WIP &= 7.785,\\
n &= 100, & WIP &= 7.785.
\end{aligned}$$

The truncated system solution changes very little as more probabilities are added beyond the first 80 probabilities. Thus, a reasonable solution to the unlimited system has been obtained. The expected cycle time in the system from Little's Law is

$$CT = WIP/\lambda = 7.785 \text{ hr} .$$

The expected number of jobs in the operator system is

$$1 \times \left(p_{(1,s,0)} + \sum_{n=2}^{\infty} p_{(n,s,p)} \right) + 2 \times \sum_{n=2}^{\infty} p_{(n,s,s)} = 0.2819 ,$$

with the probability that the operator is idle being

$$p_{(0,0,0)} + p_{(1,p,0)} + \sum_{n=2}^{\infty} p_{(n,p,p)} = 0.75 ,$$

and the machine utilization factor being

$$\frac{1}{2} \times \left(p_{(1,p,0)} + \sum_{n=2}^{\infty} p_{(n,s,p)} \right) + 1 \times \sum_{n=2}^{\infty} p_{(n,p,p)} = 0.8909 .$$

The approach of using a truncated system to approximate the unlimited capacity system leads to the problem of finding the norming constant by iteratively increasing the number allowed in the system until the total non-normed probability sum stabilizes. Using the rate-generator form of the problem for Markov processes, one can develop a closed form representation of this sum and find the non-normed probabilities total sum with the truncation mechanism (see [1]). This approach, however, requires mechanics that will not be developed in this text. □

- *Suggestion: Do Problems 4.10–4.12.*

Problems

4.1. Consider a processing time, T, with measured parameters $E[T] = 6$ and $C^2[T] = 2$, that has four *i.i.d.* sub-tasks, T_i for $i = 1,2,3,4$.
(a) Determine $E[T_i]$ and $C^2[T_i]$ for the sub-tasks.
(b) Assume that the variability of each sub-task can be reduced (identically) so that $C^2[T_i] = 2$. Determine the squared coefficient of variation of the total processing time and the percentage improvement over the "old" processing time variability.

4.2. Consider a processing time, T, with measured parameters $E[T] = 8$ and $C^2[T] = 3$, that has five *i.i.d.* sub-tasks, T_i for $i = 1,2,3,4,5$.
(a) Determine $E[T_i]$ and $C^2[T_i]$ for the sub-tasks.
(b) Assume that the variability of each sub-task can be reduced (identically) so that $C^2[T_i] = 2$. Determine the squared coefficient of variation of the total processing time and the percentage improvement over the "old" processing time variability.

4.3. Consider a processing time that has three independent sub-tasks. These are: the job setup time S, normal processing time P, and job removal time R from the machine. The distributional parameters for these sub-tasks are:

$$\begin{aligned} E[S] &= 10 \text{ min}, \quad C^2[S] = 3 \;, \\ E[P] &= 1 \text{ hr}, \quad C^2[P] = \frac{1}{2} \;, \\ E[R] &= 5 \text{ min}, \quad C^2[R] = 1 \;. \end{aligned}$$

(a) Determine the mean and squared coefficient of variation for the job residence time (total processing time).
(b) After careful study the engineering department has come up with a jig for performing a sizeable proportion of the job setup time off line (while the machine is busy processing another job). The result is that the "on-line" machine setup time is reduced, with resulting parameters $E[S] = 1$ min and $C^2[S] = 1$. In addition, due to an operator suggestion, the job removal time variability was reduced to $C^2[R] = 1/3$. Note that no improvement was made in the actual machine processing time. Determine the mean of the new total processing time (job residence time) and its squared coefficient of variation. What are the percentage improvements over the "old" job residence time parameters?

4.4. Consider a job with processing time distribution parameters $E[T] = 3$ hours and $C^2[T] = 2$. The machine breakdown and repair time characteristic parameters are:

$$\begin{aligned} E[F] &= 7 \text{ hr and } C^2[F] = 1 \;, (C^2[F] \text{ is required to be } 1) \\ E[R] &= 1 \text{ hr and } C^2[R] = 1 \;. \end{aligned}$$

Find the parameters of the effective processing time: $E[T_e]$, $V[T_e]$, and $C^2[T_e]$.

4.5. Consider a job with processing time distribution parameters $E[T] = 3.5$ hours and $C^2[T] = 1.25$. The machine breakdown and repair time characteristic parameters are:

$$E[F] = 9 \text{ hr and } C^2[F] = 1\,, (C^2[F] \text{ is required to be } 1)$$
$$E[R] = 2 \text{ hr and } C^2[R] = 1.5\,.$$

Find the parameters of the effective processing time: $E[T_e]$ and $C^2[T_e]$.

4.6. Compute the percentage increase in the cycle time for a system without machine breakdowns and the same system with breakdowns and repairs. Job arrivals are according to a Poisson process with a mean rate of 4 per hour. The service time distribution parameters are $E[S] = 0.2$ hours and $C_s^2 = 1$. The mean time between breakdowns is 2 hours and the repair time distribution parameters are $E[R] = 1/3$ hour and $C_R^2 = 2$.

4.7. Consider an $M/M/1/3$ system with a server that has exponential time between breakdowns and exponential repair times. Develop the rate-node diagram that connects the states of this system. Given an arrival rate of 5 jobs per hour, a service rate of 4 jobs per hour, a breakdown rate of once per hour, and a mean repair time of 10 minutes, determine the steady-state probabilities for the system states and compute the system performance measures of WIP_s, CT_s, and th_s. In addition, compute the proportion of the time that the machine is idle, down (i.e., under repair), and processing.

4.8. Consider $M/M/1/\infty$ system with a server that has exponential time between breakdowns and exponential repair times. Develop the rate-node diagram that connects the states of this system. Given an arrival rate of 5 jobs per hour, a service rate of 7 jobs per hour, a breakdown rate of once per hour and a mean repair time of 10 minutes, determine the steady-state probabilities for the system states and compute the system performance measures of WIP_s, CT_s, and th_s. Compare these performance results with those obtained by applying the breakdown adjustments of Eqs. (4.3) and (4.4).

4.9. Consider $M/M/1/\infty$ system with a server that has exponential time between breakdowns and Erlang-2 repair times. Develop the rate-node diagram that connects the states of this system. Given an arrival rate of 5 jobs per hour, a service rate of 7 jobs per hour, a breakdown rate of once per hour, and a mean repair time of 10 minutes, determine the steady-state probabilities for the system states and compute the system performance measures of WIP_s, CT_s, and th_s. Compare these performance results with those obtained by applying the breakdown adjustments of Eqs. (4.3) and (4.4).

4.10. Consider a two-machine one-operator system. Let all times be exponentially distributed with mean rates:

$$(\lambda = 1, \gamma = 3, \mu = \frac{2}{3})\,.$$

Set $p_{(0,0,0)} = 0.0842$, and determine the next ten probabilities; that is, $p_{(1,p,0)}$ and then one set of three probabilities for each of the three equation-set forms similar to Eqs. (4.8)–(4.10).

4.11. Develop the system equations for the steady-state probabilities for a single operator servicing three machines. What type of difficulties will have to be overcome to solve this system of equations for $n \to \infty$, where n denotes the number of machines for which the operator is responsible.

4.12. Consider an infinite capacity 3-machine 2-operator service system where an operator is required to setup a job on a machine before processing can begin. Develop the node-arc diagram for 5 or less jobs in the system. That is, develop the diagram explicitly for 0 to 5 jobs in the system with the understanding that the complete diagram would contain an infinite number of nodes. All processes, (arrivals, setups and processing) are assumed to be exponentially distributed with mean rates λ, γ, and μ, respectively.

4.13. Consider an $M/M/2/2$ system with exponential breakdowns (rate β) and repairs (rate γ). The machines are identical and when one machine breaks down with the other machine empty, the job being processed is left on the broken machine while it is being repaired. Develop the state diagram for this system.

4.14. Consider an $M/M/2/2$ system with exponential breakdowns (rate β) and repairs (rate γ). The machines are identical and when one machine breaks down with the other machine empty, the job being processed is moved from the broken machine to the operating machine instantaneously. Develop the state diagram for this system.

References

1. Feldman, R.M., Deuermeyer, B.L., and Valdez-Flores, C. (1993). Utilization of the Method of Linear Matrix Equations to Solve a Quasi-Birth-Death Problem, *J. Applied Probability*, **30**:639–649.
2. Hopp, W.J., and Spearman M.L. (2000). *Factory Physics: Foundations of Manufacturing Management*, second edn. IRWIN, Chicago

Chapter 5
Multiple-Stage Single-Product Factory Models

The mechanics for developing both exact and approximate single workstation models were developed in Chap. 3. Linking several workstations together is a necessary step towards more realistic factory models. In this chapter, the single workstation models are linked together to form more realistic factory models. The approach taken is to use general $G/G/1$ and $G/G/c$ system approximations of Properties 3.3 and 3.5 as the building blocks for multiple workstation systems. To properly connect a series of workstations, the departure process of jobs from each workstation must be characterized. Specifically, the mean of inter-departure times and their squared coefficient of variation must be computed for a workstation. These parameters then describe the arrival process for the downstream workstation. For general system configurations, there are two basic mechanisms that must be explored: (1) the merging of several input streams into a workstation, and (2) the separation or partitioning of a workstation output stream into several different streams for different target workstations. This chapter starts with workstations in series and progresses to more complex general network configurations. Single product models are studied in detail in this chapter and in Chap. 6 the methodology is generalized for multiple product systems.

5.1 Approximating the Departure Process from a Workstation

In the study of single workstation models in Chap. 3, the workstation's impact on the output flow of jobs from the workstation was not considered. This information was not needed to study the performance of a single workstation, but when the output from one workstation becomes the input to the next workstation, this information is critical to system analysis. One of the main concerns of this chapter is the impact that the workstation service and queueing processes have on traffic flow characteristics. That is, we will study how the workstation transforms the inter-arrival process characteristics into output-stream characteristics. Consider first the mean flow rate for a system in steady state. In the long run, the same number of units must depart

G.L. Curry, R.M. Feldman, *Manufacturing Systems Modeling and Analysis*, 2nd ed.,
DOI 10.1007/978-3-642-16618-1_5, © Springer-Verlag Berlin Heidelberg 2011

the workstation as enter the workstation. Otherwise, there would be a buildup (or depletion) of jobs in the workstation and the queue would grow infinitely (or units would need to be created out of nothing) as time extends to infinity. It may be that units are destroyed, but we would account for those units as departing "scrapped" units. Or, it may be that an assembly operation occurs so that the number of units appears to change; however, we would consider the assembled unit as two units so that the net flow of material in is always equal to the net flow out. Applying this conservation of flow concept, the mean output rate from a workstation must equal the mean input rate to that workstation.[1] The inter-arrival and inter-departure times random variables are denoted as T_a and T_d, where the subscripts a and d represent arrivals and departures, respectively, for the workstation. Thus, the conservation of flow concept leads to the following property.

Property 5.1. *The mean arrival rate of jobs to a workstation operating under steady-state conditions equals the mean departure rate of jobs; that is*

$$E[T_a] = E[T_d] \,.$$

For exponential systems, namely $M/M/c$ systems with $c \geq 1$, the output process is probabilistically identical to the input process; namely, the inter-departure times are exponentially distributed so that $C_d^2 = C_a^2 = C_s^2 = 1$. For non-exponential systems, obtaining the value of C_d^2 is a little more involved. Assume for the moment that the workstation is extremely busy, then the distribution of the time between departures would essentially be the service time distribution and so C_d^2 would be expected to be very close in value to C_s^2. At the other extreme, when the system is very lightly loaded, the inter-departure times should be an arrival time minus the service time for the last job plus the service time for the arriving job. Thus, the inter-departure time distribution should be similar to the inter-arrival time distribution so that C_d^2 should be very similar to C_a^2. In fact, for an $M/G/1$ system (remember that $C_a^2 = 1$ for $M/G/1$ systems), Buzacott and Shanthikumar [3] show this is exact; namely,

$$C_d^2(M/G/1) = 1 - u^2 + u^2 C_s^2 \,, \tag{5.1}$$

where u is utilization. They also develop for the $G/G/1$ system a lower bound on C_d^2 as

$$C_d^2(G/G/1) \geq (1-u)\left(1 + uC_a^2\right) C_a^2 + u^2 C_s^2 .$$

A general relationship for a $G/G/1$ system for the squared coefficient of variation was developed by Marshall [4] as

$$C_d^2 = C_a^2 + 2u^2 C_s^2 - 2u(1-u)\, CT_q / E[T_s] \,, \tag{5.2}$$

[1] The conservation of flow concept applied to networks is the same as the rate balance concept used to derive the steady-state probabilities as discussed in Sections 3.1 and 3.2.

which has the workstation queue time as an influencing variable. Using the previously developed approximation $CT_q = ((C_a^2 + C_s^2)/2)uE[T_s]/(1-u)$ (Property 3.3) and substituting it into Marshall's formula, the result is the first equation in the following property taken from Whitt [6].

Property 5.2. *The squared coefficient of variation of the inter-departure times for a single server workstation can be approximated by*

$$C_d^2(G/G/1) \approx \left(1-u^2\right) C_a^2 + u^2 C_s^2 \,,$$

and for multiple server workstations by

$$C_d^2(G/G/c) \approx \left(1-u^2\right) C_a^2 + u^2 \frac{C_s^2 + \sqrt{c} - 1}{\sqrt{c}} \,,$$

where $u = E[T_s]/(c\,E[T_a])$.

The single-server approximation is a weighed sum of the two limiting conditions C_a^2 and C_s^2. Note also that it is what one might conjecture as a generalization of (5.1) since for the $M/G/1$ case $C_a^2 = 1$.

The two approximations given in Property 5.2 will suffice for use in our general queueing network approximation system development. There will be situations, such as a batch server (Chap. 7), where a properly detailed model of the process will produce better results than relying directly on these formulas. The reason for improvements in the batching cases is due more to the lack of the independence assumption between processing times for jobs served in batches than it has to do with the inappropriateness of the C_d^2 approximations themselves.

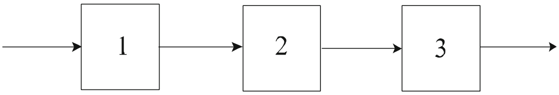
Example 5.1. For a single server workstation, the inter-arrival distribution parameters are $E[T_a] = 20$ min and $C_a^2 = 1/2$. The service time distribution parameters are $E[T_s] = 15$ min and $C_s^2 = 1/3$. Then $\lambda = 3$/hr and $\mu = 4/$hr. Thus, the system utilization factor $u = \lambda/\mu = 3/4$. Using Property 5.2, the approximate value for the squared coefficient of variation of the inter-departure times is given by

$$C_d^2 = \left(1 - \left(\frac{3}{4}\right)^2\right)\frac{1}{2} + \left(\frac{3}{4}\right)^2 \frac{1}{3} = \frac{13}{32} = 0.40625\,.$$

Note that this approximation result does not depend on the distributions of the inter-arrivals or the inter-departures, only there first two moments. □

- *Suggestion: Do Problems 5.1 and 5.2.*

Fig. 5.1 A serial factory structure with three workstations



5.2 Serial Systems Decomposition

The system under consideration in this section is a pure serial system with external inflow into the first workstation only and no branching. The departures from each workstation are the inflows into the next workstation as illustrated in Fig. 5.1. This system is treated as a series of $G/G/c/\infty$ queues with specified service parameters $(E[T_s(i)], C_s^2(i), c_i)$ for each workstation i, numbered from 1 to n. Because of the serial nature of the system, the arrival stream for workstation i is the departure stream from workstation $i-1$; thus, $C_a^2(i) = C_d^2(i-1)$ for $i = 2, \cdots, n$. In addition, the initial workstation inter-arrival time distribution parameters $E[T_a(1)]$ and $C_a^2(1)$ (arriving job characteristics) are assumed known. (In general, the characteristics of arriving jobs from external sources are always assumed to be known.)

If we were limited to exponential processes, the system as a whole could be (theoretically) modeled using the state-diagram approach of Chap. 2; however, the diagram approach becomes intractable even for small networks because of dimensionality problems of the state space. Another relatively easy approach is possible for infinite capacity exponential systems due to the fact that output for any $M/M/c/\infty$ system is a Poisson process (see Burke [2]) with the same parameters as the input process but statistically independent of the input process. Therefore, the approach to modeling the network composed of $M/M/c$ systems is to model each individual node as if it were independent of all other nodes using as input to each node the same arrival process as to the first node.

Example 5.2. Patients arrive to the emergency room according to Poisson process (i.e., with exponential inter-arrival times) with a mean rate of 4 per hour. When they arrive, there is a single clerk who takes their information. This process takes an exponentially distributed length of time with an average of 4 minutes per patient. There is a triage nurse who next sees the patient. The nurse takes an exponentially distributed length of time averaging 10 minutes per patient. Finally one of two doctors sees the patient and each doctor takes an exponentially distributed amount of time with each patient averaging 24 minutes with the doctor. We would like to know the average number of patients within the facility at any one time and the average time that a patient spends in the emergency room.

The emergency room system is composed of an $M/M/1$ system feeding a $\bullet/M/1$ system feeding a $\bullet/M/2$ system. Because of the above mentioned property that $M/M/c$ systems have exponential inter-departure times, the second and third nodes are an $M/M/1$ and $M/M/2$ system, respectively, with an arrival rate of 4 per hour (Property 5.1). Furthermore, since each of the three nodes is an infinite capacity exponential system, the system can be analyzed as three independent single node systems. The first node has a utilization factor of $u_1 = 4/15$ (note that 4 minutes per patient is 15 patients per hour) and thus the average number of patients in the first node is $WIP(1) = 4/11$ (use Eq. 3.11). The second node has a utilization factor of $u_2 = 2/3$ yielding $WIP(2) = 2$ (again use Eq. 3.11). For the third node, we first find the time spent waiting for the doctor. This is given by Property 3.4 and yields $CT_q(3) = 42.67$ min since $u_3 = 0.8$. Adding the doctor's time to the wait time (Eq. 3.21) yields the time spent in third node as $CT(3) = 1.11$ hr. Applying Little's Law (Property 2.1) gives the average number of patients at the node as $WIP(3) = 4.44$. Thus, the total number in the emergency room is $WIP_s = 4/11 + 2 + 4.44 = 6.8$. Applying Little's Law one more time, yields the average value for the total time a patient spends in the emergency room as $CT_s = 1.7$ hr. □

Although the analysis approach used in Example 5.2 is exact only under the assumptions of infinite capacity nodes and exponential distributions for inter-arrivals and processing times, it provides the motivation for approximation schemes when these assumptions do not hold. The analysis approach for general systems is based on the concept that a system's performance can be adequately approximated by separating the system into individual workstations. The performance characteristics of the individual workstations are computed separately and then these results recombined for the total system behavior. This decomposition approach is fundamental to the approximation of general network configurations. The reasons that this decomposition approach is only an approximation are two-fold: first, Property 5.2 is an approximation and second, the successive inter-departure times are not independent except for the $M/M/c/\infty$ case.

The decomposition approach is predicated on being able to establish the individual workstation parameters needed for using Property 3.3 or 3.6. The required data are the parameter set $(E[T_s(i)], C_s^2(i), c_i, E[T_a(i)], C_a^2(i))$ for each workstation i. The first three parameters are specified data for the workstation. The last two parameters in the set are for the job arrival stream into the workstation. These two inter-arrival distribution parameters need to be estimated from the departure flows from the upstream workstations and, of course, the network structure. For serial systems, the outflow from one workstation is the direct inflow into the next, so this particular serial network topology allows for a sequential computation of these unknown parameters. Starting with the known inflow data into the first workstation, all the necessary data are available and the first workstation's performance characteristics (from Properties 3.3 or 3.6) and the departure stream characteristics (from Properties 5.2) can be computed. The second workstation arrival stream characteristics are made equal to the first workstation's departure stream. Thus for the second workstation, the performance information and the departure stream parameters are obtained. This becomes the needed information for the third workstation, and so on. (It is now, hopefully, apparent how the topology of the network impacts the analysis. For a general system structure, the topology is more complex and these data must be computed simultaneously leading to the development of a system of equations as seen in Section 5.4 that must be solved to obtain the inter-arrival distribution parameters.) As always, the arrival stream and service characteristics define the workstation utilization as $u_i = E[T_s(i)]/(c_i E[T_a(i)])$.

The departure stream characteristics for each workstation consists of the mean inter-arrival time and the squared coefficient of variation of these times. For a serial system in steady state, the workstation mean inflow rates must be identical for all workstations. (The assumptions of no losses, no reworks, and one external inflow point are critical to this simplified method for computing these inflow rates.) Thus, $E[T_a(i)] = E[T_a(1)]$ for all workstations $i = 2, \cdots, n$. There remains only the task of computing the $C_d^2(i)$ term for each workstation i and the serial structure of the network allows for these computations to be carried out sequentially. A recursive algorithm can be easily developed for the factory based on the following two properties.

Property 5.3. *The mean cycle time and departure process for an infinite capacity single-server workstation within a factory that has a pure serial system topology are given by*

$$CT(i) \approx \left(\frac{C_d^2(i-1)+C_s^2(i)}{2}\right)\left(\frac{u_i}{1-u_i}\right)E[T_s(i)]+E[T_s(i)] \quad \textit{and}$$
$$C_d^2(i) \approx \left(1-u_i^2\right)C_d^2(i-1)+u_i^2 C_s^2(i)\,,$$

where i is the sequence number of the workstation and $C_d^2(0)$ is the squared coefficient of variation of the arrival stream to the first workstation. (The only arrivals are to the first workstation.)

Property 5.4. *The mean cycle time and departure process for an infinite capacity workstation with c servers within a factory that has a pure serial system topology are given by*

$$CT(i) \approx \left(\frac{C_d^2(i-1)+C_s^2(i)}{2}\right)\left(\frac{u_i^{\sqrt{2c_i+2}-1}}{c_i(1-u_i)}\right)E[T_s(i)]+E[T_s(i)] \quad \textit{and}$$
$$C_d^2(i) \approx 1+\left(1-u_i^2\right)\left(C_d^2(i-1)-1\right)+u_i^2\,\frac{\left(C_s^2(i)-1\right)}{\sqrt{c_i}}\,,$$

where i is the sequence number of the workstation and $C_d^2(0)$ is the squared coefficient of variation of the arrival stream to the first workstation. (The only arrivals are to the first workstation.)

Once the cycle times for the individual workstations have been obtained, the overall system performance measures can be determined. The cycle time in the total system can be computed for serial systems by merely summing the individual workstation times since every job visits each workstation exactly once during its processing. This is not a general computation scheme and is, therefore, forgone in favor of a method that is valid for all network topologies. The more general approach is to use Little's Law to compute the mean number of jobs, $WIP_s(i)$, in each workstation, sum the workstation means together to obtain the total factory mean number of jobs, WIP_s, and then obtain the system mean cycle time through the application of Little's Law again; thus

$$WIP_s = \sum_{i=1}^{n} WIP_s(i) = \sum_{i=1}^{n} \frac{CT(i)}{E[T_a(i)]} \quad \text{and} \tag{5.3}$$

$$CT_s = E[T_a(1)] \times WIP_s \,. \tag{5.4}$$

Equation (5.3) is independent of the job flow sequence and, hence, valid for any network topology. Notice that for the mean throughput rate, the reciprocal of the mean inter-arrival times is used since all arrivals will eventually pass through the workstation. Equation (5.4) is not very general because it assumes that all arrivals to the factory enter through the first workstation. In later sections, this may not be true.

Example 5.3. Consider a three-workstation factory with serial flow as depicted in Fig. 5.1. Each workstation has a single machine with the service time distribution parameters as listed in Table 5.1. The inter-arrival time distribution for jobs to the factory has a mean of 15 minutes or a mean rate of 4 jobs per hour, and a squared coefficient of variation of 0.75. The system mean work-in-process, cycle time, and throughput are desired.

Table 5.1 Service time characteristics for Example 5.3

Workstation i	$E[T_s(i)]$	$C_s^2(i)$
1	12 min	2.0
2	9 min	0.7
3	13.2 min	1.0

Since arrivals to the system occur at the first workstation, $E[T_a(1)] = 15$ min yielding a utilization factor of $u_1 = E[T_s(1)]/E[T_a(1)] = 0.8$. Using the network decomposition principle together with Property 5.3 yields the following for the first workstation:

$$\begin{aligned}
CT(1) &= \left(\frac{C_a^2(1) + C_s^2(1)}{2}\right)\left(\frac{u_1}{1-u_1}\right) E[T_s(1)] + E[T_s(1)] \\
&= \left(\frac{0.75 + 2.0}{2}\right)\frac{0.8}{0.2}(12 \text{ min}) + 12 \text{ min} \\
&= 78 \text{ min} = 1.3 \text{ hr} \\
C_d^2(1) &= \left(1 - u_1^2\right) C_a^2(1) + u_1^2 C_s^2(1) \\
&= \left(1 - 0.8^2\right) 0.75 + 0.8^2 (2.0) = 1.55\,, \quad \text{and}
\end{aligned}$$

$$WIP(1) = CT(1) \times \frac{1}{E[T_a(1)]} = \frac{1.3 \text{ hr}}{0.25 \text{ hr}} = 5.2 \ .$$

The last equation comes from the application of Little's Law, and since no jobs are lost, the throughput rate is $th = 1/E[T_a(1)]$. Notice that care must always be taken to make sure that the time units are consistent when applying Little's Law. Because this is a pure serial network, the arrival rate and throughput rate will be the same for each workstation; thus, the utilization factors for the other two workstations are $u_2 = E[T_s(2)]/E[T_a(1)] = 0.6$ and $u_3 = E[T_s(3)]/E[T_a(1)] = 0.88$. Applying Property 5.3 and Little's Law to the second and third workstations yield

$$\begin{aligned} CT(2) &= \left(\frac{1.55+0.7}{2}\right)\frac{0.6}{0.4}(0.15 \text{ hr}) + 0.15 \text{ hr} = 0.403 \text{ hr} \\ C_d^2(2) &= \left(1-0.6^2\right)1.55 + 0.6^2(0.7) = 1.244 \\ WIP(2) &= CT(2)/E[T_a(1)] = 1.613 \quad \text{and} \end{aligned}$$

$$\begin{aligned} CT(3) &= \left(\frac{1.244+1.0}{2}\right)\frac{0.88}{0.12}(0.22 \text{ hr}) + 0.22 \text{ hr} = 2.030 \text{ hr} \\ C_d^2(3) &= \left(1-0.88^2\right)1.244 + 0.88^2(1.0) = 1.055 \\ WIP_s(3) &= CT(3)/E[T_a(1)] = 8.121 \ . \end{aligned}$$

Finally, the total factory performance characteristics for this serial system are

$$\begin{aligned} WIP_s &= 5.200 + 1.613 + 8.121 = 14.933 \text{ jobs} \\ th_s &= \frac{1}{E[T_a(1)]} = 4/\text{hr} \\ CT_s &= \frac{WIP_s}{th_s} = 3.733 \text{ hr} \ . \end{aligned}$$

As a comparison, a simulation model was developed for this serial factory structure using Excel. (The appendix of this chapter presents the use of Excel for simulating networks for single-server workstations.) The gamma distribution was used for the random inter-arrival times and service times with the appropriate means and squared coefficients of variations. Five replicates of the model were obtained with each replication being a simulation of 32,000 customers through the system. Table 5.2 displays the analytical approximation results with those obtained from the simulation. The analytical approximations are given first followed across the row by the simulation estimates with the half-width of the 95% confidence interval also shown for the simulation. (The estimate for the squared coefficients of variation were obtained by estimating the variance and dividing by the square of the mean estimate; thus, it is a biased statistic. The confidence interval is based on Eq. (3.25) so it is technically not correct for ratios; however, it does give some idea of the variability of the estimator.)

Table 5.2 Comparison of analytical approximation results and simulation results for Example 5.3, including half-widths of the 95% confidence intervals for the simulated estimators

	Approximation CT	Approximation C_d^2	Simulation CT	Simulation C_d^2
Workstation 1	1.300 hr	1.550 hr	1.33 hr ±0.10	1.58 hr ±0.03
Workstation 2	0.403 hr	1.244 hr	0.44 hr ±0.01	1.16 hr ±0.02
Workstation 3	2.030 hr	1.055 hr	1.90 hr ±0.23	1.05 hr ±0.02
System	3.733 hr		3.67 hr ±0.21	

These comparisons are given not to verify that the mathematical models are extremely accurate, but to illustrate that the results are accurate enough for the use of decisions to be made based on these models. The analytical results are static as the distributions vary as long as means and variances remain constant; however, the simulation results vary according to the distributions chosen and between different simulation realizations of the process. □

- *Suggestion: Do Problems 5.3–5.10.*

5.3 Nonserial Network Models

Many production systems have more than one inflow point into the production system. Products that may have been found defective or that have broken may be sent back to the manufacturing facility to be reworked. These units will not necessarily enter the production line at the same point as a new job. If a defect is found during inspection after partially completing production, it may be sent to a rework station and then re-enter the production sequence at the appropriate point. To study factory structures that are more realistic than pure serial systems, two additional structures must be studied in order to compute the squared coefficients of the various streams of jobs within the factory: (1) the merging of streams entering a workstation and (2) the splitting of output streams that come from a single workstation but are routed to more than one workstation. These two processes, merging and splitting, are addressed separately. Then these processes are combined for a general network model.

5.3.1 Merging Inflow Streams

When multiple inflow streams as depicted in Fig. 5.2 arrive at a workstation with differing inter-arrival time distributions, the composite inter-arrival time distribution parameters, mean time or rate and the squared coefficient of variation, need be computed. The process of merging inflow streams is technically called a *superposition* of the individual inter-arrival processes. It is assumed that the individual input streams are independent of one another and that each has independent and

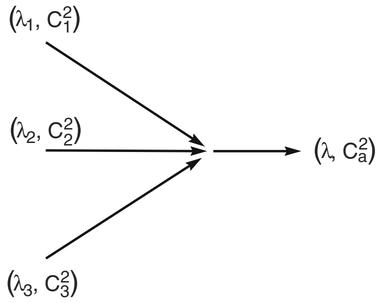


Fig. 5.2 Superposition of merging inflow streams approximated by a two parameter renewal process

identically distributed inter-arrival times (each of these input streams is said to be a *renewal process*).

Definition 5.1. A *renewal process* is the process formed by the sum of nonnegative random variables that are independent and identically distributed. If the random variables forming the sum are exponentially distributed, the renewal process is called a *Poisson process*.

Unfortunately, the superposition of renewal processes is not a renewal process unless each process is a Poisson process. The exact inter-arrival time process of the composite inflow stream is very complicated in general; therefore, we will approximate the resulting stream by (incorrectly) assuming that it is a renewal process as suggested in [1]. The issue is then how to compute the process parameters (namely, the mean and squared coefficient of variation) for the composite stream.

The mean rate of the composite stream is easy to compute since it is the sum of the mean rates of the individual streams; however, the squared coefficient of variation is more difficult to determine. One difficulty is that there is more than one method that can be used for the estimation. The method we shall use is an asymptotic approximation for the squared coefficient of variation and is based on limiting characteristics of the distribution. This method was was proposed by Whitt [5] and we use it in the following property for the composite arrival stream.

Property 5.5. *Consider an arrival stream that is formed by merging n individual arrival processes. The individual streams have mean arrival rates given by $\lambda_i = 1/E[T_i]$ and squared coefficients of variation denoted by C_i^2 for $i = 1, \cdots, n$. The mean arrival rate, λ_a, and the squared coefficient of variation, C_a^2, for a renewal process used to approximate the merged arrival process are given by*

$$\lambda_a = \sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} \frac{1}{E[T_i]}$$

$$C_a^2 = \sum_{i=1}^{n} \frac{\lambda_i}{\lambda_a} C_i^2 \, .$$

Example 5.4. An automated lubricating facility is located in the center of a manufacturing plant. Arrivals of parts needing lubrication come from three sources: manufactured parts needing assembly, defective parts that have been disassembled and will be returned for reassembly, and parts coming from a sister manufacturing facility in another part of the town. The three arrival streams have been analyzed separately. The mean arrival rates for the three streams are given by the vector $(\lambda_1, \lambda_2, \lambda_3) = (13.2/\text{hr}, 3.6/\text{hr}, 6.0/\text{hr})$. The squared coefficients of variation for the three inflow streams are $(C_1^2, C_2^2, C_3^2) = (5.0, 3.0, 2.2)$. The total inflow into the workstation is the sum of the individual inflows so that $\lambda_a = 22.8/\text{hr}$. The relative weights, 13.2/22.8, 3.6/22.8, and 6.0/22.8, are thus used to determine the composite inflow stream's squared coefficient of variation as

$$C_a^2 = \frac{13.2}{22.8} 5.0 + \frac{3.6}{22.8} 3.0 + \frac{6.0}{22.8} 2.2 = 3.947 \, .$$

To compute the mean and standard deviation of the inter-arrival times, remember that mean rates and mean times are reciprocals; therefore,

$$E[T_a] = \frac{1}{22.8} \text{ hr} = 2.63 \text{ min} \, , \quad \text{and}$$
$$V[T_a] = 3.947(2.63^2) = 27.30 \text{ min}^2 \, .$$

□

- *Suggestion: Do Problems 5.11 and 5.12.*

5.3.2 Random Splitting of the Departure Stream

Jobs that exit from a workstation can be transferred to different workstations based on several possibilities. Multiple products can be made by specializing a partially processed product. Thus, the processing sequences can be identical through some step at which point the items are branched to their unique completion workstations or sequence of workstations. Another instance occurs due to quality control testing with good items continuing on their normal route and bad items being reworked or corrected at a different workstation before continuing normal processing. If the branching decision is based on an independent random draw for each job, called a

Bernoulli decomposition or a Markovian routing, then the squared coefficients of variation for the individual resultant streams is exact and relatively easy to compute as long as the initial stream was a renewal process. Specifically, when a renewal process undergoes a Bernoulli decomposition, each individual stream is again a renewal process. Whitt [6] reminds us that this process ultimately is an approximation because in a network of workstations the output process from a workstation "is typically not a renewal process and the splitting is often not according to Markovian routing."

To illustrate the computations necessary for obtaining the mean rate and coefficient of variation for a stream that is split from another stream, assume that p is the probability that output from one workstation is directed as an arrival process to a second workstation. The arrival stream to the second workstation is made up of the sum of one or more inter-departure times from the first workstation. That is, if there are N departures from the first workstation between arrivals to the second workstation, then the second workstation sees an inter-arrival time that is the sum of those N inter-departure times from the first workstation. The number of departures, N, between routings to the target workstation is obviously a random variable, and is distributed according to a geometric distribution. Thus, the probability mass function of N is given by

$$\Pr\{N = n\} = f(n) = p\,(1-p)^{n-1}\,, n = 1, 2, \cdots ,$$

where p is the probability that a given job is routed to the second workstation, independent of previous or future routings. The characteristics for this geometric random variable N (review p. 15) are therefore given by

$$\begin{aligned} E[N] &= \frac{1}{p} \\ V[N] &= \frac{1-p}{p^2}\ . \end{aligned}$$

To compute the time between visits to the second workstation for jobs departing from the first workstation, we define the random variable T as the random sum of N of the independent and identically distributed inter-departure times, T_i; namely,

$$T = T_1 + \cdots + T_N = \sum_{i=1}^{N} T_i\ .$$

Since this is a random sum of i.i.d. random variables, we can use Property 1.9 to obtain the mean and variance of T as

$$\begin{aligned} E[T] &= \frac{E[T_1]}{p} \\ V[T] &= \frac{V[T_1]}{p} + \frac{(1-p)E[T_1]^2}{p^2}\ . \end{aligned}$$

Noting that $C^2[t] = V[T]/(E[T])^2$, it is not too hard to derive the following property for split streams.

Property 5.6. *Consider a departure stream from a specified workstation with a mean inter-departure time and coefficient of variation given by $E[T_d]$ and C_d^2, respectively. When a job departs from the specified workstation, there is a probability, p, that the job will be routed to a target workstation. If there are no other arriving streams to the target workstation, then the mean inter-arrival time and squared coefficient of variation for arrivals to target workstation are given by*

$$\begin{aligned} E[T_a] &= \frac{E[T_d]}{p} \\ C_a^2 &= pC_d^2 + 1 - p \,. \end{aligned}$$

If λ_d is the mean departure rate of jobs from the specified workstation, the mean arrival rate to the target workstation is $\lambda_a = p\,\lambda_d$.

Example 5.5. The fifth workstation within a manufacturing facility performs a quality control check on partially manufactured items. Parts receive an unqualified pass from the inspector with probability 0.8 and they are then sent to Workstation 6 to continue the manufacturing process. Approximately 18% of the time, a part has a partial pass of the quality check and is sent to Workstation 10 for rework. And approximately 2% of the time, a part completely fails the test and is sent to the hazardous waste station for disposal which is designated as Workstation 99. The throughput rate for Workstation 5 is 7 jobs per hour and the coefficient of variation for the inter-departure times is 3. As a notational convention, we let $\lambda_a(i,j)$ denote the mean arrival rate of jobs coming from Workstation i going to Workstation j. Likewise, $C_a^2(i,j)$ denotes the squared coefficient of variation for the stream of jobs from Workstation i feeding into Workstation j. Thus, Property 5.6 yields the following:

$$\begin{aligned} \lambda_a(5,6) &= 0.8 \times 7 = 5.6/\text{hr} \\ C_a^2(5,6) &= 0.8 \times 3 + 0.2 = 2.6 \end{aligned}$$

$$\begin{aligned} \lambda_a(5,10) &= 0.18 \times 7 = 1.26/\text{hr} \\ C_a^2(5,10) &= 0.18 \times 3 + 0.82 = 1.36 \end{aligned}$$

$$\begin{aligned} \lambda_a(5,99) &= 0.02 \times 7 = 0.14/\text{hr} \\ C_a^2(5,99) &= 0.02 \times 3 + 0.98 = 1.04 \,. \end{aligned}$$

Notice that as a check, the arrival rates can be summed and they must equal the departure rate from the original stream before it was split. (As a reminder, such a property does *not* hold for the squared coefficients of variation.) □

- *Suggestion: Do Problem 5.11.*

5.4 The General Network Approximation Model

Our goal is to develop a methodology for approximating the system performance measures for general factory models. In the serial models studied in the previous chapter, the flow structure was straight forward with no losses between workstations and no job feedback, no branching or other nonserial complications. To address a general factory network connection topology, the possibilities of external flows into any one of the workstations must be considered along with job feedback branching for rework purposes, splitting of the output from a workstation to different next workstations, etc. So workstation inflows can come from a variety of sources, external as well as other workstations within the factory, and this complication is handled by our flow merging mechanism. Probabilistic branching of workstation outflow requires departure stream splitting mechanics. Thus, at this point the fundamental mechanisms needed to address these more complicated system structures have been developed. The major complication that arrises is the order that the workstations are sequenced for application of the general decomposition approach. That is, since there is no longer sequential flows, parameter dependencies are also not sequential so that equations relating the parameters will have to be solved simultaneously instead of sequentially.

The concept of the decomposition approach to factory analysis is the establishment of the individual workstation parameters and then the development of each workstation's behavioral characteristics as a stand-alone analysis. These individual analyses are then merged together to estimate the total system behavior. This approach was readily implemented for a pure serial system since the parameters, such as the inflow stream characteristics, could be sequentially computed. Starting with a known inflow into the first workstation and based on its service characteristics (mean, squared coefficient of variation, and number of servers), the outflow or departure stream characteristics were computed. Then due to the serial factory flow structure, these become the characteristics of the inflow stream for the next workstation in series. This sequential process of evaluation is repeated until the last workstation in the series had been evaluated. Then, of course, the results for the individual workstations are combined for the system performance estimation. Determining the mean rates and then squared coefficients of variation for inter-arrival times involve distinct analyses so these are discussed separately in the following two subsections.

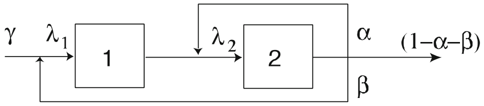


Fig. 5.3 Example of a non-serial factory model

5.4.1 Computing Workstation Mean Arrival Rates

With a non-serial network, determining the arrival stream characteristics is more complicated than for the serial systems. Consider for example, the simple two workstation example of Fig. 5.3. Arrivals from an external source enter the first workstation with a mean rate of γ. However, due to the feedback from Workstation 2 with probability β, the total inflow into Workstation 1 is not explicity given. The same situation arrises for Workstation 2 since the inflow comes from Workstation 1 plus direct feedback from its own departure stream. Since the flow rate into Workstation 1 is not known as yet, the inflow into Workstation 2 cannot be computed directly. This dilemma is a natural consequence of non-serial network flows and its resolution requires that all of the flow rates be computed simultaneously. For this example, note that λ_i for $i = 1,2$, is used to describe the net, or total, arrival rate into each Workstation i. Since steady-state conditions are assumed, λ_i is also the total outflow from Workstation i. These mean rates are defined by the system of linear equations

$$\begin{aligned}\lambda_1 &= \gamma + \beta\lambda_2,\\ \lambda_2 &= \lambda_1 + \alpha\lambda_{2,}\end{aligned}$$

where the parameters α, β, γ are all known data. This linear system rearranged in terms of the unknowns on the left side of the equality is

$$\begin{aligned}\lambda_1 - \beta\lambda_2 &= \gamma,\\ -\lambda_1 + (1-\alpha)\lambda_2 &= 0.\end{aligned}$$

The solution to this system is easily obtained when the parameters α, β, γ are known and can be written in matrix form as

$$\begin{pmatrix}\lambda_1\\ \lambda_2\end{pmatrix} = \begin{pmatrix}1 & -\beta\\ -1 & 1-\alpha\end{pmatrix}^{-1}\begin{pmatrix}\gamma\\ 0\end{pmatrix}.$$

Therefore, a system of linear equations must be established and solved to obtain the mean inflow rates for each workstation. This linear system of equations is, of course, based on the workstation connections for the factory under consideration. To formalize for a general network application, the switching rule needs to be defined.

Definition 5.2. Consider a network consisting of workstations numbered from 1 to n. The *switching rule* for the network is defined by an $n \times n$ matrix $P = (p_{ij})$, where $p_{i,j}$ is the probability that an arbitrary job leaving Workstation i will be routed directly to Workstation j. The matrix P is called the *routing matrix* for the network.

Notice that row i of the routing matrix consists of the probabilities relating to the splitting of the outflow from Workstation i into the various resultant successor Workstations j. The j^{th} column of the matrix represents the probabilities that jobs leaving the various workstations go to Workstation j. (Those familiar with Markov chains will recognize the routing matrix as a sub-Markov matrix since it is made up of nonnegative probabilities and the sum of each row is equal to or less than one.) Also define γ_i as the external inflow rate and λ_i as the total inflow rate into Workstation i. Therefore, the total rate into Workstation i must satisfy the following equation:

$$\lambda_i = \gamma_i + \sum_{k=1}^{n} p_{ki}\lambda_k, \text{ for } i = 1, \cdots, n ,$$

or in standard matrix form,

$$\boldsymbol{\lambda} = P^T\boldsymbol{\lambda} + \boldsymbol{\gamma} ,$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma}$ are n-dimensional column vectors of the λ_i and γ_i terms and P^T denotes the transpose of P. The above equation can be easily solved to yield the following property.

Property 5.7. *Consider a general network of n workstations with switching rule defined by the routing matrix P and assume that the sum of at least one row of P is strictly less than one (i.e., jobs exit the network from at least one workstation). Let $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_n)$ denote a vector consisting of the mean arrival rate of jobs from an external source to the workstations. Both P and $\boldsymbol{\gamma}$ are known. Let $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_n)$ be the (unknown) vector denoting mean arrival rates of all jobs to the workstations. The vector $\boldsymbol{\lambda}$ is given by*

$$\boldsymbol{\lambda} = \left(I - P^T\right)^{-1} \boldsymbol{\gamma} ,$$

where I is an $n \times n$ identity matrix.

Example 5.6. Consider the factory network of workstations depicted in Fig. 5.4 with the noted branching probabilities and an external flow rate into the first workstation of 5 jobs per hour.

The system of equations defining the workstation total arrival rates are

$$\begin{aligned}
\lambda_1 &= 5 + 0.10\lambda_2 + 0.05\lambda_3 \\
\lambda_2 &= 0 + 0.75\lambda_1 \\
\lambda_3 &= 0 + 0.25\lambda_1 + 0.90\lambda_2 .
\end{aligned}$$

This system rearranged is

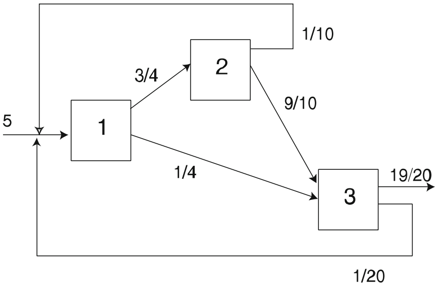


Fig. 5.4 Second example of a non-serial factory network

$$\begin{aligned} 1\lambda_1 - 0.10\lambda_2 - 0.05\lambda_3 &= 5 \\ -0.75\lambda_1 + 1\lambda_2 + 0\lambda_3 &= 0 \\ -0.25\lambda_1 - 0.90\lambda_2 + 1\lambda_3 &= 0\,, \end{aligned}$$

which has the unique solution

$$\lambda_1 = 5.690,\ \lambda_2 = 4.267,\ \lambda_3 = 5.263\,.$$

Thus, the first workstation receives 5.690 jobs per hour; 5 of these from the external source and the remaining 0.690 jobs from Workstations 2 and 3. The second workstation receives 4.267 jobs per hour, all of these from Workstation 1. The third workstation receives a total of 5.263 jobs per unit time as the combined inflow from Workstations 1 and 2. □

- *Suggestion: Do Problems 5.12–5.16.*

5.4.2 Computing Squared Coefficients of Variation for Arrivals

To obtain the squared coefficients of variation for the composite arrival stream into each workstation, a system of linear equations relating all of these coefficients must be solved; thus, the solution procedure is similar to obtaining the net inflow rates, although the individual equations are much more complex. The inflow into a given workstation, say Workstation j, is made up of the proportions of the departure stream from those workstations that feed into j along with any external stream that comes directly to j. The flow of jobs from Workstation k that are routed directly to Workstation j will be called the $k \to j$ stream and the squared coefficient of variation of inter-arrival times to j from k will be denoted by $C_a^2(k,j)$. The squared coefficient of variations for the inter-arrival times of jobs arriving from an external source is denoted similarly by $C_a^2(0,j)$ with the mean arrival rate of those jobs being γ_j. Therefore, Property 5.5 indicates that the squared coefficient of variation for the inter-arrival times satisfies the following:

$$C_a^2(j) = \frac{\gamma_j}{\lambda_j} C_a^2(0,j) + \sum_{k=1}^{n} \frac{\lambda_k p_{k,j}}{\lambda_j} C_a^2(k,j) \, , \tag{5.5}$$

where P is the routing matrix and the mean rates, λ_i, come from Property 5.7. (Frequently, $\gamma_j = 0$ and this component has no contribution.) Property 5.6 gives the relationship between departures and arrivals so that (5.5) is rewritten as

$$C_a^2(j) = \frac{\gamma_j}{\lambda_j} C_a^2(0,j) + \sum_{k=1}^{n} \frac{\lambda_k p_{k,j}}{\lambda_j} \left(p_{k,j} C_d^2(k) + 1 - p_{k,j} \right) \, . \tag{5.6}$$

The above system of equations involves both arrival stream and departure stream characterizations; thus, the final step is to express the departure streams in terms of the arrival streams using Property 5.2 and substitute this back into (5.6). You should be able to show that the resulting system of equations is as follows:

Property 5.8. *Consider a general network of n workstations with switching rule defined by the routing matrix P and assume that the sum of at least one row of P is strictly less than one. The characteristics of the flow of external jobs to Workstation j are given by γ_j and $C_a^2(0,j)$. The total mean rate of jobs coming into Workstation j is given by λ_j (from Property 5.7) and the workstation consists of c_j servers processing one job at-a-time. Each server within Workstation j has a mean service time of $E[T_j]$ and squared coefficient of variation for service of $C_s^2(j)$ with workstation utilization factor $u_j = E[T_j]\lambda_j / c_j < 1$. The values of $C_a^2(j)$ for $j = 1, \cdots, n$ that satisfy*

$$\begin{aligned} C_a^2(j) = \frac{\gamma_j}{\lambda_j} C_a^2(0,j) + \sum_{k=1}^{n} \frac{\lambda_k p_{k,j}}{\lambda_j} \Bigg[& p_{k,j}(1-u_k^2) C_a^2(k) \\ & + p_{k,j} u_k^2 \left(\frac{C_s^2(k) + \sqrt{c_k} - 1}{\sqrt{c_k}} \right) + 1 - p_{k,j} \Bigg] \quad \textit{for } j = 1, \cdots, n \end{aligned}$$

are the squared coefficients of variation for the inter-arrival times of jobs entering the various workstations.

Because the formula for determining the squared coefficient of variation of merging arrival streams is an approximation and in some cases the formula for the squared coefficient of variation for inter-departure times is an approximation, the terms obtained from Property 5.8 are approximations. The system of equations given by the property can be solved fairly rapidly by an iterative procedure known as successive substitution. The idea is to initialize the $C_a^2(i)$ terms at some arbitrary value, say 1.0, and then use these values in the right-hand side of the system in Property 5.8 which will yield new values for the $C_a^2(i)$ terms. After the new values are obtained, these new values are used for the next iteration for the right-hand side of the equations again to obtain new values. This is repeated several times until the new values obtained on the left-hand side are equal (within some specified degree of accuracy) to the values placed on the right-hand side of the equation. This method is illustrated in Examples 5.7 and 5.8 that follow the next property.

Since the system of equations in Property 5.8 is a linear system, a matrix solution is also available as given by the next property.

Property 5.9. *Consider the workstation network described in Property 5.8. Let $\mathbf{c}_a^2$ denote the vector of squared coefficients of variation for the arrival streams; that is, $\mathbf{c}_a^2 = (C_a^2(1), \cdots, C_a^2(n))$ and*

$$\mathbf{c}_a^2 \approx \left(I - Q^T\right)^{-1} \mathbf{b}\,,$$

where I is an $n \times n$ identity matrix, the elements of Q are given by

$$q_{k,j} = \frac{\lambda_k p_{k,j}^2 (1 - u_k^2)}{\lambda_j}$$

and the elements of the **b** *are given by*

$$b_j = \frac{\gamma_j}{\lambda_j} C_a^2(0, j) + \sum_{k=1}^{n} \frac{\lambda_k p_{k,j}}{\lambda_j} \left(p_{k,j} u_k^2 \frac{C_s^2(k) + \sqrt{c_k} - 1}{\sqrt{c_k}} + 1 - p_{k,j} \right) .$$

To analyze a general network, the mean arrival rate into each workstation is first determined, then workstation utilization factors are calculated since these depend on the just computed arrival rates, and finally the squared coefficients of variation for the arrival streams are computed either by a successive substitution iteration or by finding the inverse matrix. At this point, the network can be decomposed and each workstation treated individually. Finally, these results are combined to estimate the performance characteristics of the system as a whole. The following is a summary of the solution procedure used to fully develop a general factory model, obtain the values of the unknown parameter sets, and derive the relevant performance measures.

1. Workstation mean flow rates of jobs (and thus also their reciprocals, the mean flow times) are obtained through the system of equations given in Property 5.7.
2. Workstation offered workloads and utilization factors are calculated next, where the offered workload is the mean flow rate multiplied by the mean processing time and the utilization factor is the offered workload divided by the number of available servers in the workstation. (Utilization factors must be strictly less than one for steady-state conditions to hold.)
3. Workstation squared coefficients of variation of the inter-arrival times are obtained either through successive substitution using the system of equations in Property 5.8 or the matrix solution of Property 5.9.

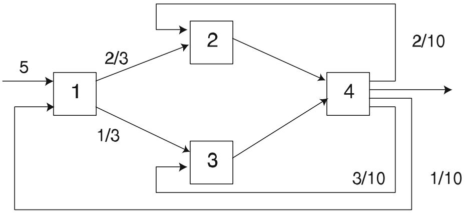


Fig. 5.5 Factory topology used in Example 5.7

4. The decomposition principle is used to obtain the mean time spent in the queue at each workstation using either Property 3.3 or 3.6. The mean service time is added to the time in queue to obtain the mean workstation cycle time and then Little's Law (Property 2.1) is used to obtain workstation WIP.
5. Factory WIP is obtained by summing the individual workstation WIPs, then the total mean cycle time for a job within the factory is derived from the application of Little's Law again. Factory throughput is merely the sum of the external inflows into the system, under the assumption of the existence of steady-state and no turning away of jobs.

This analysis process is illustrated with two examples starting with a system of single server workstations, each processing a single job at a time. The second example has a mixture of single and multiple server workstations.

Example 5.7. Consider a factory that consists entirely of single-server workstations with service time data for each workstation given by Table 5.3. Arrivals from an external source enter into the factory at the first workstation, and the arrivals are exponentially distributed with a mean rate of 5 jobs per hour. After initial processing, 2/3 of the jobs are sent to Workstation 2 and 1/3 are sent to Workstation 3. After the second step of processing, jobs are tested at Workstation 4, and only 40% of the jobs are found to be acceptable. Ten percent of the completed jobs fail the testing completely and are scrapped, at which time a new job is started to replace the scrapped jobs. Fifty percent of the jobs partially fail the testing and can be reworked. Sixty percent of the partial failures are sent to Workstation 3 and the others are sent to Workstation 2. After reworking, the jobs are sent again for testing at Workstation 4 with the same percentage of passing, partially failing, and completely failing the testing. (Figure 5.5 illustrates these job flows and switching probabilities.)

Table 5.3 Workstation characteristics for Example 5.7

Workstation i	$E[T_s(i)]$	$C_s^2(i)$
1	7.80 min	1.0355
2	7.80 min	1.7751
3	9.60 min	0.3906
4	3.84 min	2.4414

Management is interested in the mean cycle time for jobs, factory inventory levels, and workloads at each workstation. To answer these questions, each of the five steps detailed on page 143 are discussed in detail.

Step 1: Workstation Arrival Rates. The goal is to obtain the composite inflow rate into each workstation. These rates are functions of the external inflows into the systems and the routing characteristics of the job as illustrated in Fig. 5.5. The equations that define these rates for the example problem under consideration are:

$$\begin{aligned}
\lambda_1 &= 5 + \frac{1}{10}\lambda_4 \\
\lambda_2 &= 0 + \frac{2}{3}\lambda_1 + \frac{2}{10}\lambda_4 \\
\lambda_3 &= 0 + \frac{1}{3}\lambda_1 + \frac{3}{10}\lambda_4 \\
\lambda_4 &= 0 + \lambda_2 + \lambda_3 \,.
\end{aligned}$$

The solution to this system of equations is

$$(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (6.25, 6.667, 5.833, 12.5) \,.$$

Thus, even though there are only 5 jobs per hour that enter into the factory, the job arrival rate into Workstation 4 is 12.5 per hour. The reason for this increase is due to the high proportion of feedback of jobs that exit Workstation 4. If all jobs that exit Workstation 4 were acceptable in quality, then there would be no feedback or reworking of jobs and the inflow rate into Workstation 4 would merely be 5 jobs per hour. The 12.5/hr rate is a consequence of these feedback probabilities and the fact that a job that has been reworked can again be rejected and reworked over and over again. Since there is a 6/10 probability of a job being reworked, there is a $(6/10)^2$ chance of it being reworked twice, and a $(6/10)^3$ chance of being reworked three times, etc. Since the mean number of jobs that eventually enter Workstation 4 follows a geometric series, we could obtain the mean arrival rate for the workstation by

$$5\left(1 + \frac{6}{10} + \left(\frac{6}{10}\right)^2 + \left(\frac{6}{10}\right)^3 + \cdots\right) = 5\left(\frac{1}{1-0.6}\right) = 12.5 \,.$$

This type of series analysis is not necessary since the system of linear equations accounts for the total feedback effect.

Step 2: Workstation Utilizations. The offered workload to each workstation is the mean job arrival rate multiplied by the mean processing time per job which then equals the utilization factor since each workstation has only one processor. This analysis is displayed in Table 5.4 including two factors (squared utilization terms) that will be needed.

The resulting utilization factors are all in the 80% to 90% range. If the offered workload were greater than one, the number of machines would need to be increased to insure that the utilization factor is less than one. Otherwise, the system cannot handle the necessary workload and in the long run the queues for these workstation

Table 5.4 Workstation data: arrival rates, mean service times (in hours), and utilization terms

Workstation i	λ_i	$E[T_s(i)]$	u_i	u_i^2	$1-u_i^2$
1	6.250/hr	0.130 hr	0.8125	0.6602	0.3398
2	6.667/hr	0.130 hr	0.8667	0.7512	0.2488
3	5.833/hr	0.160 hr	0.9333	0.8710	0.1290
4	12.50/hr	0.064 hr	0.8000	0.6400	0.3600

will grow indefinitely. This violates the steady-state assumption underlying all our models and further analysis could not be performed.

Step 3: Squared Coefficients of Variation. The equations defining the squared coefficients of variations of the job inter-arrival times for each workstation are much more complicated that the equations needed to determine the mean flow rates. However, because the equations are still linear, their solution is straight-forward. We first demonstrate the successive substitution scheme for solving the system of equations from Property 5.8. First observe that $\gamma_2 = \gamma_3 = \gamma_4 = 0$ and that since the external arrival stream to the first workstation is exponential, we have $\gamma_1 = 5$/hr and $C_a^2(0,1) = 1$. Letting all numbers be in terms of hours, Property 5.8 yields

$$C_a^2(1) = \frac{5}{6.25} + \frac{12.5(0.1)}{6.25}\left[\frac{1}{10}\left(0.36C_a^2(4) + 0.64 \times 2.4414\right) + \frac{9}{10}\right]$$

$$\begin{aligned} C_a^2(2) = {} & \frac{6.25(0.6667)}{6.6667}\left[\frac{2}{3}\left(0.3398C_a^2(1) + 0.6602 \times 1.0355\right) + \frac{1}{3}\right] \\ & + \frac{12.5(0.2)}{6.6667}\left[\frac{2}{10}\left(0.36C_a^2(4) + 0.64 \times 2.4414\right) + \frac{8}{10}\right] \end{aligned}$$

$$\begin{aligned} C_a^2(3) = {} & \frac{6.25(0.3333)}{5.8333}\left[\frac{1}{3}\left(0.3398C_a^2(1) + 0.6602 \times 1.0355\right) + \frac{2}{3}\right] \\ & + \frac{12.5(0.3)}{5.8333}\left[\frac{3}{10}\left(0.36C_a^2(4) + 0.64 \times 2.4414\right) + \frac{7}{10}\right] \end{aligned}$$

$$\begin{aligned} C_a^2(4) = {} & \frac{6.6667(1)}{12.5}\left[1\left(0.2488C_a^2(2) + 0.7512 \times 1.7751\right) + 0\right] \\ & + \frac{5.8333(1)}{12.5}\left[1\left(0.1290C_a^2(3) + 0.8710 \times 0.3906\right) + 0\right] . \end{aligned}$$

Simplifying terms and rewriting the equations produces the following system.

$$\begin{aligned} C_a^2(1) &= 0.0072C_a^2(4) + 1.0112 \qquad (5.7) \\ C_a^2(2) &= 0.1416C_a^2(1) + 0.0270C_a^2(4) + 0.9104 \end{aligned}$$

$$\begin{aligned} C_a^2(3) &= 0.0405\,C_a^2(1) + 0.0694\,C_a^2(4) + 1.0708 \\ C_a^2(4) &= 0.1327\,C_a^2(2) + 0.0602\,C_a^2(3) + 0.8699\,. \end{aligned}$$

To use the successive substitution algorithm on the (5.7), first set

$$\mathbf{c}^2_{a-step\,1} = (\,C_a^2(1), C_a^2(2), C_a^2(3), C_a^2(4)\,)_{step\,1} = (1,1,1,1)\,.$$

After one step of the algorithm, we have

$$\mathbf{c}^2_{a-step\,2} = (1.0184, 1.0790, 1.1807, 1.0628)\,.$$

The next step gives

$$\mathbf{c}^2_{a-step\,3} = (1.0189, 1.0833, 1.1858, 1.0628)\,.$$

By the fifth iteration, the values for the squared coefficients of variation converge to

$$\mathbf{c}^2_{a-step\,5} = (1.0190, 1.0840, 1.1874, 1.0852)\,.$$

If Excel, or other software containing matrix inversion procedures, is available so that matrix inverses are easy, we could use Property 5.9 that gives

$$\mathbf{c}_a^2 = \begin{pmatrix} 1 & 0 & 0 & -0.0072 \\ -0.1416 & 1 & 0 & -0.0270 \\ -0.04045 & 0 & 1 & -0.0694 \\ 0 & -0.1327 & -0.0602 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1.0112 \\ 0.9104 \\ 1.0708 \\ 0.8699 \end{pmatrix} = \begin{pmatrix} 1.0190 \\ 1.0840 \\ 1.1874 \\ 1.0852 \end{pmatrix}\,.$$

Step 4: Decomposition. With the determination of arrival rates and the squared coefficients of variation, each workstation is analyzed as if it were an isolated workstation. Equation (3.19) is used to obtain the workstation mean cycle time and then Little's Law is used to obtain the workstation's *WIP*. These computations are:

$$\begin{aligned} CT(1) &= \left(\frac{1.0191 + 1.0355}{2}\right)\left(\frac{0.8125}{1 - 0.8125}\right)(0.130) + 0.130 = 0.709 \text{ hr} \\ WIP_s(1) &= 0.709 \times 6.25 = 4.429 \end{aligned}$$

$$\begin{aligned} CT(2) &= \left(\frac{1.0840 + 1.7751}{2}\right)\left(\frac{0.8667}{1 - 0.8667}\right)(0.130) + 0.130 = 1.338 \text{ hr} \\ WIP_s(2) &= 1.338 \times 6.6667 = 8.920 \end{aligned}$$

$$\begin{aligned} CT(3) &= \left(\frac{1.1874 + 0.3906}{2}\right)\left(\frac{0.9333}{1 - 0.9333}\right)(0.160) + 0.160 = 1.927 \text{ hr} \\ WIP_s(3) &= 1.927 \times 5.8333 = 11.243 \end{aligned}$$

$$CT(4) = \left(\frac{1.0852+2.4414}{2}\right)\left(\frac{0.8}{1-0.8}\right)(0.064)+0.064 = 0.515 \text{ hr}$$
$$WIP_s(4) = 0.5154 \times 12.5 = 6.443 \,.$$

Step 5: Factory Performance Measures. The factory throughput rate must equal to the inflow rate; therefore, $th_s = 5$/hr. The work-in-process for the whole factory is the sum of the individual workstation work-in-process numbers; therefore, $WIP_s = 31.03$, and Little's Law yields the mean cycle time; namely, $CT_s = 31.03/5 = 6.206$ hr. Notice that CT_s is greater than the sum of the individual workstation cycle times because most jobs visit some of the workstations more than once. □

Example 5.8. Reconsider the factory of the previous example as represented in Fig. 5.5 except that Workstation 3 has been changed. Workstation 3 now has two machines, each with a mean service time of 16.8 minutes with a squared coefficient of variation of 0.7653. Although the machines are slightly slower, the processing rate of the workstation is faster since there are two machines but the variability of the individual machines is increased. These data are shown in Table 5.5.

Table 5.5 Workstation characteristics for Example 5.8

Workstation i	$E[T_s(i)]$	$C_s^2(i)$	c_i
1	0.130 hr	1.0355	1
2	0.130 hr	1.7751	1
3	0.280 hr	0.7653	2
4	0.064 hr	2.4414	1

The external arrival rate and the switching probabilities have not changed; therefore, the workstation mean arrival rates remain as

$$(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (6.25, 6.6667, 5.8333, 12.5) \,.$$

Since the mean arrival rates are the same in the previous example, the three unchanged workstations having the same utilization factors. Workstation 3, however, now has two servers, $c_3 = 2$, with a different mean service times so the utilization factor is recalculated as

$$u_3 = \lambda_3 E[T_s(3)]/c_3 = \frac{5.8333(0.28)}{2} = 0.8167 \,.$$

Since the service mechanism is changed for Workstation 3, its departure process will be changed which directly effects the arrival process for Workstation 4; therefore, the defining equation for $C_a^2(4)$ will be changed. The departure stream from Workstation 3 does not directly flow into any other workstation so all other defining equations for the squared coefficients of variation remain the same. This new equation for $C_a^2(4)$ is

$$C_a^2(4) = \frac{6.6667(1)}{12.5}\left[1\left(0.2488\,C_a^2(2)+0.7512\times 1.7751\right)\right] + \frac{5.8333(1)}{12.5}\left[1\left(0.3330\,C_a^2(3)+0.6670\,\frac{0.7653+\sqrt{2}-1}{\sqrt{2}}\right)\right].$$

which reduces to

$$C_a^2(4) = 0.1327\,C_a^2(2)+0.1554\,C_a^2(3)+0.9708\,. \tag{5.8}$$

Replacing the fourth equation in the system defined by Eqs. (5.7) with Eq. (5.8) yields the new coefficients of variation given by

$$\mathbf{c}_a^2 = (1.0206, 1.0901, 1.2025, 1.3023)\,.$$

These values are only slightly changed for Workstations 1, 2, and 3, but significantly increased for Workstation 4. This difference is due to the multiple server characteristic of Workstation 3 and the change in the squared coefficient of variation for the service time at the Workstation 3 machines.

The performance measures at the workstation level for this example are displayed below. Note that the cycle time estimate for the third workstation is now based on the multiple-server approximation from Property 3.6.

$$\begin{aligned} CT(1) &= \left(\frac{1.0206+1.0355}{2}\right)\left(\frac{0.8125}{1-0.8125}\right)(0.130)+0.130 = 0.709 \text{ hr}\\ WIP_s(1) &= 0.709\times 6.25 = 4.432 \end{aligned}$$

$$\begin{aligned} CT(2) &= \left(\frac{1.0900+1.7751}{2}\right)\left(\frac{0.8667}{1-0.8667}\right)(0.130)+0.130 = 1.341 \text{ hr}\\ WIP_s(2) &= 1.341\times 6.6667 = 8.937 \end{aligned}$$

$$\begin{aligned} CT(3) &= \left(\frac{1.2025+0.7653}{2}\right)\left(\frac{0.8167^{\sqrt{6}-1}}{2(1-0.8167)}\right)(0.280)+0.280 = 0.840 \text{ hr}\\ WIP_s(3) &= 0.840\times 5.8333 = 4.901 \end{aligned}$$

$$\begin{aligned} CT(4) &= \left(\frac{1.3023+2.4414}{2}\right)\left(\frac{0.8}{1-0.8}\right)(0.064)+0.064 = 0.543 \text{ hr}\\ WIP_s(4) &= 0.543\times 12.5 = 6.790\,. \end{aligned}$$

The factory level measures become $th_s = 5$/hr, $WIP_s = 25.06$, $CT_s = 25.06/5 = 5.012$ hr. □

- *Suggestion: Do Problems 5.17–5.22.*

Appendix

The appendix of Chap. 3 presented a relatively easy method for simulating a single workstation containing one processor. In addition the appendix also discussed the use of Excel in solving linear systems of equations. In this chapter, we extend these concepts to networks of workstations.

Simulation for a Network of Single-Server Workstations. The use of Excel for simulating a network of single-server workstations will be demonstrated using Example 5.3. The concept of the network simulation is to use the equation for the queueing time delay (Eq. 3.22) and include specific times for arrivals and departures. Thus, our spreadsheet model is very similar to the spreadsheet example on Page 99 with some extra columns. In the formulas used below, note that all times are in terms of minutes and the data are the same as used for Example 5.3, namely, a factory with a serial topology of three workstations is to be simulated.

	A	**B**	**C**
1	Inter Arrival-1	Arrive Time-1	Service Time-1
2	0	0	=GAMMAINV(RAND(),0.5,24)
3	=GAMMAINV(RAND(),1.3333,11.25)	=B2+A3	=GAMMAINV(RAND(),0.5,24)

	D	**E**	**F**	**G**
1	Que Delay-1	Depart Time-1	Inter Arrive-2	Service Time-2
2	0	=B2+C2+D2	=E2	=GAMMAINV(RAND(),1.4286,6.3)
3	=MAX(0,D2+C2-A3)	=B3+C3+D3	=E3-E2	=GAMMAINV(RAND(),1.4286,6.3)

	H	**I**	**J**	**K**
1	Que Delay-2	Depart Time-2	Inter Arrive-3	Service Time-3
2	0	=E2+G2+H2	=I2	=-13.2*LN(RAND())
3	=MAX(0,H2+G2-F3)	=E3+G3+H3	=I3-I2	=-13.2*LN(RAND())

	L	**M**	**N**
1	Que Delay-3	Depart Time-3	System Cycle Time
2	0	=I2+K2+L2	M2-B2
3	=MAX(0,L2+K2-J3)	=I3+K3+L3	=M3-B3

Notice that exponential random variates are used for the service times in the third workstation (Column K) since a gamma distribution with a coefficient of variation of 1.0 is an exponential distribution. Also, the spreadsheet can be made slightly more compact by using “Wrap Text” in the first row, and increasing the height of the first row. As before, the cells A3:N3 should be copied down several thousands of rows to simulate the system. Finally the average of the values in Column N will yield the associated estimate for the system mean cycle time.

Equation Generation using Excel. The use of Properties 5.7 and 5.9 is straight forward, but it can be tedious to implement because the matrix Q and vector $\mathbf{b}$ of Property 5.9 involve several terms. If Excel is going to be used for determining the inverse, then it can also be used to help generate the coefficients. The Excel example that follows is the solution to Example 5.7. In order to clearly identify the various matrices and vectors in the spreadsheet, we label each matrix by placing its identifier just to the left of the first row, and we label each vector by placing its identifier to the top of column. Also, remember that to use an Excel function that produces an array as output, the "Shift-Control" keys must be pressed when the "Enter" key is used.

In the following, we shall make use of Excel's naming ability because it will make it easier to use some of the matrix functions. To name a range of cells, highlight the range and while the cells are highlighted, type the name in the "Name Box" which is towards the upper left of the screen, namely, the area immediately above Columns A and B of the spreadsheet. Using this Name Box define the range B1:E4 to be named `Identity`; define the range B6:E9 to be named `pMatrix`; define the range B12:E15 to be named `qMatrix`; define F6:F9 to be named `gamVector`; and define the range G17:G20 to be named `bVector` It also helps visually to place a border around these three ranges to easily identify the matrices. In the cell A1 type `I`; in the cell A6 type `P`; and in the cell A12 type `Q`. In the range B1:E4 type the identity matrix; namely, type `1` in B1, C2, D3, and E4, and type 0 in the other cells within the range. Type the switching probabilities in the B6:E9 range; namely, it should look as follows.

	B	**C**	**D**	**E**
6	0	0.6667	0.3333	0
7	0	0	0	1
8	0	0	0	1
9	0.1	0.2	0.3	0

The remainder of the basic data should follow to the right of the switching probability matrix as follows.

	F	**G**	**H**	**I**
5	`Gamma`	`C(0,k)^2`	`E[Ts]`	`Cs^2`
6	5	1	0.13	1.0355
7	0	1	0.13	1.7751
8	0	1	0.16	0.3906
9	0	1	0.064	2.4414

Step 1 of our calculations is to obtain the total arrival rates using Property 5.7. This will involve matrix arithmetic with an array output, so we must first highlight the cells that will contain the answer. Therefore, highlight the range J6:J9 and type

```
=MMULT(MINVERSE(Identity-TRANSPOSE(pMatrix)),gamVector)
```

and while holding down shift-control, hit the "Enter" key. In order to easily identify the resulting vectors, in cell J5 type `Lambda`, in cell K5 `Util`, and in cell L5 type `Util^2`. Step 2 of the calculations is to obtain the utilization factors. It is also convenient to have the squared terms available so in cell K6 type `=J6*H6` and in cell L6 type `=K6*K6`, and then copy these formulas down through cells K9 and L9.

Before generating the coefficients required for obtaining the coefficients of variation of the arrival streams, it is convenient to copy the arrival rates to the cells below the routing matrix as follows.

	A	B	C	D	E
10	`Lambda`	`=J6`	`=J7`	`=J8`	`=J9`

The Q matrix of Property 5.9 is obtained by typing the following formula in cell B12

```
=$J6*B6*B6*(1-$L6)/B$10
```

and then copy the formula to the right through cell E12 and down through cells B15:E15. It is important to include the $ in exactly the same location as is shown above since some terms refer to rows and some terms refer to columns.

Before obtaining the vector **b** of Property 5.9, it is best to calculate a B matrix and then sum the columns to obtain **b**. To accomplish this type the following in cell B17.

```
=$J6*B6*(B6*$L6*$I6+1-B6)/B$10
```

and then copy the formula to the right through cell E17 and down through cells B20:E20. The vector **b** can now be obtained from the column sums by typing the following:

	G
17	`=F6*G6/J6+SUM(B17:B20)`
18	`=F7*G7/J7+SUM(C17:C20)`
19	`=F8*G8/J8+SUM(D17:D20)`
20	`=F9*G9/J9+SUM(E17:E20)`

Notice that a copy-down command will not work from cell G17 because each sum is a column sum and not a row sum. The squared coefficients of variation for each workstation's inter-arrival times is now obtained by the following matrix operation that is typed into cell J17 after highlighting J17:J20

```
=MMULT(MINVERSE(Identity-TRANSPOSE(qMartix)),bVector)
```

and then using the control-shift keys while hitting the "Enter" key. The remainder of the performance measures should now be straight-forward. For example, the mean time spent waiting for service in the first workstation would be given by the formula `=0.5*(J17+I6)*K6*H6/(1-K6)`.

Problems

5.1. A workstation has a workload that uses 85% of its single machine capacity. Arrivals to the workstation are exponentially distributed and the service time SCV is 1.5. What is the estimated SCV of the departure stream?

5.2. A two-machine workstation has a utilization factor of 80%. The arrival stream SCV is 2.0 and the service time is Erlang-2. What is the estimated SCV of the departure stream?

5.3. Find the system performance measures of CT_s, WIP_s, and throughput for a pure serial system consisting of three single capacity workstations. The arrival rate to the system is 3 jobs per hour, with the inter-arrival time being exponentially distributed. The processing time data:

Workstation i	$E[T_i]$	$C^2[T_i]$
1	0.25 hr	4
2	0.29 hr	3
3	0.30 hr	2

5.4. Resolve Problem 3 under the assumption that the machine in each workstation is subject to breakdowns. The necessary data the processor at each workstation are given in the following table (reference Section 4.2).

Workstation #	Availability	$E[R]$	$C^2[R]$
1	0.85	1 hr	1.50
2	0.90	1 hr	1.75
3	0.95	1 hr	2.00

5.5. Find the system performance measures of CT_s, WIP_s, and throughput for a three-workstation pure serial system. The mean arrival rate to the system is one job every two hours with an SCV of 2.0. The processing time data for the three single-capacity workstations are given below. Assume that the machines are available 100% of the time.

Workstation i	$E[T_i]$	$C^2[T_i]$
1	1.6 hr	0.75
2	1.5 hr	1.50
3	1.7 hr	2.00

5.6. Find the system performance measures of CT_s, WIP_s, and throughput for a three workstation pure serial system. The arrival rate to the system is one job every two hours with an SCV of 2.0. The machine data for the three single-capacity workstations are given below.

Workstation i	$E[T_i]$	$C^2[T_i]$	Availability	$E[R]$	$C^2[R]$
1	1.6 hr	0.75	0.85	2.0 hr	1.30
2	1.5 hr	1.50	0.90	2.5 hr	1.50
3	1.7 hr	2.00	0.90	3.0 hr	1.75

5.7. Develop a spreadsheet model to solve Problem 5.5.

5.8. Develop a spreadsheet model to solve Problem 5.6.

5.9. Consider again the serial flow factory of Problem 5.5. Management expects there to be a slow increase in demand (i.e., arrival rates) over the next few years.
(a) In order to help management plan for the future, find the system performance measures (CT_{sys} and WIP_{sys}) for arrival rates of 0.51/hr, 0.53/hr, and 0.55/hr.

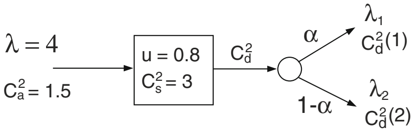


Fig. 5.6 Diagram for Problem 5.11

(b) The engineering department is considering some machine changes that will reduce the processing time variance for the bottleneck machine (i.e., the machine with the largest cycle time). Assuming the arrival rate increases to 0.51/hr, what percentage reduction in the C_s^2 for the bottleneck machine is necessary so that the average system cycle time remains the same as for the original system with an arrival rate of 0.5/hr?

(c) It turns out that reducing the processing time variance is not possible; however, it is possible to reduce the mean service time while the coefficient of variation remains according to the original system. Assuming that the arrival rate increases to 0.55/hr, what mean service rate is necessary for the bottleneck machine so that the average system cycle time remains the same as for the original system with an arrival rate of 0.5/hr?

(d) If the mean service time of the bottleneck machine is reduced enough, the bottleneck will "move" to a different machine. With an arrival rate of 0.55/hr, what is the mean service time of the current bottleneck machine that is required so that two workstations become "tied" for the bottleneck location?

5.10. Consider a three-workstation serial system, with one machine in workstations one and three and two machines available in workstation two. The external flow enters workstation one, with parameters of $\lambda_1 = 4$ jobs per hour and $C_a^2(1) = 0.75$, and proceeds sequentially through workstation two and then workstation three (i.e., a serial system). The processing time data for the three workstations are given below. Find the system performance measures of CT_s, WIP_s, and throughput for this system. To accomplish this, you need to compute, for each workstation i, $C_d^2(i)$, $CT(i)$, and $WIP_s(i)$.

Workstation i	$E[T_i]$	$C^2[T_i]$
1	12 min	2.0
2	18 min	0.7
3	13.2 min	1.0

5.11. Solve the spitting branch problem for the unknowns $(C_d^2, \lambda_1, \lambda_2, C_d^2(1), C_d^2(2))$ for three different values of branching probabilities $\alpha = (1/3, 1/2, 3/4)$ as shown in Fig. 5.6.

5.12. Solve the merging branch problem illustrated in Fig. 5.7 for the unknowns.

5.13. Obtain the mean flow rates for the system illustrated in Fig. 5.8.

5.14. Obtain the mean flow rates for the system illustrated in Fig. 5.9.

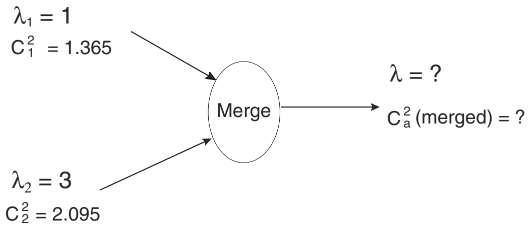


Fig. 5.7 Diagram for Problem 5.12

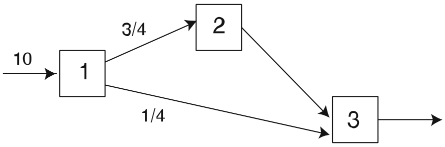


Fig. 5.8 Diagram for Problem 5.13

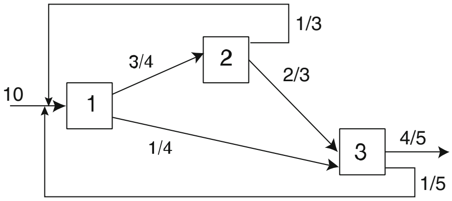


Fig. 5.9 Diagram for Problem 5.14

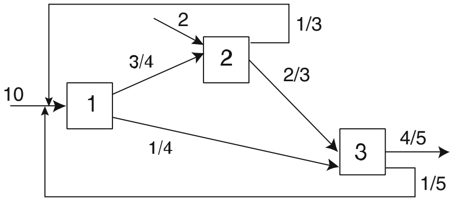


Fig. 5.10 Diagram for Problem 5.15

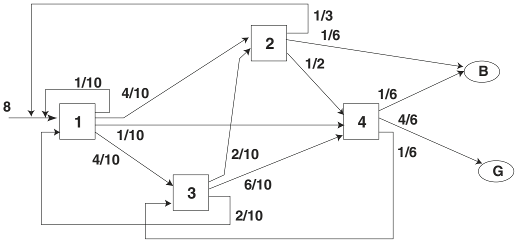
5.15. Obtain the mean flow rates for the system illustrated in Fig. 5.10.

5.16. For the network illustrated in Fig. 5.11, find the total inflows (arrival) rates for each workstation and terminator (B and G). Terminator G represents good jobs and Terminator B represents bad product. What is the probability that a job ends up good?

5.17. Using a spreadsheet program such as Excel, solve Problem 5.15.

5.18. Reconsider Problem 5.13 using the following service time data for each single-server workstation and assuming that the squared coefficient of variation of the inter-arrival times for the jobs arriving from an external source is 1.5.
(a) Compute the system performance measures of throughput, cycle time and work-in-process for this network.

Fig. 5.11 Diagram for Problem 5.16



Workstation i	$E[T_s(i)]$	$C_s^2(i)$
1	0.086	1.3521
2	0.110	0.8264
3	0.080	1.5625

(b) Compute the system performance measures of throughput, cycle time and work-in-process for this network given that the machines have breakdowns. The availability data and parameters for the repair time, random variable R, by workstation are given in the following table.

Workstation #	Availability	$E[R]$	$C^2[R]$
1	0.95	0.2	1
2	0.93	0.3	1
3	0.87	0.4	1

5.19. Reconsider Problem 5.15 using the following service time data for each workstation and assuming that the squared coefficient of variation of the inter-arrival times for the jobs arriving from an external source is 1.5.

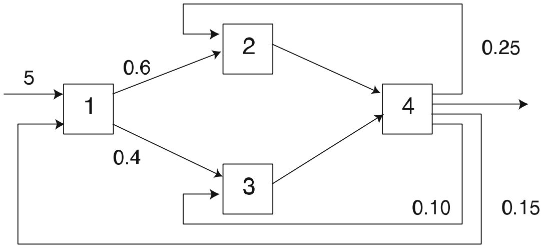
(a) Compute the system performance measures of throughput, cycle time and work-in-process for this network.

Workstation i	$E[T_s(i)]$	$C_s^2(i)$	c_i
1	0.086	1.3521	2
2	0.110	0.8264	2
3	0.080	1.5625	2

(b) Compute the system performance measures of throughput, cycle time and work-in-process for this network given that the machines have breakdowns. The availability data and parameters for the repair time, random variable R, by workstation are given in the following table.

Workstation #	Availability	$E[R]$	$C^2[R]$
1	0.95	0.2	1
2	0.93	0.3	1
3	0.87	0.4	1

Fig. 5.12 Diagram for Problem 5.20



5.20. Consider the factory model illustrated in Fig. 5.12 with $C_a^2(0,1) = 2$ and the workstation service time data displayed below. Compute the system performance measures of throughput, cycle time and system work-in-progress assuming that there is only one machine at each workstation.

Workstation i	$E[T_s(i)]$	$V[T_s(i)]$
1	0.13	0.02
2	0.13	0.03
3	0.20	0.04
4	0.08	0.01

5.21. Using a spreadsheet program such as Excel, solve Problem 5.18(b).

5.22. Using a spreadsheet program such as Excel, solve Problem 5.19(b).

5.23. Using a spreadsheet program such as Excel, solve Problem 5.20.

References

1. Albin, S.L. (1984). Approximating a Point Process by a Renewal Process, II: Superposition Arrival Processes to Queues. *Operations Research*, **30**:1133–1162.
2. Burke, P.J. (1968). The Output Process of a Stationary M/M/s Queueing System. *Annuals Math. Stat.*, **39**:1144–1152.
3. Buzacott, J.A., and Shanthikumar, J.G. (1963). *Stochastic Models of Manufacturing Systems*. Prentice Hall, Englewood Cliffs, NJ.
4. Marshall, K.T. (1968). Some Inequalities in Queueing, *Operations Research*, **16**:651–665.
5. Whitt, W. (1982). Approximating a Point Process by a Renewal Process, I: Two Basic Methods. *Operations Research*, **30**:125–147.
6. Whitt, W. (1983). The Queueing Network Analyzer, *The Bell System Technical Journal*, **62**:2779–2814.

Chapter 6
Multiple Product Factory Models

Most manufacturing facilities are setup to produce more than a single product. Even in the case of single product facilities, if the product visits a workstation more than once with different processing times at each visit, then the workstation sees the equivalent of multiple products. Such revisiting production schemes, called reentrant flow systems, are prevalent in the semiconductor industry where it is not unusual for a product to be routed to the same machine group for distinct processing 20 or more times.

Modeling multiple product facilities is not significantly more difficult than single product models. There are two basic principles to keep in mind. First, the workload on a workstation is, as before, the sum of all the visits multiplied by the processing time per visit. This concept was introduced in the previous chapter (see p. 143) and since we use it in a more general setting here, we give a formal definition.

Definition 6.1. The *offered workload* (or simply the *workload*) of a workstation is the total amount of work that is required of a workstation per unit of time. The workload is determined by the sum of the total arrival rate (per hour) for each product type multiplied by its associated mean processing time (in hours). For purposes of determining workload, when a specific product type revisits a workstation, it is considered as a separate product type.

The second basic principle is that job flow needs to be maintained by product type. That is, the number of visits to each workstation by product class is needed. Different products can have different probabilistic flows through the production facility as well as different processing time characteristics. Hence, the number of visits to each workstation by product needs to be developed. This analysis requires the solution of a network flow system of equations by product. Here again as was done in the preceding chapter, the processing time is assumed to follow the same distribution for each product on each visit to a given workstation (of course due to randomness, the actual processing times will vary even though the distribution is the same). The re-entrant flow situation with different processing distributions per visit requires a different modeling paradigm that is taken up in Sect. 6.5.

G.L. Curry, R.M. Feldman, *Manufacturing Systems Modeling and Analysis*, 2nd ed.,
DOI 10.1007/978-3-642-16618-1_6, © Springer-Verlag Berlin Heidelberg 2011

6.1 Product Flow Rates

To compute the workload on the workstation, the number of visits to the workstation by each product is computed first. This requires an analysis for each product similar to that performed in Sect. 5.4.1 for a single product. A method of distinguishing between products visiting the same workstation is required. Previously a subscript was used to denote the workstation visited by a product so that λ_k denoted the arrival rate of jobs to Workstation k. Two subscripts will now be used to distinguish among the various product types; thus, $\lambda_{i,k}$ is the arrival rate of Product Type i to Workstation k. Since a single subscript refers to a workstation, we will use a superscript when a single index refers to a product type; thus, $\boldsymbol{\lambda}^i$ is a vector giving the total arrival rates of Product Type i into each workstation so that the k^{th} component of the vector $\boldsymbol{\lambda}^i$ is $\lambda_{i,k}$.

Arrivals from an external source are denoted as before by γ so that $\gamma_{i,k}$ is the external arrival rate of Product i into Workstation k. Additionally a workstation branching probability matrix for each product type will be needed. Since it is standard to already use two subscripts for this matrix of probabilities, the product type will be denoted by a superscript such as p^i_{jk} meaning the probability that an individual item of Product i leaving Workstation j goes to Workstation k. The matrix of these probabilities for Product i is denoted as P^i.

With the above notation, we can rewrite Property 5.7 so that it applies to more than one product type.

Property 6.1. *Consider a factory of n workstations where Product Type i follows the switching rule defined by the routing matrix P^i and assume that the sum of at least one row of P^i is strictly less than one (i.e., jobs exit the network from at least one workstation). Let $\boldsymbol{\gamma}^i = (\gamma_{i,1}, \cdots, \gamma_{i,n})$ denote a vector consisting of the mean arrival rate of Type i jobs from an external source to the workstations. Both P^i and $\boldsymbol{\gamma}^i$ are known. Let $\boldsymbol{\lambda}^i = (\lambda_{i,1}, \cdots, \lambda_{i,n})$ be the (unknown) vector denoting mean arrival rate of all Type i jobs to the workstations. The vector $\boldsymbol{\lambda}^i$ is given by*

$$\boldsymbol{\lambda}^i = \left(I - (P^i)^T\right)^{-1} \boldsymbol{\gamma}^i ,$$

where I is an $n \times n$ identity matrix and $(P^i)^T$ is the transpose of P^i.

Once the arrival rates for the various product types have been determined, the total arrival rate of jobs to Workstation k is given by the sum of the different product types; that is

$$\lambda_k = \sum_{i=1}^{m} \lambda_{i,k} ,$$

where m is the total number of product types within the factory.

After the product flow rates have been computed, it is straight-forward to obtain the expected number of visits to each workstation per product type. For example, if a given product type arrives from an external source at a rate of 5 per hour, but the calculated total arrival rate to a workstation is 10 per hour, it follows that each job visit the workstation an average of two times. This reasoning leads to the following property.

Property 6.2. *Consider a factory of n workstations with m different job types, and let the arrival rate of Job Type i from an external source be given by* $\sum_{k=1}^{n} \gamma_{i,k}$*. Then the expected number of visits to Workstation k by Job Type i is* $\lambda_{i,k} / \sum_{j=1}^{n} \gamma_{i,j}$*, where* $\lambda_{i,k}$ *is the arrival rate as determined by Property 6.1.*

Example 6.1. To demonstrate Property 6.1, we take advantage of two examples from the previous chapter. Consider a four workstation facility that processes two products with each product arriving to the first workstation according to individual Poisson arrival streams, each at a rate of 5 per hour. Product 1 uses only the first three workstations with the routing structure displayed in Fig. 5.4 (p. 141). Product 2 uses all four workstations with the routing structure displayed in Fig. 5.5 (p. 144). To determine the mean arrival rate to each workstation of Type 1 jobs is simply to repeat the steps of Example 5.6 yielding

$$\boldsymbol{\lambda}^1 = (5.690, 4.267, 5.263, 0) \ .$$

The calculations necessary to give the mean arrival rates for Type 2 jobs are contained in Step 1 of Example 5.7 and are

$$\boldsymbol{\lambda}^2 = (6.25, 6.667, 5.833, 12.5) \ .$$

The total rate into each workstation is merely the sum of the individual product inflows; namely $\boldsymbol{\lambda} = \sum_{i=1}^{m} \boldsymbol{\lambda}^i$, and is given as

$$\boldsymbol{\lambda} = (11.940, 10.934, 11.096, 12.5) \ .$$

The average number of visits of Job Type 1 to Workstation 1 is 1.138, but the average number of visits of Job Type 1 to the second workstation is slightly less than 1 (actually it equals 0.8534) implying that some jobs bypass Workstation 2 completely and some jobs visit the workstation more than once. The most visited workstation by a single product type is the fourth workstation that has each Job Type 2 visiting it an average of 2.5 times. □

6.2 Workstation Workloads

Once the workstation arrival rates by product type have been determined, the workload for each workstation can be computed. Again, the previously used notation for the service time is extended for product types by including a second index to denote the type; namely, the mean and squared coefficient of variation of the processing time of Product i at Workstation k are denoted as $E[T_s(i,k)]$ and $C_s^2(i,k)$, respectively. By Definition 6.1, the workload at Workstation k, WL_k, is computed as the sum of the product visits multiplied by their respectively mean processing times; that is,

$$WL_k = \sum_{i=1}^{m} \lambda_{ik} E[T_s(i,k)] \;, \tag{6.1}$$

where m is the total number of product types within the factory.

The utilization factor, u_k, for Workstation k is then the workload divided by the available capacity; thus,

$$u_k = \frac{WL_k}{c_k} = \frac{\sum_{i=1}^{m} \lambda_{i,k} E[T_s(i,k)]}{c_k} \;, \tag{6.2}$$

where c_k is the number of identical processors available at Workstation k to handle the workload.

Example 6.2. We return to Example 6.1 and assume that there is one machine at each workstation and that the processing time data for the two products are as given in Table 6.1. Since there is one machine per workstation, the workload and utilization factors are the same at each workstation so that

Table 6.1 Processing time characteristics for Example 6.2

Workstation k	$E[T_s(1,k)]$	$C_s^2(1,k)$	$E[T_s(2,k)]$	$C_s^2(2,k)$
1	1/14 hr	0.8	1/15 hr	1.33
2	1/10 hr	1.2	1/18 hr	2.00
3	1/15 hr	1.5	1/12 hr	1.50
4	—	—	0.06 hr	0.75

$$\mathbf{u} = (0.8231, 0.7971, 0.8369, 0.75) \;.$$

With utilization factors all less than 1.0, the factory can achieve steady-state and further analysis is possible. □

6.3 Service Time Characteristics

Although the determination of arrival rates under multiple product types is a simple extension of results from the previous chapter, the calculations required for the mean and squared coefficient of variation of the service time are slightly more involved and are based on the material contained within Sect. 1.6.3. Specifically, for Workstation k, the service time will be the random variable $T_s(i,k)$ whenever Product i is being processed. The service time for an arbitrary job, independent of the job type, is the random variable denoted by $T_s(k)$. In the long-run, the probability that a given machine at Workstation k will be processing a Type i job is $\lambda_{i,k}/\lambda_k$; thus, $T_s(k)$ is a mixture of random variables since

$$T_s(k) = \begin{cases} T_s(1,k) & \text{with probability } \frac{\lambda_{1,k}}{\lambda_k} \\ \vdots & \\ T_s(m,k) & \text{with probability } \frac{\lambda_{m,k}}{\lambda_k}\ , \end{cases}$$

where m is the number of products within the factory.

The mean and coefficient of variation for the service time at Workstation k can be computed using Property 1.10. That is, the mean is

$$E[T_s(k)] = \sum_{i=1}^{m} \frac{\lambda_{i,k}}{\lambda_k} E[T_s(i,k)] = \frac{WL_k}{\lambda_k}\ , \tag{6.3}$$

and the second moment is

$$E[(T_s(k))^2] = \sum_{i=1}^{m} \frac{\lambda_{ik}}{\lambda_k} E[T_s(i,k)^2]\ .$$

It is not too hard to show the identity $E[X^2] = E[X]^2(1 + C^2[X])$ which will then yield an equivalent expression for the second moment as

$$E[(T_s(k))^2] = \sum_{i=1}^{m} \frac{\lambda_{ik}}{\lambda_k} E[T_s(i,k)]^2(1 + C_s^2(i,k))\ .$$

Combining the above two equations yields an expression for the squared coefficient of variation for the service times at Workstation k when there are m product types within the factory as

$$C_s^2(k) = \frac{\sum_{i=1}^{m}(\lambda_{i,k}/\lambda_k)E[T_s(i,k)]^2(1 + C_s^2(i,k))}{\left(\sum_{i=1}^{m}(\lambda_{i,k}/\lambda_k)E[T_s(i,k)]\right)^2} - 1\ . \tag{6.4}$$

Example 6.3. We are now ready to derive the mean and squared coefficients of variation for the four workstation service times using the arrival rate data of Example 6.1 and the service time data of Example 6.2. We show the calculations necessary for the first workstation and leave it to the reader to verify the remaining three workstations.

The total arrival rate for the first workstation is 11.94/hr and thus,

$$E[T_s(1)] = \left(\frac{5.690}{11.94}\right)\frac{1}{14} + \left(\frac{6.250}{11.94}\right)\frac{1}{15} = 0.0689 \text{ hr} .$$

The computations for the squared coefficient of variation are

$$C_s^2(1) = \frac{\left(\frac{5.690}{11.94}\right)\left(\frac{1}{14}\right)^2(1+0.8) + \left(\frac{6.250}{11.94}\right)\left(\frac{1}{15}\right)^2(1+1.33)}{(0.0689)^2} - 1 = 1.0616 .$$

Note that some of the numbers used in the above equation were taken from Table 6.1. The final results for the service time characteristics for the four workstations are contained in Table 6.2.

Table 6.2 Service time characteristics for Example 6.3

Workstation k	$E[T_s(k)]$	$C_s^2(k)$
1	0.069	1.062
2	0.073	1.678
3	0.075	1.530
4	0.060	0.750

□

6.4 Workstation Performance Measures

The multiple product facility problem is now reduced to a problem similar to the single product analysis since the workstation composite service time data are now known. The workstation level variables, namely λ_k, $E[T_s(k)]$ and $C_s^2(k)$, are used in place of the individual product data. The final terms needed are the switching probabilities.

Property 6.3. *Consider a factory of n workstations with m different job types. Assume that the total arrival rate of Job Type i to Workstation k is given by $\lambda_{i,k}$, and the probability that a job of Type i leaving Workstation j will be routed to Workstation k is given by $p^i_{j,k}$. The composite routing matrix, $P = (p_{jk})$ gives the switching probabilities of an arbitrary job and is determined by*

$$p_{jk} = \frac{\sum_{i=1}^m \lambda_{ij}\, p^i_{jk}}{\lambda_j} \quad \textit{for } j,k = 1,\cdots,n .$$

Once the composite routing matrix is obtained, Property 5.8 can be used to determine the squared coefficients of variation for the workstation arrival streams, and then the composite waiting times for an arbitrary job, $CT_q(k)$ for Workstation k, can be derived using either Property 3.3 or 3.6. As long as there is no priority being given to specific job types, all jobs experience the same queue; therefore, the mean cycle time within Workstation k by Job Type i is given as

$$CT_s(i,k) = CT_q(k) + E[T_s(i,k)] \ . \tag{6.5}$$

Combining Property 6.2 with Eq. (6.5) allows for the computation of the mean time that each product type spends within the factory.

Property 6.4. *Consider a factory of n workstations with m different job types. Assume that the external arrival rate of jobs of Type i to Workstation k is given by $\gamma_{i,k}$, and the total arrival rate of Job Type i to Workstation k is given by $\lambda_{i,k}$. Furthermore assume that the mean time spent waiting for processing in Workstation k by an arbitrary job (namely, $CT_q(k)$) has been determined. Then the mean time spent within the factory by a Type i job is given by*

$$CT_s^i = \frac{\sum_{k=1}^n \lambda_{ik}(CT_q(k) + E[T_s(i,k)])}{\sum_{j=1}^n \gamma_{ij}}$$

for $i = 1, \cdots, m$.

Conditional cycle time information for individual products given their destination (such as good or bad parts) is considerably more complex and requires a Markov process modeling approach [4] beyond the scope of this book.

Example 6.4. We now complete the analysis of the factory contained in Examples 6.1–6.3. The matrix of probabilities are obtained from Property 6.3. For example, the probability of going from Workstation 2 to Workstation 1 is determined as

$$p_{21} = \frac{\lambda_{12}p_{21}^1 + \lambda_{22}p_{21}^2}{\lambda_2} = \frac{4.267(0.1) + 6.667(0)}{10.934} = 0.039 \ .$$

Continuing with the other workstations should yield

$$P = \begin{bmatrix} 0 & 0.706 & 0.294 & 0 \\ 0.039 & 0 & 0.351 & 0.610 \\ 0.024 & 0 & 0 & 0.526 \\ 0.100 & 0.200 & 0.300 & 0 \end{bmatrix} .$$

The analysis required to obtain the mean waiting times in the workstations is the same procedure as for individual product systems once the composite product data and transition probability matrix P have been developed. The squared coefficient of variation for the arrival streams into each workstation is again obtained by solving the C_a^2 system of equations (Property 5.8).

$$\begin{aligned}
C_a^2(1) &= 0.00051\,C_a^2(2) + 0.00016\,C_a^2(3) + 0.00458\,C_a^2(4) + 0.9943 \\
C_a^2(2) &= 0.17554\,C_a^2(1) + 0.02001\,C_a^2(4) + 0.8205 \\
C_a^2(3) &= 0.03\,C_a^2(1) + 0.04427\,C_a^2(2) + 0.04436\,C_a^2(4) + 0.9235 \\
C_a^2(4) &= 0.11868\,C_a^2(2) + 0.07358\,C_a^2(3) + 1.0396\,.
\end{aligned}$$

The solution to this system is

$$\mathbf{c}_a^2 = (1.0007, 1.0209, 1.0537, 1.2383)\,.$$

The cycle time by workstation is given as the composite time for all products visiting that workstation. The computations for this example are displayed in the following table.

Table 6.3 Cycle times and WIP for each workstation of Example 6.4

Workstation k	$CT_q(k)$	$CT(k)$	$WIP(k)$
1	0.331 hr	0.400 hr	4.772
2	0.387 hr	0.460 hr	5.029
3	0.502 hr	0.577 hr	6.402
4	0.183 hr	0.243 hr	3.036

The total facility performance measures are for the total work in the facility and are not distinguishable by product type. The total system work-in-process is the sum of the workstation WIP's and equals 19.238. The total inflow and, hence, throughput for the system is 10/hr. Thus, the average cycle time in the system for all items by Little's Law is 19.238/10 = 1.9238 hours.

Property 6.4 is combined with the data of Tables 6.1 and 6.3 to produce the system mean cycle times by individual product type. For this example these computations are:

$$\begin{aligned}
CT^1 = {} & [\,5.690(0.3307 + 0.0714) + 4.2674(0.3870 + 0.1) \\
& + 5.2632(0.5015 + 0.0667)\,]\,/\,5 = 1.4714 \text{ hr}
\end{aligned}$$

$$\begin{aligned}
CT^2 = {} & [\,6.25(0.3307 + 0.0667) + 6.6667(0.3870 + 0.0556) \\
& + 5.8333(0.5015 + 0.0833) + 12.5(0.1828 + 0.06)\,]\,/\,5 = 2.3763 \text{ hr}\,.
\end{aligned}$$

These two products are produced in equal quantities, so the average cycle time for the factory is the average of these two individual product cycle times or 1.9238 hours.

To demonstrate that this modeling approach is adequate for most decision making situations, these analytical results are compared with simulation results in the following table. All of the critical parameters are close enough for the analytical model to be a usable tool for decision purposes. Due to the quantity of the data, the information is given by rows for each workstation in Table 6.4. One row labeled S *i* for simulation results for Workstation *i*, and the associated analytical results in the following row labeled A *i*.

Table 6.4 Comparison of simulation and analytical results for Example 6.4

Workstation	CT	WIP	$E[T_a]$	$C^2[T_a]$	$E[T_d]$	$C^2[T_d]$
S 1	0.398	4.744	0.084	1.001	0.084	1.050
A 1	0.400	4.772	0.084	1.000	0.084	1.042
S 2	0.427	4.677	0.091	1.028	0.091	1.443
A 2	0.460	5.029	0.091	1.021	0.091	1.440
S 3	0.569	6.309	0.090	1.035	0.090	1.397
A 3	0.577	6.402	0.090	1.053	0.090	1.389
S 4	0.248	3.107	0.080	1.330	0.080	1.044
A 4	0.243	3.036	0.080	1.238	0.080	0.983
S sys	1.888	18.84	—	—	—	—
A sys	1.924	19.24	—	—	—	—

□

- *Suggestion: Do Problems 6.1–6.2 and 6.5–6.11.*

6.5 Processing Step Modeling Paradigm

To this point, all analyses have considered that every visit to a workstation was probabilistically identical to all other visits to the same workstation. In other words, the mean and standard deviation of processing time was the same whenever the same type of job visited the same workstation. Furthermore, the switching probabilities only depended on job type and not on whether or not the job was visiting the workstation for the first or the second time. There are many facilities where jobs make more than one scheduled visit to various workstations and the processing characteristics are different for the various visits. These re-entrant flow systems are prevalent in the semiconductor industry as well as many job shop production type facilities. When a job requires a different processing time distribution from visit to visit or when a job is scheduled to visit a workstation more than once, it is necessary to keep track of not only the job location but also the visit number to that location. To accomplish this extra requirement for job location control, a data description method is used that is based on the process step that the job is undergoing.

The processing step modeling paradigm is a rather straight forward method of accomplishing the informational requirements of re-entrant flow systems. The idea is to list the processing steps that a job must go through during the production process. Associated with each processing step is the information needed for processing that includes the workstation being visited and the processing time characteristics.

Hence, a product can require several processing steps yet these steps might be performed by only a few workstations. There is only a slight change in the informational requirements, but the modeling flexibility that this allows is much greater than before. The processing step paradigm is the standard industrial method of specifying product production information, except for assembly line like processes.

To use the processing step modeling paradigm, a processing step to workstation mapping is needed for each job. This is typically accomplished by using a list where the location or list index denotes the processing step and the number in that location in the list denotes the workstation. Previously a workstation list was used for specifying the processing time information. With the processing step approach, a step indexed list contains the necessary information about the processing requirements and the job's location within its processing step sequence is maintained. This is, obviously, only a slight change in the modeling approach but by focusing on the processing step instead of the workstation index allows for considerably more complex production schemes to be analyzed. The two methods yield the same result when there is a one-for-one correspondence between processing steps and workstations. However, more complex situations can be handled with this approach than were previously possible.

Definition 6.2. Consider a factory with n workstations and a job of Type i that has v_i processing steps in its production plan. The *workstation mapping function*, denoted by $\widetilde{w}^i(\ell)$ for $\ell = 1, \cdots, v_i$, gives the workstation assigned to the ℓ^{th} step of the production plan; thus $\widetilde{w}^i(\cdot)$ is an integer-valued function with range $1, \cdots, n$.

One of the difficulties on the processing step paradigm is being clear on whether a subscript or parameter refers to a workstation number or a step number. To help differentiate between a workstation function and a step function, a "tilde" will be used to indicate that a function's parameter or a variable's subscript refers to a step number.

To illustrate the processing step paradigm, consider a situation where a factory with three workstations produces two product types. Consider Table 6.5 that shows the production plan. Notice that the product flow for this example is deterministic and workstations are revisited but in different sequences depending on the job type. The sequence of workstations in which jobs of Type 1 are processed is 1, 2, 3, 1

Table 6.5 Processing data in hours in processing step form for two different products

Product 1	Step #	1	2	3	4
	Workstation #	1	2	3	1
	$E[T_s]$	3.0	7.2	1.62	2.5
	$C^2[T_s]$	1.5	2.0	0.75	1.5
Product 2	Step #	1	2	3	4
	Workstation #	1	3	2	3
	$E[T_s]$	3.2	1.45	7.0	1.0
	$C^2[T_s]$	1.0	1.75	1.7	0.45

whereas the sequence of workstation in which jobs of Type 2 are processed is 1, 3, 2, 3. As an example of the workstation mapping function, notice that $\widetilde{w}^1(2) = 2$ and $\widetilde{w}^2(2) = 3$.

It is also possible to include probabilistic branching with the production plan. However, because workstations may be visited more than once, the probabilistic branching must be given by step number and not by workstation number. Because branching probabilities may depend on step numbers, the standard routing matrix (Definition 5.2) cannot be used because it is based on workstations. Thus, a step-wise routing matrix is needed.

Definition 6.3. Consider a factory with m job types, where Job Type i has a production plan consisting of ν_i steps. The *step-wise routing matrix*, denoted by $\widetilde{P}^i$, for Job Type i is a square matrix of size $\nu_i \times \nu_i$ where $\widetilde{p}^i_{\ell,j}$ gives the probability that Job Type i will be routed to Step j after completing Step ℓ.

Example 6.5. Consider the production plan given in Table 6.6 involving a factory with three workstations. Assume that Workstations 1 and 2 are reliable but that Workstation 3 is not. There is 10% chance that jobs being processed through the third workstation for the first time (i.e., Step 2) must be returned to Workstation 1 (Step 1), and a 5% chance that jobs being processed through the third workstation for the second (i.e., Step 5) time must be returned to Workstation 2 (Step 3). In this case, the workstation mapping function is

Table 6.6 Processing step paradigm for multiple visits to workstations with the data in hours

Step #	1	2	3	4	5
Workstation #	1	3	2	1	3
$E[T_s]$	3.0	2.5	3.7	4.0	3.6
$C^2[T_s]$	1.0	0.75	1.25	1.75	1.32

$$\widetilde{w}^1(1) = 1,\ \widetilde{w}^1(2) = 3,\ \widetilde{w}^1(3) = 2,\ \widetilde{w}^1(4) = 1,\ \widetilde{w}^1(5) = 3\ ,$$

and the step-wise routing matrix is given by

$$\widetilde{P}^1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0.1 & 0 & 0.9 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0.05 & 0 & 0 \end{bmatrix} . \tag{6.6}$$

and a diagram illustrating these flows is displayed in Fig. 6.1. □

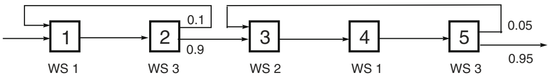


Fig. 6.1 Process flows according to production plan of Example 6.5

6.5.1 Service Time Characteristics

In order to obtain average cycle times and inventory levels within the factory, the effective service time characteristics for each workstation must be determined. These characteristics need arrival rates (see Eqs. 6.3 and 6.4) to each workstation, and an indicator function is needed so that the proper workstation can be associated with each processing step.

Definition 6.4. An *indicator function* for integers, denoted by $I(j,j)$ for i and j integers, is defined by

$$I(i,j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j\,. \end{cases}$$

Notice that an identity matrix is an indicator function where the domain for i and j are the same.

The indicator function and step-wise routing matrix are combined with obtain the total arrival rates into each workstation according to the following property.

Property 6.5. *Consider a factory of n workstations with m different job types. Job Type i has a production plan described by the workstation mapping function* $\widetilde{w}^i(\ell)$ *for* $\ell = 1, \cdots, \nu_i$. *The mean number of Type i jobs passing through each step is given by the vector* $\widetilde{\boldsymbol{\lambda}}^i$ *where*

$$\widetilde{\boldsymbol{\lambda}}^i = \left(I - (\widetilde{P}^i)^T\right)^{-1} \widetilde{\boldsymbol{\gamma}}^i\,,$$

where $\widetilde{\gamma}^i_\ell$ *is the mean arrival rate from an external source of Type i jobs to Step* ℓ. *Then the total mean arrival rate of all jobs to Workstation k is*

$$\lambda_k = \sum_{i=1}^{m} \sum_{\ell=1}^{\nu_i} \widetilde{\lambda}_{i,\ell}\, I(\widetilde{w}^i(\ell), k)\,,$$

where $\widetilde{\lambda}_{i,\ell}$ *is the mean arrival rate of Type i jobs to Step* ℓ. *Note that the components of the vector* $\widetilde{\boldsymbol{\lambda}}^i$ *are the values of* $\widetilde{\lambda}_{i,\ell}$ *for* $\ell = 1, \cdots, \nu_i$.

Note that an alternative method of writing the above sum is

$$\lambda_k = \sum_{i=1}^{m} \sum_{\ell \in \{\widetilde{w}^i(\ell)=k\}} \widetilde{\lambda}_{i,\ell} \; ;$$

namely, the effect of the indicator function is to sum only those values of $\widetilde{\lambda}_\ell^i$ for which Workstation k is associated with the ℓ^{th} step.

Each visit to a workstation by a job may have different processing requirements; therefore, to denote these differences we must extend our notation one more time. We let the random variable $\widetilde{T}_s(i,\ell)$ denote the processing time for Job Type i during the ℓ^{th} step of its production plan. The mean service time for Job Type i during Step ℓ is denoted by $E[\widetilde{T}_s(i,\ell)]$ and this occurs at the workstation designated by $\widetilde{w}^i(\ell)$. Likewise, the squared coefficient of variation of the service time is given by $\widetilde{C}_s^2(i,\ell)$. With these definitions, the workload and utilization for Workstation k (compare to Eq. 6.2) are

$$u_k = \frac{WL_k}{c_k} = \left(\sum_{i=1}^{m} \sum_{\ell=1}^{\nu_i} \widetilde{\lambda}_{i,\ell} E[\widetilde{T}_s(i,\ell)] \, I(\widetilde{w}^i(\ell),k) \right) / \, c_k \, , \tag{6.7}$$

where c_k is the number of identical processors available at Workstation k to handle the workload, m is the number of job types, and ν_i is the number of production steps for Job Type i.

The service time characteristics for Workstation k are also given similarly and are analogous to Eqs. (6.3) and (6.4):

$$E[T_s(k)] = \sum_{i=1}^{m} \sum_{\ell=1}^{\nu_i} \frac{\widetilde{\lambda}_{i,\ell}}{\lambda_k} E[\widetilde{T}_s(i,\ell)] \, I(\widetilde{w}^i(\ell),k) = \frac{WL_k}{\lambda_k} \, , \tag{6.8}$$

where λ_k comes from Property 6.5 and

$$C_s^2(k) = \frac{\sum_{i=1}^{m} \sum_{\ell=1}^{\nu_i} (\widetilde{\lambda}_{i,\ell}/\lambda_k) E[\widetilde{T}_s(i,\ell)]^2 (1 + \widetilde{C}_s^2(i,\ell)) \, I(\widetilde{w}^i(\ell),k)}{E[T_s(k)]^2} - 1 \, . \tag{6.9}$$

Example 6.6. Consider a factory with three workstations that is open 24/7 and manufactures one job type. Order for jobs are released randomly throughout the 24-hour period and it has been determined that the number of jobs ordered each day is Poisson with a mean of 4.8 jobs. All jobs begin processing at Workstation 1 and then follow the route with processing characteristics specified by Table 6.6 with branching probabilities given in Example 6.5 and defined by the step-wise routing matrix of Eq. (6.6). Since the number of arrivals per unit time is Poisson, the inter-arrival times must be exponential; therefore, the arrival stream has a squared coefficient of variation of 1.0. The 4.8 per day rate of arrival of jobs is equivalent to 0.2 arrivals per hour; thus $\widetilde{\gamma}_1 = \gamma_1 = 0.2$/hr. (Notice that we are dropping the subscript indicating the job type since there is only one type.) The application of Property 6.5 yields the following step-wise arrival rates

$\widetilde{\lambda}_1 = 0.2222/\text{hr}$, $\widetilde{\lambda}_2 = 0.2222/\text{hr}$, $\widetilde{\lambda}_3 = 0.2105/\text{hr}$, $\widetilde{\lambda}_4 = 0.2105/\text{hr}$, $\widetilde{\lambda}_5 = 0.2105/\text{hr}$,

and the following workstation arrival rates

$$\lambda_1 = 0.4327/\text{hr},\ \lambda_2 = 0.2105/\text{hr},\ \lambda_3 = 0.4327/\text{hr}.$$

The workload calculations for the three workstations are

$$\begin{aligned} WL_1 &= 0.2222 \times 3.0 + 0.2105 \times 4.0 = 1.5086 \\ WL_2 &= 0.2105 \times 3.7 = 0.7789 \\ WL_3 &= 0.2222 \times 2.5 + 0.2105 \times 3.6 = 1.3140\,. \end{aligned}$$

For a steady-state to exist, the number of machines at each workstation must be strictly greater than the workload; therefore, there must be at least two machines for Workstations 1 and 3 and one machine at Workstation 2. Assuming the minimum requirements, the workstation utilization vector is (75.4%, 78.0%, 65.7%).

The service time characteristics for Workstation 1 are calculated as

$$\begin{aligned} E[T_s(1)] &= \frac{1.5086}{0.4327} = 3.486 \text{ and} \\ C^2[T_s(1)] &= \frac{(0.2222/0.4327)(3^2)(1+1) + (0.2105/0.4327)(4^2)(1+1.75)}{3.486^2} - 1 \\ &= 1.522\,. \end{aligned}$$

Performing similar computations results in the values as displayed in Table 6.7. □

Table 6.7 The composite processing data for Example 6.6

Workstation k	c_k	u_k	$E[T_s(k)]$	$C_s^2(k)$
1	2	0.7544	3.486 hr	1.521
2	1	0.7789	3.700 hr	1.250
3	2	0.6567	3.035 hr	1.198

6.5.2 Performance Measures

The final terms that are needed to complete the factory analysis are the squared coefficients of variation for the arrival streams to each workstation. To obtain the system of equations that define these terms, the factory with multiple routing schemes will be converted to a similar factory with probabilistic routing by the following route matrix.

Property 6.6. *Consider a factory of n workstations with m different job types. Job Type i has a production plan described by the workstation mapping function $\widetilde{w}^i(\ell)$ for $\ell = 1, \cdots, \nu_i$. The workstation routing matrix, P is defined, for $k = 1, \cdots, n$, by*

$$p_{k,j} = \left(\sum_{i=1}^{m} \sum_{\ell=1}^{\nu_i} \sum_{r=1}^{\nu_i} \widetilde{\lambda}_{i,\ell} \widetilde{p}^i_{\ell,r} \, I(\widetilde{w}^i(\ell), k) \, I(\widetilde{w}^i(r), j) \right) / \lambda_k \, ,$$

where the terms $\widetilde{\lambda}_{i,\ell}$ and λ_k are determined by Property 6.5.

Our goal here is to determine the characteristics of the arrival streams to the workstations, therefore, we need the coefficient of variation for the external arrivals. Let these be denoted by $\widetilde{C}^2_a(i,0,\ell)$; in other words, $\widetilde{C}^2_a(i,0,\ell)$ is the squared coefficient of variation for the inter-arrival times of Job Type i from an external source that enter the production process at Step ℓ of the i^{th} production plan. The characteristics of the external arrival streams are given, for Workstation k, by

$$\gamma_k = \sum_{i=1}^{m} \sum_{\ell=1}^{\nu_i} \widetilde{\gamma}^i_\ell \, I(\widetilde{w}^i(\ell), k) \, , \tag{6.10}$$

and

$$C^2_a(0,j) = \left(\sum_{i=1}^{m} \sum_{\ell=1}^{\nu_i} \widetilde{\gamma}^i_\ell \widetilde{C}^2_a(i,0,\ell) \, I(\widetilde{w}^i(\ell), k) \right) / \gamma_k \, . \tag{6.11}$$

The system of equations defined by Property 5.8 or 5.9 can now be used to find the squared coefficients of variation for the arrival streams to each workstation.

Example 6.7. Example 6.6 can now be completed (Fig. 6.1). The associated average product routing matrix for the three workstations obtained from Property 6.6 is

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0.0541 & 0.4865 & 0 \end{bmatrix} .$$

The system of equations for computing the coefficients of variation for the average product arrival streams at each workstation is

$$\begin{aligned} C^2_a(1) &= \frac{0.2}{0.4327}(1) + \frac{0.2105}{0.4327}\left[\left(1-u_2^2\right)C^2_a(2) + u_2^2 C^2_s(2)\right] + \frac{0.4327 \times 0.0514}{0.4327} \\ &\quad \times \left[0.0514\left(1-u_3^2\right)C^2_a(3) + 0.0514 u_3^2 \left(\frac{C^2_s(3)+\sqrt{2}-1}{\sqrt{2}}\right) + 1 - 0.0514\right] \\ &= 0.1913 C^2_a(2) + 0.0015 C^2_a(3) + 0.8811 \end{aligned}$$

$$
\begin{aligned}
C_a^2(2) &= \frac{0.4327 \times 0.4865}{0.2105} \\
&\quad \times \left[0.4865\left(1-u_3^2\right)C_a^2(3) + 0.4865u_3^2\left(\frac{C_s^2(3)+\sqrt{2}-1}{\sqrt{2}}\right) + 1 - 0.4865\right] \\
&= 0.2767C_a^2(3) + 0.7526
\end{aligned}
$$

$$
\begin{aligned}
C_a^2(3) &= \left(1-u_1^2\right)C_a^2(1) + u_1^2\left(\frac{C_s^2(1)+\sqrt{2}-1}{\sqrt{2}}\right) \\
&= 0.4309C_a^2(1) + 0.7789\,.
\end{aligned}
$$

The solution to this linear system of equations

$$\mathbf{c}_a^2 = (1.093, 1.098, 1.250)\,.$$

This results in the workstation performance measures given in Table 6.8.

Table 6.8 Cycle time and *WIP* results for Example 6.7

Workstation #	CT_q	CT	WIP
1	6.167 hr	9.653 hr	4.177
2	15.310 hr	19.010 hr	4.002
3	2.941 hr	5.976 hr	2.586

The average total system *WIP* for the factory is the sum of the three workstation *WIP*'s resulting in 10.765 jobs. Thus, by Little's Law the mean cycle time in the system is 53.8 hours. Notice that the mean cycle time of a job within the factory is more than the simple sum of the three workstation mean cycle times because of the reentrant flows. □

The processing step modeling paradigm is a useful and surprisingly powerful analysis methodology. This approach can be used for all the problems that have been studied in this text, whereas the workstation modeling approach cannot be used for all cases.

6.5.3 Alternate Approaches

The approach taken in this textbook for analyzing problems of multiple product systems with deterministic routings is to treat departing jobs from a workstation as if their type and, therefore, their next workstation are unknown. Without the job type information, jobs appear to branch probabilistically to their next workstation. Thus, based on Property 5.6, the SCV of the inter-arrivals to Workstation j coming from

[2] Section 6.5.3 can be omitted without affecting the continuity of the remainder of the text.

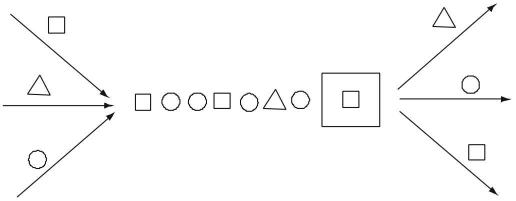


Fig. 6.2 Illustration of a multiple product deterministic routing process with the products being represented by distinct symbols

Workstation k, $C_a^2(k,j)$ is obtained from the departing workstation's composite C_d^2 by

$$C_a^2(k,j) = p_{k,j}C_d^2 + 1 - p_{k,j} \; , \tag{6.12}$$

where $p_{k,j}$ is the job's branching probability derived from Property 6.6. For probabilistic (Markovian) routings, this SCV adjustment is mathematically exact. But for deterministic routings, this approach can be significantly inaccurate, especially when there are only a few deterministic routes with very little re-entrant flows. The purpose of this section is to present an alternate method for determining the squared coefficient of variation for the workstation arrival streams, although for most situations, the models presented above should prove to be sufficiently accurate for most purposes. The following example demonstrates the potential for inaccurate estimates.

Example 6.8. To illustrate the potential inaccuracy of (6.12), consider a workstation that processes three products with each going to a specified and different workstation upon leaving this workstation as illustrated by Fig. 6.2. For purposes of this example, the workstation of the figure will be designated as Workstation 4, and the Products 1, 2, and 3 are sent to Workstations 1, 2, and 3, respectively. The individual arrival stream information (into this workstation) and necessary workstation processing time parameters are listed in the following table.

Table 6.9 Arrival stream and service time characteristics by product type for Fig. 6.2

Product i	λ_i	$C_a^2(i)$	$E[S_i]$	$C_s^2(i)$
1	1	1.50	0.3	1.5
2	1	2.50	0.3	1.5
3	1	0.75	0.3	1.5

The workstation utilization factor is 3(0.3) = 0.9; there is only one machine available. The C_d^2 for the composite departure stream, using the typical *i.i.d.* approximation of Property 5.2 is

$$\begin{aligned} C_d^2(4) &= (1-u_4^2)C_a(4)^2 + u_4^2 C_s^2(4) \\ &= \left(1-0.9^2\right)\frac{(1.50+2.50+0.75)}{3} + 0.9^2(1.5) = 1.516 \; . \end{aligned} \tag{6.13}$$

The mean arrival rates for the three products are identical, so the probability of an output unit being of a specific type is 1/3. Hence, using the probabilistic routing approach, the squared coefficients of variation for each individual product arrival stream at the next workstations are estimated to be equal, with value

$$C_a^2(4,i) = \frac{1}{3}(1.516) + \frac{2}{3} = 1.172 \text{ for } i = 1,2,3 .$$

Simulating this situation with over 270,000 observations yields the following estimates:

$$\begin{aligned} C_a^2(4,1) &= 1.466 \\ C_a^2(4,2) &= 2.056 \\ C_a^2(4,3) &= 1.057 . \end{aligned}$$

These results deviate quite drastically from the probabilistic routing estimates, with approximate errors of 25%, 75%, and -10%, for the three products, respectively. □

This deterministic routing phenomenon was first studied in [2] and recently generalized in [3]. The approach taken is based on approximating the output process from a workstation as an *i.i.d.* process but recognizing that different products may have, on average, different numbers of other products between departures of the same product. This recognition lead to the development of a product's inter-arrival time SCV at the next workstation by assuming various distributions for the number of other product units intervening between departures of the same product. Bitran and Tirupati in [2] use a limit result that the superposition of a large number of independent renewal processes can be approximated by a Poisson process and, assuming that the number of intervening product units is Poisson distributed, develop the estimator

$$C_d^2(k,i) = p_{k,i}C_d^2(k) + (1-p_{k,i})^2 C_a^2(k,i) + p_{k,i}(1-p_{k,i}) , \tag{6.14}$$

where each job type has its own stream; thus, $C_d^2(k,i)$ is the SCV for inter-departures of Type i that leave Workstation k designated to enter another workstation, and $C_a^2(k,i)$ is the SCV for inter-arrivals to Workstation k of Type i coming from another workstation.

Caldentey in [3] develops a general approach to the estimation problem, but points out the computational difficulties encountered in the absence of an intervening number of units assumption. He develops an asymptotic approximation (first proposed in [6]), assuming the individual product's intensity is small in comparison to the aggregate stream, which is

$$C_d^2(k,i) = p_{k,i}C_d^2(k) + (1-p_{k,i})^2 C_a^2(k,i) + p_{k,i} \sum_{j' \neq i} p_{k,j} C_a^2(j',k) . \tag{6.15}$$

Applying these two estimators to the example problem yields the results:

Table 6.10 Comparisons of three methods for estimating splitting with simulation results

Method	$C_d^2(4,1)$	$C_d^2(4,2)$	$C_d^2(4,3)$
Simulation	1.466	2.056	1.057
Markovian Routing	1.172	1.172	1.172
Poisson (Eq. 6.14)	1.394	1.839	1.061
Asymptotic (Eq. 6.15)	1.533	1.866	1.283

For this situation, the Poisson approximation for the number of intervening units yields the best overall approximation. However, other assumptions such as the assumption in [2] of a small number of intervening units approximations based on an Erlang assumption might yield better approximations. The Erlang approach unfortunately does not result in an analytical expression and numerical evaluations must be made.

- *Suggestion: Do Problems 6.3–6.4, 6.12 and 6.14.*

6.6 Group Technology and Cellular Manufacturing

Batch manufacturing is the most common form of production used in the United States [5, p. 420] making up approximately 50% of the production activity. One method that attempts to make batch manufacturing more efficient is group technology. The basic idea of group technology is to essentially establish sub-factories within a factory with each sub-factory being dedicated to the production of a subset of the total number of part types produced by the factory, where the part types have been grouped by common characteristics. (Of course, this concept can be applied to the production of subsequences as well as the full production process for a part type.) Thus, the machines of the factory are grouped into cells of machines needed to produce the job type family assigned to that sub-factory. Part families are chosen so that the parts have as similar processing operations as possible. The forming of these part families is called group technology.

Definition 6.5. *Group technology* is the analysis of processing operations with the goal of determining the similarity of the processing functions and, hence, the grouping of the associated parts for production purposes.

The normal factory organization is to group similar machines together and produce all part types by routing each part through this one grouping of machines for a given processing operation. However, group technology takes advantage of grouping machines according to the similarities of the parts being manufactured which is called cellular manufacturing.

Definition 6.6. The concept of organizing the factory into sub-factories with the capability to produce a technology group is called *cellular manufacturing*.

The advantages sought in grouping the parts into technology groups for separate processing are:

1. More efficient processing by specializing in a smaller set of parts with as similar as possible processing operations. Thus, improvements could come from reduced setup times between part types due to their production similarities and from the learning-curve effects of part specialization. Reduced setups lead to smaller batch sizes and processing procedures that more closely resemble a flow shop.
2. Reduced WIP in each machining area since parts only encounter other parts from the same technology group as well as due to a reduction in the service time squared coefficient of variation (C_s^2). The major factors leading to a reduction in WIP, however, are the impacts of reduced setups and smaller batch sizes.
3. Reduced material handling requirements since distances the jobs must travel between machines within a cell are usually much smaller than the length of the routes needed within a traditional setting. Some material handling processes can be approximated by the techniques discussed in this text, but some processes cannot. For example, if movement of parts is by a forklift, a "forklift" workstation could be defined and the batching techniques discussed in the following chapter could be used. However, modeling a conveyor system that is subject is beyond the scope of this text.

The analysis methods for grouping parts with similar production processes and for the sequencing machines within the group production cells (sub-factories) to best accommodate group part-flow sequences are not discussed in this presentation. Suggested readings for discussions of these methodologies are textbooks by Groover [5, Chap. 15] and by Askin and Standridge [1, Chap. 6]. In particular, [5] discusses several additional aspects of cellular manufacturing such as the physical consideration of cell layouts to facilitate various material handling methodologies. In keeping with our simplification of the factory analysis methodologies, material handling and facility layout issues are not addressed here.

The issue of factory performance when the cellular processing organization is used can be studied with the tools that have already been developed. Conceptually, the standard (batch) production organization is to have one large production facility with similar machines/operations located together in workstations. This is the modeling paradigm that we have followed up to this point. The cellular approach can be modeled by thinking of the manufacturing cells as smaller production facilities each organized to process only one technology group.

A down side of the cellular manufacturing approach is that the economy of scale is lost with respect to the total number of machines needed to produce all technology groups. Another disadvantage of the sub-grouping of machines is that when a machine goes down there is a greater disruptive effect because there are fewer machines available with which to continue processing. Note that a cell with only one machine of a given type will be essentially shutdown while that machine is not operating. Another issue is that the separation of the machines into cells must be whole machines while the workload separation may not coincide properly. This can lead to situations where a good balance between the workload and the number of machines (utilization factor) in the combined organization separates into imbalances in the cellular organization. Thus, some groups might have too high a utilization factor and others too low. To illustrate this point, consider a factory separated into four technology groups with 1/4 of the workload for a given machining operation placed into each group. Further assume that each technology group has a total workload requirement of 1.5 hours (each hour) for this particular machine type. Then the single factory organization needs enough machines to handle a workload of 6 hours per hour. Since the number of machines must be strictly less than the workload, the factory could use 7 machines resulting in an 85.7% average utilization. Using the cellular processing organization, however, two machines of this type would be needed in each manufacturing cell to handle the workload of 1.5 hours per hour. There is no feasible method for partitioning these machines to properly cover the group loads. This means that a total of 8 machines are needed in the factory as a whole using a cellular processing organization. Of course, this extra machining power would reduce the cell utilization for this machine type to 75% yielding a possible cycle time reduction at the expense of an extra machine.

Example 6.9. To illustrate the modeling approach used for group technology and cellular manufacturing, we contrast the part group performance measures for this approach with that for the standard batch processing approach. We will give the basic data in this example and then follow this example with two examples giving the analysis for a traditional factory and then for the cellular factory. It should be noted that the attendant advantages of a cellular processing organization (such as reduced setup times, reduced variability in processing times, and reduced material handling times) are not automatically reflected in a model of this production process. The reduction in setup times and material handling times must wait until the modeling approaches of the next chapter have been introduced. For now, it is necessary for the modeler to estimate these impacts and adjust the model data accordingly.

Consider a manufacturing facility with 4 products and 5 machine types. To determine whether or not a cellular structure is worthwhile, we first look at a table showing which workstations are needed by the different job types. Table 6.11 contains the processing requirements by job type and machine type (workstation) with a 1 representing that the given machine group is used and a zero indicating it is not used. From this table, it is easy to see that a two-group partitioning of the products is possible. The resulting cellular organization of the factory will have two cells with

Table 6.11 Machine usage by job type for Example 6.9, where a 1 indicates the job requires processing at the workstation and a 0 indicates the job does not require the workstation

	Workstation #				
Job Type	1	2	3	4	5
1	1	1	1	0	0
2	1	1	1	0	0
3	0	0	1	1	1
4	0	0	1	1	1

both cells needing Workstation 3. So these three machines will need to be partitioned (if possible) into the cells according to the work demand in each group.

Each job type requires 4 processing steps as shown in Table 6.12. This table contains the mean arrival rate for each job type, the sequence in which the workstations must be visited, and the mean processing time at each step.

Table 6.12 Arrival rates, processing sequence, and mean service times by job type and processing step for Example 6.9

		Workstation Sequence by Step #				Mean Service Time by Step #			
Job Type	Arrival Rate	1	2	3	4	1	2	3	4
1	0.064/hr	3	1	2	1	8 hr	6 hr	4.5 hr	6 hr
2	0.096/hr	1	2	3	1	5 hr	6 hr	8 hr	4 hr
3	0.080/hr	4	3	5	4	2 hr	4 hr	8 hr	4 hr
4	0.100/hr	3	4	5	3	7 hr	3 hr	2 hr	4 hr

The arrival processes are each assumed to be exponentially distributed ($C_a^2 = 1$) and the processing times are assumed to follow an Erlang-2 distribution ($C_s^2 = 1/2$). The number of machines at each workstation are 2, 1, 3, 1, and 1 for Workstations 1 through 5, respectively. □

Example 6.10. **Traditional Factory Model.** In this example, we summarize the analysis of Sect. 6.5 for the data of Example 6.9. The standard (batch) processing organization model of this system has 5 workstations for processing the 4 part types.

The workload for each workstation (machine group) is computed by considering all products that visit the workstation and the number of times they visit. For example, Workstation 1 is visited twice by Job Type 1 (6 hours processing on visit 1 and 6 hours processing on visit 2) twice by Job Type 2 (5 hours processing on visit 1 and 4 hours processing on visit 2). The release rate is 0.064 jobs/hour for Type 1 and 0.096 jobs/hour for Type 2. Hence the total amount of work that is released for Workstation 1 is

$$\text{workload}_1 = (6+6)0.064 + (5+4)0.096 = 1.632\,.$$

Thus, at least two machines are needed in Workstation 1. The utilization factor for Workstation 1, u_1, is the workload divided by the number of machines available (assuming 100% availability)

$$u_1 = 1.632/2 = 0.816\,.$$

A similar analysis for the other four workstations yields the results of Table 6.13.

The expected processing time for Workstation 1 is a function of the three distinct processing times (Job Type 1 uses the machine twice but has the same processing time for each visit) and the relative frequencies of these visits. That is, the machine processing time distribution characteristics are developed using the mixture

Table 6.13 Workload and utilization factors for Example 6.10

Workstation #	Num Machines	Workload	Utilization
1	2	1.632	0.816
2	1	0.864	0.864
3	3	2.700	0.900
4	1	0.780	0.780
5	1	0.840	0.840

of distributions methodology (as in 1.6.3). That is, each visit to the machine by a job can possibly have a different processing time distribution. Thus, we need to use Eqs. (6.3) and (6.4) for the mean and SCV computations, respectively. For Workstation 1, the total arrival rate of jobs is 0.32 per hour (two inflows of Job Type 1 at a rate of 0.064 per hour and two inflows of Job Type 2 at a rate of 0.096 per hour). Thus, the mean processing time is computed as

$$E[S_1] = \left(\frac{0.064}{0.32}\right)6 + \left(\frac{0.064}{0.32}\right)6 + \left(\frac{0.096}{0.32}\right)5 + \left(\frac{0.096}{0.32}\right)4 = 5.100 \text{ hr} .$$

Recall that all processing times are assumed to be distributed according to an Erlang-2 with specified means. Thus, the SCV is computed as

$$\begin{aligned} E[S_1^2] &= 2\left(\frac{0.064}{0.32}\right)6^2(1+1/2) + \left(\frac{0.096}{0.32}\right)5^2(1+1/2) \\ &\quad + \left(\frac{0.096}{0.32}\right)4^2(1+1/2) = 40.05 \text{ hr}^2 \end{aligned}$$

$$C_s^2(1) = \frac{E[S_1^2] - E[S_1]^2}{E[S_1]^2} = \frac{40.05 - 26.01}{26.01} = 0.540 .$$

Continuing with the other four workstations yields the data of Table 6.14.

Table 6.14 Service time characteristics for Example 6.10

Workstation k	λ_k	$E[S_k]$	$C_s^2(k)$
1	0.32/hr	5.100 hr	0.540
2	0.16/hr	5.400 hr	0.528
3	0.44/hr	6.136 hr	0.631
4	0.26/hr	3.000 hr	0.603
5	0.18/hr	4.667 hr	1.112

The final step before obtained the system of equations defining the squared coefficients of variation is the calculation of the probabilities of a job leaving one workstation being sent to another workstation; namely, implementing Property 6.6 which yields

$$P = \begin{bmatrix} 0 & \frac{0.064+0.096}{0.32} & 0 & 0 & 0 \\ \frac{0.064}{0.16} & 0 & \frac{0.096}{0.16} & 0 & 0 \\ \frac{0.064+0.096}{0.44} & 0 & 0 & \frac{0.1}{0.44} & \frac{0.08}{0.44} \\ 0 & 0 & \frac{0.08}{0.26} & 0 & \frac{0.1}{0.26} \\ 0 & 0 & \frac{0.1}{0.18} & \frac{0.08}{0.18} & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0.5 & 0 & 0 & 0 \\ 0.4 & 0 & 0.6 & 0 & 0 \\ 0.3636 & 0 & 0 & 0.2273 & 0.1818 \\ 0 & 0 & 0.3077 & 0 & 0.3846 \\ 0 & 0 & 0.5556 & 0.4444 & 0 \end{bmatrix} .$$

The routing matrix, P, can now be used with the previously obtained quantities in Tables 6.12–6.14 together with the fact that the external arrival streams have an SCV of 1.0 to derive the squared coefficients of variation of the inter-arrival times to each workstation. As you recall from the previous chapter, a system of equations must be developed and solved simultaneously to obtain these terms. In particular, Property 5.8 yields the following system:

$$\begin{aligned} C_a^2(1) &= 0.0203C_a^2(2) + 0.0346C_a^2(3) + 0.8856 \\ C_a^2(2) &= 0.1671C_a^2(1) + 0.7246 \\ C_a^2(3) &= 0.0332C_a^2(2) + 0.0219C_a^2(4) + 0.0372C_a^2(5) + 0.8581 \\ C_a^2(4) &= 0.0166C_a^2(3) + 0.0403C_a^2(5) + 0.9388 \\ C_a^2(5) &= 0.0154C_a^2(3) + 0.0837C_a^2(4) + 0.8354 \,. \end{aligned}$$

The solution to this system is

$$\mathbf{c}_a^2 = (0.9361, 0.8810, 0.9437, 0.9921, 0.9329) \,.$$

The performance measures for Workstation 1 are computed using

$$\begin{aligned} CT(1) &= \left(\frac{C_a^2(1) + C_s^2(1)}{2}\right)\left(\frac{u_1^{\sqrt{6}-1}}{2(1-u_1)}\right) E[T_s(1)] + E[T_s(1)] \\ &= \left(\frac{0.936 + 0.540}{2}\right)\left(\frac{0.816^{1.449}}{0.368}\right) 5.100 + 5.100 = 12.72 \text{ hr} \,. \end{aligned}$$

Notice that we used the approximation of Property 3.6 in the above equation together with the fact that Workstation 1 had two servers. Using Little's Law yields $WIP(1) = 0.32(12.714) = 4.068$ jobs. A similar analysis for the other workstations yields the results given in Table 6.15. Adding all of the workstation WIP's together gives a total system WIP_s of 25.67 jobs. The total external arrival rate and, thus,

Table 6.15 Cycle times and WIP for Example 6.10

Workstation k	λ_k	$CT(k)$	$WIP(k)$
1	0.32/hr	12.714 hr	4.068
2	0.16/hr	29.557 hr	4.729
3	0.44/hr	19.408 hr	8.539
4	0.26/hr	11.482 hr	2.985
5	0.18/hr	29.718 hr	5.349

throughput is 0.34 jobs per hour; therefore, the average cycle time for a job through this factory is $25.67/0.34 = 75.50$ hours. □

Example 6.11. **Cellular Factory Model.** Using a cellular factory organization, the products are separated into two groups with Job Types 1 and 2 in Group 1 and produced by Cell 1, and Job Types 3 and 4 in Group 2 produced in Cell 2. Assuming no improvements in processing times (no setup reductions, etc.), both groups have Machine 3 requirements with workloads by group of 1.280 and 1.420, respectively, for Groups 1 and 2. Notice that the sum of these two workloads equals the workload of 2.7 that was used in the previous example for Workstation 3. Since both of these cells require at least two machines of Type 3, an additional machine must be purchased to implement the disjoint cellular manufacturing approach. Treating these cells as separate sub-factories, the system performance measures can be computed using the same approach as Example 6.10 except that each cell is treated as a separate three-workstation factory. These results are given in Table 6.16.

Table 6.16 Cell performance measures for Example 6.11 with no adjustment in service requirements

	th	WIP	CT
Cell 1	0.16/hr	10.543	65.896 hr
Cell 2	0.18/hr	10.943	60.792 hr

The group technology/cellular manufacturing organization of this total factory, using two technology groups, appears to yield lower cycle times for each technology group in comparison to the standard combined approach; however, the comparison is not fair since an extra machine had to be purchased to establish the cellular organization. To appropriately compare the two factory organizational schemes, the performance measures of the traditional factory layout are recalculated using an additional machine for Workstation 3. The recalculation yields a total system WIP of 20.578 for the traditional factory as compared to the total system WIP of 21.486 for the cellular factory.

One of the keys for cellular manufacturing to be worthwhile is the reduction in processing times due to the similarities of jobs being processed on a machine. For this example, the savings should appear for the processing times on those machines in Workstation 3. For planning purposes, we assume a 25% decrease in the processing time for Machine 3 for both technology groups. After an analysis with the new processing times, the resulting performance measures for the cellular factory are given in Table 6.17. Thus, if the cellular organization permits the 25% reduction in Machine 3 mean processing time, the mean cycle time for Group 1 jobs experiences a 1.7% increase and the mean cycle time for Group 2 jobs experiences a 10.2% decrease with respect to the traditional factor layout using four machines for Workstation 3. It should be noted that we only considered that the cellular organization allowed for the improvement of the processing times for Machine Type 3. Since the other machines, for this example, were not used in other technology groups. Hence, the rational is that the processing time gains due to specialization should have already occurred. □

Table 6.17 Cell performance measures for Example 6.11 with a 25% reduction in mean processing time for Machine 3

	th	WIP	CT
Cell 1	0.16/hr	9.848	61.548 hr
Cell 2	0.18/hr	9.785	54.359 hr

This example illustrates that a group technology/cellular manufacturing organization of the factory can yield a cycle time reduction when implemented in a logical fashion only if there are resulting reductions in the setup and/or processing times. The partitioning of the factory into several non-overlapping production cells is not the actual phenomena from which the improvements in the performance measures are gained. The gains are mainly due to the improvements in production that can be associated with specialization: setup reductions, learning curve effects (reduced processing times), processing simplifications, and improved quality due to specialization. In addition, the material handling/part transportation aspects of the factory may also be more specialized and certainly less travel distance will be realized in a cellular organization.

- *Suggestion: Do Problem 6.13.*

Problems

6.1. Consider a facility that produces two products in three workstations. Product 1 follows the probabilistic workstation transition matrix given by

From/To	1	2	3
1	0.0	0.3	0.5
2	0.2	0.0	0.8
3	0.4	0.5	0.0

while Product 2 has the transition matrix

From/To	1	2	3
1	0.0	0.6	0.4
2	0.3	0.0	0.7
3	0.4	0.1	0.0

The workstation processing time distributions are different by product. For Product 1, these data are

Workstation #	$E[T_s]$	C_s^2
1	1.1 hr	1.0
2	1.0 hr	1.5
3	0.6 hr	2.0

For Product 2, these data are

Workstation #	$E[T_s]$	C_s^2
1	0.25 hr	1.0
2	0.35 hr	1.5
3	0.60 hr	2.0

The mean release rate for Product 1 is 0.2 jobs per hour and for Product 2 is 0.3 jobs per hour, both releases according to a Poisson process into Workstation 1.
(a) Determine the minimum number of (identical) machines that must be placed in each workstation so that a steady-state system results.
(b) Using the number of machines determined in Part (a), find the workstation and system performance measures: cycle time, work-in-process, and throughput.

6.2. Resolve Problem 1 with machine availabilities less than one. Two sets of machine availabilities and repair time data for the three workstations are given below. Compare these answers with those of Problem 1.
(a)

Workstation #	1	2	3
Availability	0.9	0.9	0.9
$E[R]$	1 hr	1 hr	1 hr
$C^2[R]$	1.5	1.75	2.0

(b)

Workstation #	1	2	3
Availability	0.95	0.90	0.87
$E[R]$	42 min	60 min	72 min
$C^2[R]$	1.5	1.75	2.0

6.3. Consider a factory with the process flow as given in the following table.

Step #	1	2	3	4	5	6
WS #	1	2	3	1	2	4
$E[T_s]$	10 min	7.5 min	7.5 min	8.6 min	10 min	10.9 min
$V[T_s]$	79.2 min^2	68.4 min^2	82.8 min^2	72 min^2	126 min^2	90 min^2

In addition, an inspection is preformed after the third processing step, and 10% of the jobs must be totally reworked and are returned to the beginning of process. Compute the system and workstation measures of effectiveness of throughput, WIP and cycle time. There are two machines in Workstations 1 and 2 and one machine in Workstations 3 and 4. Consider an arrival rate of jobs of 5 per hour (exponentially distributed time between arrivals) from an external source.

6.4. Resolve Problem 6.3 with machine availabilities and repair time data given by:

Workstation #	1	2	3	4
Availability	0.9	0.85	0.8	0.99
$E[R]$	1 hr	1 hr	1 hr	1 hr
$C^2[R]$	1	1	1	1

6.5. The Southwestern Specialties Company has a line of four products that they produce in their factory located in Houston, Texas. The Houston factory consists of three workstations (called Workstations 1, 2 and 3). The four products take different routes through the three workstations and have different numbers of processing steps. There are currently three machines in Workstation 1 and 3, and one machine in Workstation 2.

Orders are released to the factory in a random fashion with the mean rate of total order releases being 7.68 per day. The random order release implies that the time between orders is exponentially distributed. The release of orders to the factor is random with 20% of the orders being for Product 1, 30% for Product 2, 25% for Product 3, and 25% for Product 4.

The processing step sequence and mean processing times in hours, are given in the following table where the individual processing times follow an Erlang-2 distribution.

	Step #	1	2	3	4	5
Product 1	Workstation #	3	1	2	1	—
	Mean Time	8.0	6.0	1.7	6.0	—
Product 2	Workstation #	1	2	3	2	1
	Mean Time	5.0	1.6	8.0	1.5	5.0
Product 3	Workstation #	2	1	2	3	1
	Mean Time	1.9	4.0	2.2	8.0	4.0
Product 4	Workstation #	3	1	2	—	—
	Mean Time	8.0	3.0	2.2	—	—

What is the average cycle time for all products combined? What is the mean cycle time for each product?

6.6. This problem is designed to encompasses all of the basic components of building a multiple product model. To help reduce the time to solve the problem, many of the numerical values are given so that the entire problem need not be worked out. There are ten parts to the problem with several tables of numerical values being given; however, many of the tables will contain incomplete information, so those places in the table should be filled in.

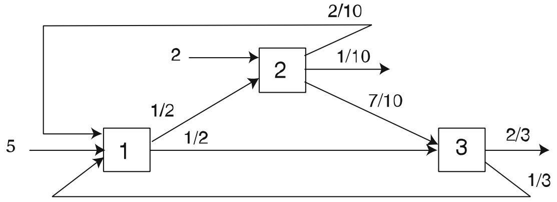


Fig. 6.3a Process routing for Product 1 for Problem 6.6

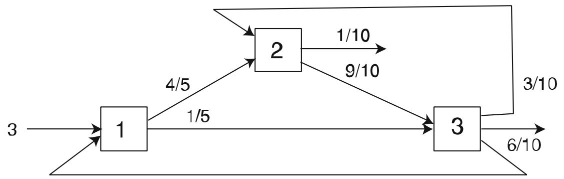


Fig. 6.3b Process routing for Product 2 for Problem 6.6

A company is developing a factory to produce two different products. Both products use three distinct machining processes; thus, the factory will require three workstations. Company management would like to know several things about the factory before it is built. To support this analysis engineering has developed estimates for the necessary product processing information. This information is listed below, and Figs. 6.3a and 6.3b depict the product processing routings. Answer the questions and fill in the missing data.

Mean processing times by product and workstation.

Product/WS	1	2	3
1	0.120 hr	0.100 hr	0.060 hr
2	0.100 hr	0.035 hr	0.060 hr

SCV of the processing times by product and workstation.

Product/WS	1	2	3
1	0.7	0.8	0.9
2	0.8	0.9	1.0

Machine availabilities and repair time characteristics.

Workstation #	1	2	3
Availability	0.90	0.95	0.93
$E[R]$	1.00 hr	1.00 hr	1.00 hr
$C^2[R]$	1.50	1.75	2.00

Answer the following questions and fill in the missing information.

(a) Write the system of equations needed to find the mean flow rates into each workstation for Product 1. These equations would yield the following mean arrival rates for the two products into the three workstations:

Product/WS	1	2	3
1	9.5135/hr	6.7568/hr	9.4865/hr
2	3.4326/hr	4.0439/hr	4.3260/hr

(b) Mean processing times adjusted for breakdowns and repairs (exponential time between breakdowns):

Product/WS	1	2	3
1	?	0.1053 hr	0.0645 hr
2	0.1111 hr	0.0368 hr	0.0645 hr

(c) Processing time SCV's adjusted for breakdowns and repairs (exponential time between breakdowns):

Product/WS	1	2	3
1	?	2.1062	4.1550
2	3.05	4.6321	4.2550

(d) Composite product mean and SCV processing time data by workstation:

Workstation	1	2	3
$E[S]$	0.1274 hr	?	0.0645 hr
$C^2[S]$	2.6919	?	4.1863

(e) The offered loads and, hence, the minimum number of machines required for each workstation:

Workstation	1	2	3
Workloads	?	?	?
Min Machines	?	?	?

(f) Average product branching probability by workstation:

From/To	1	2	3
1	0	0.5795	0.4205
2	0.1251	0	?
3	0.2603	0.0940	0

(g) Write the equation (give explicit numbers whenever possible) for the average product arrival SCV into Workstation 2, $C_a^2(2)$. The external arrival streams are assumed be Poisson processes. Evaluate $C_a^2(2)$ given the other $C_a^2(k)$'s.

Workstation	1	2	3
$C_a^2(k)$	1.2144	?	1.8346

(h) Complete the workstation performance measures.

Workstation	$CT_q(k)$	$CT(k)$	$WIP(k)$
1	0.5379 hr	0.6653 hr	8.6131
2	1.0371 hr	1.1167 hr	12.0608
3	?	?	?

(i) What are the values of the factory system performance measures of cycle time (CT_s), work-in-process (WIP_s) and throughput?

(j) What are the values of the individual product system performance measures of throughput and cycle time (CT^i) for product $i = 1,2,3$?

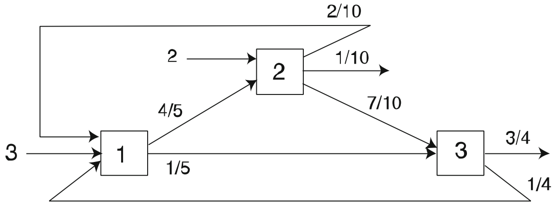


Fig. 6.4a Process routing for Product 1 for Problem 6.7

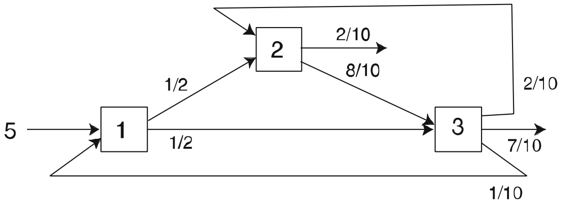


Fig. 6.4b Process routing for Product 2 for Problem 6.7

6.7. This problem is designed to encompasses all of the basic components of building a multiple product model. To help reduce the time to solve the problem, many of the numerical values are given so that the entire problem need not be worked out. There are ten parts to the problem with several tables of numerical values being given; however, many of the tables will contain incomplete information, so those places in the table should be filled in.

A company is developing a factory to produce two different products. Both products use three distinct machining processes; thus, the factory will require three workstations. Company management would like to know several things about the factory before it is built. To support this analysis engineering has developed estimates for the necessary product processing information. This information is listed below, and Figs. 6.4a and 6.4b depict the product processing routings. Answer the questions and fill in the missing data.

Mean processing times by product and workstation.

Product/WS	1	2	3
1	0.120 hr	0.100 hr	0.060 hr
2	0.100 hr	0.035 hr	0.060 hr

SCV of the processing times by product and workstation.

Product/WS	1	2	3
1	1.0	0.8	0.7
2	0.8	0.9	1.0

Machine availabilities and repair time characteristics.

Workstation	1	2	3
Availability	0.90	0.95	0.93
$E[R]$	1.00 hr	1.00 hr	1.00 hr
$C^2[R]$	1.75	2.00	1.50

Answer the following questions and fill in the missing information.
(a) Write the system of equations needed to find the mean flow rates into each workstation for Product 1. These equations would yield the following mean arrival rates for the two products into the three workstations:

Product/WS	1	2	3
1	5.7692/hr	6.6154/hr	5.7846/hr
2	5.6000/hr	4.0000/hr	6.0000/hr

(b) The mean processing times adjusted for breakdowns and repairs (exponential time between breakdowns):

Product/WS	1	2	3
1	0.1333 hr	0.1053 hr	0.0645 hr
2	?	0.0368 hr	0.0645 hr

(c) The processing times SCV's adjusted for breakdowns and repairs (exponential time between breakdowns):

Product/WS	1	2	3
1	3.0625	2.2250	3.4125
2	3.2750	4.9714	?

(d) Composite product mean and SCV processing time data by workstation:

Workstation	1	2	3
$E[S]$	?	0.0795 hr	0.0645 hr
$C^2[S]$	?	3.0086	3.5652

(e) The minimum number of machines required for each workstation:

Workstation	1	2	3
Workload	?	?	?
Min Machines	?	?	?

(f) Average product branching probability by workstation is:

From/To	1	2	3
1	0	0.6522	0.3478
2	?	0	0.7377
3	0.1736	0.1018	0

(g) Write the equation (give explicit numbers whenever possible) for the average product arrival SCV into Workstation 1, $C_a^2(1)$. The external arrivals are assumed to be Poisson processes. Evaluate $C_a^2(1)$ given the other $C_a^2(k)$'s.

Workstation	1	2	3
$C_a^2(k)$	?	1.3802	1.8465

(h) Complete the workstation performance measures:

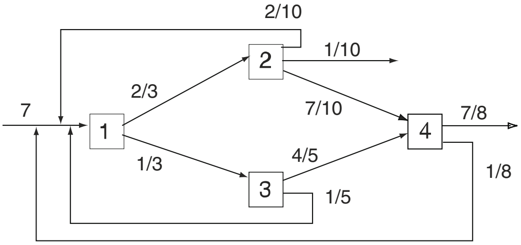


Fig. 6.5a Process routing for Product 1 for Problem 6.8

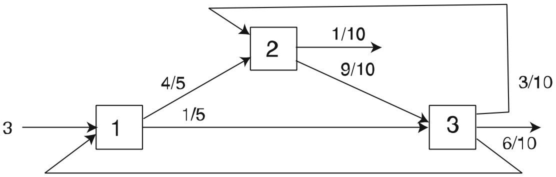


Fig. 6.5b Process routing for Product 2 for Problem 6.8

Workstation	$CT_q(k)$	$CT(k)$	$WIP(k)$
1	0.2533 hr	0.3757 hr	4.2714
2	?	?	?
3	0.5537 hr	0.6182 hr	7.2857

(i) What are the values of the factory system performance measures of cycle time (CT_s), work-in-process (WIP_s) and throughput?

(j) What are the values of the individual product system performance measures of throughput and cycle time (CT^i) for $i = 1,2$?

6.8. This problem is designed to encompasses all of the basic components of building a multiple product model. To help reduce the time to solve the problem, many of the numerical values are given so that the entire problem need not be worked out. There are nine parts to the problem with several tables of numerical values being given; however, many of the tables will contain incomplete information, so those places in the table should be filled in.

A company is developing a factory to produce two different products. The first products use four distinct machining processes; whereas, the second product uses only the first three workstations used by Product 1. Company management would like to know several things about the factory before it is built. To support this analysis engineering has developed estimates for the necessary product processing information. This information is listed below, and Figs. 6.5a and 6.5b depict the product processing routings.

Mean processing times by product and workstation.

Product/WS	1	2	3	4
1	7.2 min	6 min	9 min	7.2 min
2	6 min	2.1 min	3.6 min	

SCV of the processing times by product and workstation.

Product/WS	1	2	3	4
1	0.7	0.8	0.9	1.0
2	0.8	0.9	1.0	

Machine availabilities and repair time characteristics.

Workstation	1	2	3	4
Availability	0.90	0.95	0.93	0.95
$E[R]$	1 hr	1 hr	1 hr	1 hr
$C^2[R]$	1.50	1.75	2.00	5/3

Answer the following questions and fill in the missing information.
(a) Write the system of equations needed to find the mean flow rates into each workstation for Product 1. These equations would yield the following mean arrival rates for the two products into the four workstations:

Product/WS	1	2	3	4
1	9.882/hr	6.588/hr	3.294/hr	7.247/hr
2	3.433/hr	4.044/hr	4.326/hr	

(b) Composite product mean and SCV processing time data by workstation, this data is not adjusted for downtime and repairs:

Workstation	1	2	3	4
$E[S]$	?	4.5 min	5.94 min	7.2 min
$C^2[S]$	?	1.125	1.307	1.000

(c) The average processing times and SCV's adjusted for breakdowns and repairs (exponential time between breakdowns):

Workstation	1	2	3	4
$E[S]$	7.68 min	?	6.36 min	7.56 min
$C^2[S]$	2.689	?	3.282	2.056

(d) Average product branching probability by workstation is:

From/To	1	2	3	4
1	0	0.701	0.299	0
2	0.124	0	?	0.434
3	0.143	0.170	0	0.346
4	0.125	0	0	0

(e) The minimum number of machines required for each workstation:

Workstation	1	2	3	4
Workload	?	?	?	?
Min Machines	?	?	?	?

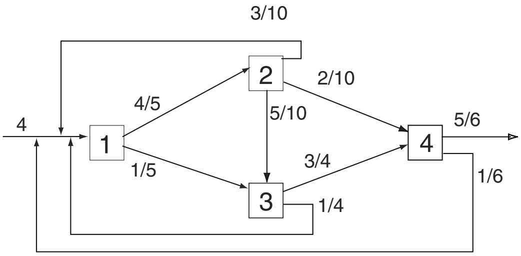


Fig. 6.6a Process routing for Product 1 for Problem 6.9

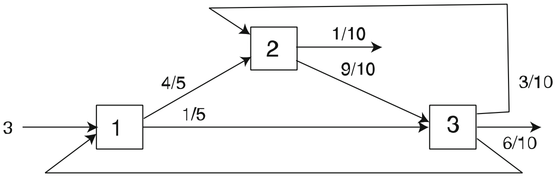


Fig. 6.6b Process routing for Product 2 for Problem 6.9

(f) Write the equation (give explicit numbers whenever possible) for the average product arrival SCV into Workstation 2, $C_a^2(2)$. The external arrivals are assumed to be Poisson processes. Evaluate $C_a^2(2)$ given the other $C_a^2(i)$'s.

Workstation	1	2	3	4
$C_a^2(i)$	1.046	?	1.380	1.615

(g) Complete the workstation performance measures:

Workstation	$CT_q(k)$	$CT(k)$	$WIP(k)$
1	0.625 hr	0.753 hr	10.021
2	0.939 hr	1.018 hr	10.828
3	?	?	?
4	2.509 hr	2.635 hr	19.098

(h) What are the values of the factory system performance measures of cycle time (CT_s), work-in-process (WIP_s) and throughput?

(i) What are the values of the individual product system performance measures of throughput and cycle time (CT^i)?

6.9. This problem is designed to encompasses all of the basic components of building a multiple product model. To help reduce the time to solve the problem, many of the numerical values are given so that the entire problem need not be worked out. There are nine parts to the problem with several tables of numerical values being given; however, many of the tables will contain incomplete information, so those places in the table should be filled in.

A company is developing a factory to produce two different products. The first products use four distinct machining processes; whereas, the second product uses only the first three workstations used by Product 1. Company management would like to know several things about the factory before it is built. To support this analysis engineering has developed estimates for the necessary product processing information. This information is listed below, and Figs. 6.6a and 6.6b depict the product processing routings. Answer the questions and fill in the missing data.

Mean processing times by product and workstation.

Product/WS	1	2	3	4
1	7.2 min	6 min	6 min	9 min
2	6 min	2.1 min	3.6 min	

SCV of the processing times by product and workstation.

Product/WS	1	2	3	4
1	1.0	0.9	0.8	0.7
2	0.8	0.9	1.0	

Machine availabilities and repair time characteristics.

Workstation	1	2	3	4
availability	0.90	0.95	0.93	0.95
$E[R]$	1 hr	1 hr	1 hr	1 hr
$C^2[R]$	1.75	2.00	1.50	5/3

Answer the following questions and fill in the missing information.

(a) Write the system of equations needed to find the mean flow rates into each workstation for Product 1. These equations would yield the following mean arrival rates for the two products into the four workstations:

Product/WS	1	2	3	4
1	7.869/hr	6.295/hr	4.721/hr	4.800/hr
2	3.433/hr	4.044/hr	4.326/hr	
Total	11.301/hr	10.339/hr	9.047/hr	4.800/hr

(b) Composite product mean and SCV processing time data by workstation, this data is not adjusted for downtime and repairs:

Workstation	1	2	3	4
$E[S]$	6.84 min	?	4.86 min	9 min
$C^2[S]$	0.966	?	0.963	0.700

(c) The average processing times and SCV's adjusted for breakdowns and repairs (exponential time between breakdowns):

Workstation	1	2	3	4
$E[S]$	?	4.74 min	5.22 min	9.48 min
$C^2[S]$	?	3.155	2.975	1.544

(d) Average product branching probability by workstation is:

From/To	1	2	3	4
1	0	0.800	0.200	0
2	0.183	0	0.656	0.122
3	?	0.143	0	0.391
4	0.167	0	0	0

(e) The minimum number of machines required for each workstation:

Workstation	1	2	3	4
Workload	?	?	?	?
Min Machines	?	?	?	?

(f) Write the equation (give explicit numbers whenever possible) for the average product arrival SCV into Workstation 1, $C_a^2(1)$. The external arrivals are assumed to be Poisson processes. Evaluate $C_a^2(1)$ given the other $C_a^2(k)$'s.

Workstation	1	2	3	4
$C_a^2(i)$	?	1.602	1.841	1.497

(g) Complete the workstation performance measures:

Workstation	$CT_q(i)$	$CT(i)$	$WIP(i)$
1	17.40 min	24.96 min	4.703
2	48.24 min	52.98 min	9.127
3	?	?	?
4	45.12 min	54.60 min	4.366

(h) What are the values of the factory system performance measures of cycle time (CT_s), work-in-process (WIP_s) and throughput?
(i) What are the values of the individual product system performance measures of throughput and cycle time (CT^i)?

6.10. Using a spreadsheet program such as Excel, solve Problem 6.1.

6.11. Using a spreadsheet program such as Excel, solve Problem 6.2.

6.12. A factory consists of five workstations and produces two products. Develop the product and factory performance measures of throughput, cycle time and work-in-process. Job Type 1 arrives according to a Poisson process with a mean rate of 5 per hour and Job Type 2 arrives with a mean rate of 3 per hour and a squared coefficient of variation of the inter-arrival times of 2. There are two machines at Workstation 2; all other workstations have only one machine. The process flow, mean processing times in hours, and squared coefficient of variation of the processing times are as follows:

	Workstation by Step #			Mean Service Time by Step #			SCV Service Time by Step #		
Job Type	1	2	3	1	2	3	1	2	3
1	1	2	3	0.16	0.17	0.18	2.0	1.3	1.00
2	4	2	5	0.30	0.30	0.28	1.0	1.5	0.75

6.13. For the factory of Problem 6.12, the factory can be segmented into two cellular lines - one product manufactured in each line. Assuming that the mean processing times for Machine 2 can be reduced by 15% when the products are not processed on the same machine. That is, the workstation can specialize its setup operation by product type. Compute the product and total factory performance measures and compare these with the composite factory organization computed in Problem 6.12.

6.14. Consider a workstation that processes three products with each going to a specified and different workstation upon leaving this workstation. Figure 6.2 illustrates this situation. The individual arrival stream information and necessary processing time parameters are:

Product	λ_i	$C_a^2(i)$	$E[S_i]$	$C_s^2(i)$
1	1/hr	1.50	9 min	1.5
2	2/hr	2.50	9 min	1.5
3	3/hr	0.75	9 min	1.5

Compute the arrival SCV, C_a^2, at the next workstation for each product using the three different estimators: probabilistic routing (Eq. 6.9), Poisson intervening units assumption (Eq. 6.14), and the asymptotic method (Eq. 6.15).

References

1. Askin, R.G. and Standridge, C.R. (1993). *Modeling and Analysis of Manufacturing Systems*. John Wiley & Sons, New York.
2. Bitran, G.R., and Tirupati, D. (1988). Multiproduct Queueing Networks with Deterministic Routings: Decomposition Approach and the Notion of Interference. *Management Science*, **34**:75–100.
3. Caldentey, R. (2001). Approximations for Multi-Class Departure Processes. *Queueing Systems*, **38**:205–212.
4. Deuermeyer, B.L., Curry, G.L., and Feldman, R.M. (1993). An Automatic Modeling Approach to the Strategic Analysis of Semiconductor Fabrication Facilities. *Production and Operations Management*, **2**:195–220.
5. Groover, M.P. (2001). *Automation, Production Systems, and Computer-Integrated Manufacturing (Second Ed.)* Prentice-Hall, Upper Saddle River, NJ.
6. Whitt, W. (1994). Towards Better Multi-class Parametric-Decomposition Approximations for Open Queueing Networks. *Ann. Oper. Res.*, **48**:221–248.

Chapter 7
Models of Various Forms of Batching

Grouping individual jobs into sets, called batches, is a strategy frequently used in industry. One cause of batching is for the purpose of transportation between workstations. For instance, workers may require mechanical help for moving heavy items between two machines. If the mechanical help is a large machine such as a forklift, then a pallet might be loaded first before a forklift truck is requested. Another form of batching occurs when items are batched by type for the purpose of sharing a machine setup step even though the items are actually processed individually. By batching like items, only one setup need be performed for the whole set. And finally, a frequently encountered batch service process is that of a multiple service capacity resource such as an oven. Due to the slow processing rates of some heat-treatment or plating processes, large capacity machines have been developed that can process several units of an item simultaneously.

The batching phenomenon is motivated by a perceived beneficial effect of grouping. However, the impact on downstream processing stations can be significant. To illustrate, consider the batch move concept where, say k, items are grouped together for the convenience of moving them to a subsequent single unit processing station. Items will arrive at the next workstation k at a time, so the workstation might be idle for a while and then instantaneously have a queue of waiting units. The variability of a batch arrival process when the batch is broken back into individuals (frequently caused by processing items simultaneously) is much greater than the inter-arrival variability of the individual items and, the workstation queueing behavior will be exacerbated. This leads to increased cycle times and larger WIP levels at the downstream workstation. In addition, the batch process itself causes an increased delay because units must wait for the completion of other units before they can be grouped and continue processing.

In this chapter, models are developed for various forms of batching and so that the benefits and costs of the grouping process under consideration can be quantified. For the setup sharing situation, there will be a trade-off between the cycle time increase and the setup time savings due to batching. The chapter is concluded with a discussion of network models that include a batch (oven-type) processing workstation. The term “job” can be confusing because in some contexts a job my refer

G.L. Curry, R.M. Feldman, *Manufacturing Systems Modeling and Analysis*, 2nd ed.,
DOI 10.1007/978-3-642-16618-1_7, © Springer-Verlag Berlin Heidelberg 2011

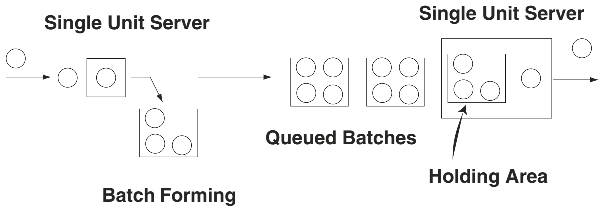


Fig. 7.1 The batch move model structure: batches are formed after single unit processing and are transported to the next workstation; batches wait in the queue until service on individual items within the batch commences; finally, items leave as individuals as soon their processing has been completed

to an individual item and in other contexts it may refer to the entire batch. To avoid confusion, the term "item" will always be used for an individual job and never to the entire batch.

7.1 Batch Moves

Consider a situation where individual items are grouped together into fixed batches of size k at the completion of processing at a workstation that processes single units. Items wait in the incomplete batch until the proper quantity has accrued and then the full batch is transported to the next workstation. (A basic assumption used throughout this text is that transportation time is negligible and, therefore, is not explicitly considered. If transportation time is significant, it can often be approximated in the model by considering the transporter as a separate workstation.) An additional assumption is made that the receiving workstation processes items individually, hence, the batch is merely a convenient transportation tool. The modeling of the batch move situation is a building block for more complex models. In addition, we will demonstrate that batch moves add to the cycle time in comparison to a system where items are moved individually. Figure 7.1 illustrates a batch move system.

To model the batch move, several aspects of the problem will have to be considered. First, the batch forming time as it contributes to each individual item, or the average item delay within a partial batch, needs to be computed. (The batch forming time is added to the cycle time of the workstation receiving the batch even though the batch forming actually takes place at the workstation sending the batch.) Then the arrival stream characteristics for the batch receiving workstation need be developed; that is, the mean arrival rate for batches and the squared coefficient of variation of the inter-arrival batch times must be computed. And finally, the modeling approach for developing the second workstation cycle time is different than our previous analyses. The cycle time model is separated into the standard components of the queue time and the service time. The queue time, however, is developed from the batch point of view. The individual item's service time is the average time for processing individuals. These individuals are assumed to be released immediately after processing to move on to their next workstation. But since there are k items in the batch, the items have different delays while awaiting their turn at service. The first item served from a batch has no additional delay due to waiting for others from the same batch, while the second item serviced from the batch waits for the first item, the third item waits for the first two selected items, and so on. The average delay is then taken over the composite delay for all items in the batch. These cycle time component analyses are addressed one at a time below.

7.1.1 Batch Forming Time

A batch is formed by grouping k individually arriving items together. Let T_d be the random variable denoting the time between departures from the source workstation that are to be batched for transportation to the destination workstation. Note that the arrival rate of individuals $\lambda(I)$, in the absence of batching, to the destination workstation is given by

$$\lambda(I) = \frac{1}{E[T_d]} .$$

The random variable $T(B)$ is the inter-arrival time of batches. This random variable is the sum of k individual inter-departure times

$$T(B) = T_{d,1} + T_{d,2} + \cdots + T_{d,k} .$$

The individual inter-departure times $T_{d,i}$ for $i = 1, \cdots, k$ are independent and identically distributed ($i.i.d.$) random variables; thus, the expected value of the batch inter-arrival time is

$$E[T(B)] = kE[T_d] .$$

Hence, the arrival rate of batches to the destination workstation, $\lambda(B)$, is

$$\lambda(B) = \frac{1}{E[T(B)]} = \frac{1}{kE[T_d]} = \frac{\lambda(I)}{k}. \tag{7.1}$$

The squared coefficient of variation, $C^2[T(B)]$, of the batch inter-arrival times at the destination workstation is obtained from

$$C^2[T(B)] = \frac{V[T(B)]}{E[T(B)]^2},$$

where the variance is computed from

$$V[T(B)] = V[T_{d,1} + T_{d,2} + \cdots + T_{d,k}] = kV[T_d] ,$$

since the inter-departure random variables are assumed independent (see Property 1.6 or Eq. 1.27). Thus, the squared coefficient of variation of the batch inter-arrival times can be computed from the squared coefficient of variation of the individual inter-arrival times by

$$C^2[T(B)] = \frac{V[T(B)]}{E[T(B)]^2} = \frac{kV[T_d]}{(kE[T_d])^2} = \frac{C^2[T_d]}{k} \,. \tag{7.2}$$

The delay that an individual item encounters when being placed into a batch depends on where it is among the k batched items. The first departing item used to start the formation of a new batch must wait for $k-1$ more items to depart before the batch has been formed and released for transportation to the destination workstation. Denote this delay by the random variable D_1 where

$$D_1 = T_{d,2} + \cdots + T_{d,k} \,.$$

The second item forming the new batch has to wait for $k-2$ succeeding departures and its waiting time is the random variable D_2 given by

$$D_2 = T_{d,3} + \cdots + T_{d,k} \,.$$

The other items in the batch have delay times similarly developed with the last item encountering no delay (i.e., $D_k = 0$). The last item's arrival signals the batch is complete and the batch is instantaneously transported to the destination workstation. The average delay encountered by an item in the batch is then the expected value of the sum of all these delays divided by the batch size k; that is,

$$\begin{aligned} E[D] &= \frac{E[D_1 + D_2 + \cdots + D_{k-1} + D_k]}{k} \\ &= \frac{(k-1)E[T_d] + (k-2)E[T_d] + \cdots + 1E[T_d] + 0E[T_d]}{k} \\ &= \frac{((k-1)k/2)\,E[T_d]}{k} \\ &= \frac{(k-1)}{2} E[T_d] \,. \end{aligned} \tag{7.3}$$

Thus, the average delay encountered by an individual item when waiting for a batch of size k to form is $(k-1)E[T_d]/2$.

One should recognize that the term $E[T_d]$ in the expected batch waiting time per individual is the time between arrivals for the particular situation that was used to motivate this analysis. In this situation, batching after job completion at a workstation is done for the purpose of transporting the items to the next workstation. The batching operation could occur at the front of a batch service workstation such as an oven. In this case, the average delay would follow the same form as the result just derived, but the individual item's inter-arrival time would then be denoted as

$E[T_a]$. This is the expected time between arrivals to the workstation and, therefore, the inter-arrival time for batch items.

The batching form that has been analyzed is for a single product or for indiscriminate grouping of multiple products. It is very likely that batching is restricted to jobs of the same type. In this situation, then multiple batch types can be forming simultaneously and the wait associated with batch formation of a given item type would be a function of the inter-arrival time to the workstation of that job type.

7.1.2 Batch Queue Cycle Time

Modeling the cycle time for the recipient workstation for the batch move situation has two distinct components: the queue time and the service time. The queueing delay is modeled from the batch units point of view. The items within a batch see this queueing phenomenon as batches waiting and then moving up in the line based on batches being served. The arrival rate to this queue is $\lambda(B)$ with corresponding squared coefficient of variation $C^2[T(B)]$. It may be clearer to denote the random variable $T(B)$ as $T_a(B)$ to distinguish and relate this random variable with the individual inter-arrival time random variable $T_a(I)$.

The service time that these batches observe while they wait in the queue is for batches (they move forward one location in the queue whenever a batch has been completely served). The service time for these batches is the time it takes the server to process all of the items within the batch being served, which is the random variable $T_s(B)$ given by

$$T_s(B) = T_{s,1}(I) + T_{s,2}(I) + \cdots + T_{s,k}(I) ,$$

where the (B) and (I) notation again stands for batches and individuals, respectively. Note that since this is a single item service facility (items are processed one at a time), the processing times $T_{s,i}(I)$ are independent and identically distributed random variables with known mean $E[T_s(I)]$ and known squared coefficient of variation $C^2[T_s(I)]$. Thus, the (perceived) batch service time characteristics can be computed from these known individual item data as

$$E[T_s(B)] = kE[T_s(I)] ,$$

$$C^2[T_s(B)] = \frac{C^2[T_s(I)]}{k} .$$

The $G/G/1$ cycle time approximation model is used to compute the cycle time in the queue for the waiting batches. The utilization factor for the workstation must also be computed for batches. This computation is

$$\begin{aligned} u(B) &= \lambda(B)E[T_s(B)] \\ &= \frac{\lambda(I)}{k}\,(kE[T_s(I)]) \\ &= \lambda(I)E[T_s(I)] = u(I)\,. \end{aligned}$$

Thus, the utilization factor for the workstation in the batch service mode as perceived by the waiting move batches is the same as the normal single unit processing utilization factor for the workstation. The queue time estimate is given by

$$CT_q(B) = \left(\frac{C^2[T_a(B)] + C^2[T_s(B)]}{2}\right)\left(\frac{u(B)}{1-u(B)}\right)E[T_s(B)] \tag{7.4}$$

$$\begin{aligned} &= \left(\frac{(C^2[T_a(I)]/k) + (C^2[T_s(I)]/k)}{2}\right)\left(\frac{u(I)}{1-u(I)}\right)kE[T_s(I)] \\ &= \left(\frac{C^2[T_a(I)] + C^2[T_s(I)]}{2}\right)\left(\frac{u(I)}{1-u(I)}\right)E[T_s(I)] = CT_q(I)\,. \end{aligned}$$

So the expected cycle time in the queue for individuals in a move batch is identical to the cycle time in the queue for the workstation operating in a single item mode. Therefore, utilizing move batches does not change the expected queueing delay time. (It should be noted here that this is an approximation. The expected cycle time in the queue for individuals in a move batch is actually smaller than the expected cycle time in the queue for the workstation operating in a single item mode, but the Kingman approximations for the two situations are the same.) It does, however, affect the processing time delay as seen by individuals within a batch as is developed in the next section.

7.1.3 Batch Move Processing Time Delays

The final element in the cycle time computation for batch moves between successive workstations is the processing time delay for batched items. The individual item processing time has not been altered by the batching process. The cycle time in the workstation has been separated into the queue waiting time for the batch and the processing time for the batch. Hence, the previous section developed the time that the batch waits until the server is working on items within the batch. There is another element to the actual waiting time until a particular item is served that consists of the delay encountered while the item is waiting its turn for service with respect to the other items within the batch. This delay is analogous to the batch forming delay analyzed above. That is, the first item selected from the batch for processing sees no delay at the processor due to having been part of a move batch. However, the second and subsequent items must wait their turn for service. Thus, items are delayed for their own processing time plus the processing times of all items that were part of their move batch and were selected for service before the item in question. So the first item serviced has a zero extra wait, the second item serviced from the batch has to wait for the service time of the first item, $T_{s,1}$, the third item has to wait for the first two processing times, $T_{s,1}+T_{s,2}$, and so forth through the batch until all items have been processed. The typical item sees an average extra delay that consists of the expected value for the total extra waiting time divided by the number of items, k, in the batch. Again a series is summed for this delay, D,

$$\begin{aligned} D &= T_{s,1}+(T_{s,1}+T_{s,2})+(T_{s,1}+T_{s,2}+T_{s,3})+\cdots+(T_{s,1}+T_{s,2}+\cdots+T_{s,k-1}) \\ &= (k-1)T_{s,1}+(k-2)T_{s,2}+\cdots+(1)T_{s,k-1}. \end{aligned}$$

Since all these processing times are again *i.i.d.* random variables, this is the same series as previously developed for the batch forming time with services replacing inter-arrivals. The expected value for this delay is

$$E[D] = \{(k-1)+(k-2)+\cdots+(1)\}E[T_s]\,. \tag{7.5}$$

The sum of the first $k-1$ integers equals $k(k-1)/2$, so the average extra delay associated with an item waiting its turn within the batch for processing is

$$\frac{E[D]}{k} = \frac{(k(k-1)/2)\,E[T_s]}{k} = \frac{(k-1)}{2}E[T_s]. \tag{7.6}$$

The average processing time delay for a batched item is this extra delay plus the item's expected processing time $E[T_s]$.

The cycle time associated with using a move batch between two workstations consists of the delays encountered at the second workstation plus the batch forming time. Putting these results together yields the following property.

Property 7.1. *Assume a pure serial system layout with Workstation i sending jobs directly to Workstation j. The mean arrival rate of individuals to i is $E[T_a(i)]$ and the SCV of inter-departures of individuals from the processor of i is $C_d^2(i)$. Transporting jobs from i to j is by batch moves of size k and all jobs are processed one-at-a-time at j. The mean and SCV of (individual) processing times at j are denoted by $E[T_s(j)]$ and $C_s^2(j)$, respectively. The mean system cycle time per job at Workstation j is given by*

$$\begin{aligned} CT(j) = {} & \frac{(k-1)}{2}E[T_a(i)] + \left(\frac{C_d^2(i)+C_s^2(j)}{2}\right)\left(\frac{u}{1-u}\right)E[T_s(j)] \\ & + \frac{(k-1)}{2}E[T_s(j)] + E[T_s(j)]\,, \end{aligned}$$

where the batch formation time after processing at Workstation i is considered part of the cycle time at Workstation j.

Comparing Property 7.1 with the standard Kingman approximation (Eq. 3.19) for cycle times, we see that the batch move process adds two elements to the cycle time. Both of these additional delays are due to batching, the batch forming time and the average delay for service due to waiting for other items within the batch to be processed first.

There are times in which a batch has already been formed (e.g., after a batch processor as in Sect. 7.3) so that the basic arrival rate is in terms of batches to a workstation in which processing is by individual job. In this case there would be batch formation times so the following property would be used.

Property 7.2. *Consider a workstation with a mean and SCV of times between batch arrivals being denoted by $E[T_a(B)]$ and $C_a^2(B)$, respectively, and the mean and SCV of individual processing times being denoted by $E[T_s(I)]$ and $C_s^2(I)$, respectively. The mean system cycle time per job at the workstation is thus*

$$CT_s = \left(\frac{kC_a^2(B) + C_s^2(I)}{2}\right)\left(\frac{u}{1-u}\right) E[T_s(I)] + \frac{(k+1)}{2} E[T_s(I)] \,,$$

where the utilization is $u = kE[T_s(I)]/E[T_a(B)]$.

7.1.4 Inter-departure Time SCV with Batch Move Arrivals

The departures from a batch-move single-unit-service workstation have a cyclic behavior. The inter-departure time associated with the first item processed in a batch can consist of two elements, an idle time delay plus a service time delay. All other inter-departure times for items in the batch are merely separated by service time delays. If the workstation is busy processing items when the batch arrives, then the first item processed from the batch will also only experience a service time inter-departure delay. So in modeling the inter-departure times, there is a dependency between the inter-departure times for elements from the same batch. Dependencies between successive inter-departure times are prevalent in most queueing systems. The general approach for modeling departures from $G/G/1$ workstations is to approximate the inter-departure process by a renewal process (see Definition 5.1). (See [1] for a discussion of approximation approaches for departure processes.)

Using a renewal process (*i.i.d.*) approximation to the inter-departure process, Curry and Deuermeyer in [3] developed the inter-departure time squared coefficient of variation for individuals, $C_d^2(I)$, for the batch-move server for a single machine workstation as

$$C_d^2(I) = kC_a^2(B)(1-u^2) + (k-1)(1-u)^2 + u^2 C_s^2(I) \,, \tag{7.7}$$

where k is the fixed batch size of the arriving batches, $C_a^2(B)$ is the SCV of the arriving batch stream to the workstation, $C_s^2(I)$ is the service time SCV for individuals, and u is the workstation utilization factor. In the context of a serial system, the result in [3] can be expressed, using Eq. (7.2), as the following property.

Property 7.3. *Assume a pure serial system layout with Workstation i sending jobs directly to Workstation j by batch moves of size k. Using the same notation as in Property 7.1, the squared coefficient of variation of the inter-departures of individuals from Workstation j is given by*

$$C_d^2(j) = C_d^2(i)(1-u_j^2) + (k-1)(1-u_j)^2 + u_j^2 C_s^2(j) \,.$$

In a simulation study of the departures from a fixed batch arrival system with individual service, a set of 30 simulations was run with batch sizes from 1 to 5 and with $C_a^2(B)$ and $C_s^2(I)$ both ranging over 3/4, 1, and 3/2. Each of these simulations consisted of 100,000 simulated hours. The average absolute error between the theoretical estimate and the simulation estimate for $C_d^2(I)$ for these 30 studies was 1.39% with a maximum error of 4%. Although this study was not over the whole range of values for utilization and inter-arrival and service time SCV's, it does indicate that the *i.i.d.* approximation for the SCV of departures is a viable approach for modeling purposes.

Example 7.1. Arrivals to a sub-factory with two workstations in series occurs at a mean rate of 3 per hour and a squared coefficient of variation of the inter-arrivals of 2. Both workstations consist of only one machine. The processor of the first workstation processes jobs according to a gamma distribution with a shape parameter of 0.5 and a scale parameter of 30 minutes. The processor of the second workstation processes jobs according to a gamma distribution with a shape parameter of 2/3 and a scale parameter of 22.5 minutes. All items must be moved between workstations in batches of size 4. What is the average cycle time in the second workstation and what are the departing stream's characteristics (mean and SCV of the inter-departure times)?

First notice that the mean and SCV of service for the first workstation are 0.25 hour and 2, respectively, and the values for the second workstation are 0.25 hour and 1.5, respectively. Since the SCV of the inter-arrivals and service process of the first workstation are the same, the SCV of the inter-departures will also be the same. Thus, the mean and SCV for the inter-departure of individuals from the server in the first workstation is 1/3 hr and 2, respectively.

The utilization factor for the second workstation is $u_2 = 3(0.25) = 0.75$. Therefore, the average cycle time per item is given from Property 7.1 by

$$CT(2) = \frac{4-1}{2}(1/3) + \frac{2+1.5}{2}\,\frac{0.75}{0.25}(0.25) + \frac{4-1}{2}0.25 + 0.25 = 2.4375 \text{ hr} \,.$$

The mean departure rate of individuals equals the mean arrival rate of individuals which is 3 per hour. The squared coefficient of variation of the inter-departure times is approximated from Property 7.3 by

$$C_d^2(2) = 2(1-0.75^2) + 3(1-0.75)^2 + 0.75^2(1.5) = 1.9063 \,.$$

□

- *Suggestion: Do Problems 7.1 and 7.2.*

7.2 Batching for Setup Reduction

The Batch Move Model of Sect. 7.1 was developed to model situations where individual items are batched together for the purpose of transporting them simultaneously to the next workstation. A similar situation exists when a single-unit processing workstation must be setup immediately before a group (or batch) of items of the same job type are to be processed on the machine. Frequently, this setup operation uses a significant amount of time and this can make it inefficient or even infeasible to run single unit batches. For this situation, the batching operation can be thought of as occurring at the front of this workstation rather than at the end the predecessor workstation. Accordingly, the batch forming time will be accounted for in this workstation. The reason for forming a batch of say size k is to spread the batch setup time across k jobs rather than one job. Hence, if the setup time for a certain class of items is one hour, then if these items are run “one-at-a-time” each one would add essentially one hour to their processing time. If this type of item is processed in batches of size 4, then the one hour setup time would be required only once for the batch, essentially adding 1/4 hour of setup time to each item processed instead of one hour. Thus, there are many choices for the batch size for each item and a batch quantity should be chosen that balances the setup time reduction against the increased batching delay as the batching quantity is increased. If there is only one item type, then the optimal batch size can be found by searching over the single parameter k. Of course, if there really is only one item type, the machine would always be left setup for that type and then there would not be a setup time balancing problem, unless recalibration, cleaning, or similar operation is periodically required, and this operation would usually specify the batch size k. Most realistic problems involving setups consist of at least two job types where processing alternates between types.

There are at least two different procedures possible in forming the batches. One method would be to form the batch as the individuals arrive to the workstation and another method would be to form the batch just before processing. The procedure used would depend on the physical properties of the jobs and the machining requirements. For simplicity, we will assume the first procedure, that is, batches for processing are formed as the individual jobs arrive to the workstation. We shall also begin our model development with only one job type to keep the mathematics simple. Extensions to more than one job type will be discussed later.

Let the setup batch size be k and assume that items arrive to the workstation one-at-a-time with known mean rate $\lambda(I) = 1/E[T_a(I)]$ and known inter-arrival time SCV $C_a^2(I)$. The delay associated with forming each batch is the same as previously developed and is given by Eq. (7.3).

Each batch has a service time that consists of the setup time random variable R plus k individual random services $T_{s,1}, T_{s,2}, \cdots, T_{s,k}$. The expected processing time and variance for the batch are (assuming that the service times are *i.i.d.* random variables)

$$E[T_s(B)] = E[R] + kE[T_s(I)] \tag{7.8}$$

$$V[T_s(B)] = V[R] + kV[T_s(I)] \,.$$

The squared coefficient of variation for the batch service time is then computed from the definition

$$C^2[T_s(B)] = \frac{V[T_s(B)]}{E[T_s(B)]^2} = \frac{C^2[R]E[R]^2 + kC^2[T_s(I)]E[T_s(I)]^2}{E[T_s(B)]^2} \,. \tag{7.9}$$

Further reduction of this form is not possible for the general case.

The utilization factor is slightly different from the previous result because of the necessity to account for the setup time. This is given by

$$u = \lambda(I)\frac{E[T_s(B)]}{k} = \lambda(I)\left(\frac{E[R]}{k} + E[T_s(I)]\right) \,. \tag{7.10}$$

The cycle time in the queue, CT_q, is the same as that developed for batch moves (7.4) as long as the utilization factor is computed according to (7.10). The cycle time in the system, CT_s, is the sum of the four components: the batch forming time, the cycle time in the queue for batches, the expected processing time for an individual, and the average waiting time of the individual units for their turn in service. Given the new values for the service time characteristics, the workstation cycle time when a server setup is needed per batch is given by the following property.

Property 7.4. *Consider a single-server workstation that processes jobs one-at-a-time; however, the jobs are placed in batches of size k when they enter the workstation, and a setup operation is performed on the server immediately before any of the individual jobs within each batch are processed. The mean and squared coefficient of variation of the setup operation are denoted by E[R] and $C^2[R]$, respectively. Jobs arrive to the workstation individually and are processed individually after the setup operation. The mean cycle time per job at the workstation is given by*

$$CT(I) = \frac{(k-1)}{2}E[T_a(I)] + \left(\frac{(C_a^2(I)/k) + C_s^2(B)}{2}\right)\left(\frac{u}{1-u}\right)E[T_s(B)]$$
$$+ \frac{(k+1)}{2}E[T_s(I)] + E[R]\,,$$

where batch times are given by Eqs. (7.8) and (7.9) and the utilization factor is given by Eq. (7.10).

Example 7.2. Consider finding the batch size k that results in the minimum cycle time for a single product with unit processing characteristics $E[T_s(I)] = 0.1$ hours, and $C_s^2(I) = 1.5$, and setup time characteristics $E[R] = 0.2$ hours, and $C^2[R] = 1.0$. Assume that the arrival rate of individual units is $\lambda(I) = 5.666$ per hour ($E[T_a(I)] = 0.1765$ hours), and $C_a^2(I) = 3.0$.

The workstation utilization is given by

$$u = \lambda(B)E[T_s(B)] = \frac{5.666}{k}(0.2 + 0.1k).$$

Note that the feasibility condition is that k must be large enough so that $u < 1$. For $k = 1$, $u = 1.7 > 1$, and $k = 2$ yields $u = 1.133 > 1$; hence, k must be greater than or equal to 3.

The main computational difficulty is in computing the SCV of the batch service time, $C^2[T_s(B)]$. To compute this parameter, the variance relationships

$$V[R] = C^2[R]E[R]^2,$$

$$V[T_s(I)] = C^2[T_s(I)]E[T_s(I)]^2,$$

are used, along with the fact that variances of independent variables add, to obtain

$$V[T_s(B)] = V[R] + kV[T_s(I)].$$

Then the batch service time (including setup) has a SCV of

$$C^2[T_s(B)] = \frac{V[R] + kV[T_s(I)]}{(E[R] + kE[T_s(I)])^2}.$$

The following table displays the computed information for each batch size over the range of $k \in \{3, \cdots, 9\}$. The optimal batch size occurs at $k = 6$ using the minimum CT_s as the criterion. The minimum average cycle time in this workstation is 1.860 hours per item.

□

Table 7.1 Data for varying batch sizes for Example 7.2

k	u	$E[T_s(B)]$	$C^2[T_s(B)]$	CT_s
3	0.944	0.5	0.340	6.254
4	0.850	0.6	0.278	2.460
5	0.793	0.7	0.235	1.974
6	0.755	0.8	0.203	1.860
7	0.728	0.9	0.179	1.863
8	0.708	1.0	0.160	1.917
9	0.692	1.1	0.145	1.998

7.2.1 Inter-departure Time SCV with Batch Setups

The squared coefficient of variation of the inter-departure times for the workstation with batch setups can be approximated by an *i.i.d.* departure model. We again refer to [3] for the departure model as given in the following property.

Property 7.5. *The squared coefficient of variation for the inter-departure times from a workstation that processes jobs one-at-a-time with a batch set-up is*

$$C^2[T_d(I)] = kC^2[T_a(B)](1-u^2) + k(1-u)^2 - 1 + \frac{2k(1-u)(E[R]+E[T_s(I)])}{E[T_a(B)]} + \frac{k(E[R]^2(C^2[R]+1) + kE[T_s(I)]^2(C^2[T_s(I)]+1) + 2E[R]E[T_s(I)])}{(E[T_a(B)])^2},$$

where the notation is the same as in Property 7.4.

To illustrate the accuracy of this approximation, consider the case for $k = 4$ of the previous table. A simulation of this case, using 567,715 observations for the individual inter-departure times yielded a mean value of $C^2[T_d(I)] = 2.225$ with the *i.i.d.* approximation being 2.2037, equation 7.11, which is less than -1.0% off of the measured value. The measured value for $C^2[T_s(B)]$ was 0.279 versus the computed value of 0.278.

- *Suggestion: Do Problem 7.3.*

7.3 Batch Service Model

A batch server is a processor that can process several jobs simultaneously. Ovens and metal plating operations are examples, and this is the type of operation that is analyzed in this section. Namely, we consider a batch server model where a fixed number of items are loaded and processed at the same time. Processing is not started until the number of units in the batch, say k, are available. The batch is then loaded and held in the server for the allotted time. At the completion of service, the batch is removed from the server and the units either as a group or individually are sent to their next workstation. Different types of items may be grouped together or the batching operation may be restricted to item specific groups.

7.3.1 Cycle Time for Batch Service

The arrival rate for batches and the associated squared coefficient of variation of inter-arrival times of the batches, as they are related to the individual flow characteristics, were developed in Sect. 7.1.1. The modeling approach for a general arrival and general service situation is to utilize the $G/G/1$ cycle time approximation, using batch timing characteristics in place of individual item information. Previously the adjustment equations for batch arrival data were developed given individual inter-arrival time information. These relationships are

$$\lambda(B) = \frac{\lambda(I)}{k} = \frac{1}{kE[T_a(I)]}, \tag{7.11}$$

$$C^2[T_a(B)] = \frac{C^2[T_a(I)]}{k}. \tag{7.12}$$

The service time data, namely $E[T_s(B)]$ and $C^2[T_s(B)]$, are characteristics of the job and processor and so are known data. The workstation cycle time for batch processing is given by the following property.

Property 7.6. *Consider a single-server workstation that processes jobs in fixed batches of size k. Jobs arrive to the workstation individually. Upon entering the workstation, the individual jobs are placed in batches before proceeding into the workstation. Service cannot start until a full batch is available. The mean and SCV of batch processing times are denoted by $E[T_s(B)]$ and $C_s^2(B)$, respectively. The mean system cycle time per job at the workstation is given by*

$$\begin{aligned} CT_s = {} & \frac{(k-1)}{2} E[T_a(I)] + E[T_s(B)] \\ & + \left(\frac{C^2[T_a(B)] + C^2[T_s(B)]}{2} \right) \left(\frac{u(B)}{1-u(B)} \right) E[T_s(B)] \,, \end{aligned}$$

where the utilization factor, $u(B)$, is computed as $u(B) = \lambda(B)E[T_s(B)]$.

Notice that there are some adjustments that might be necessary when applying Property 7.6. It is possible that the batches are already formed so that arrivals are by batches instead of individually. If the batches were formed at the previous workstation, then the $E[T_a(I)]$ expression in the first term should refer to the departure rate of individuals from the previous processor and no change is needed in the formula. If the previous workstation was a batch processor of the same size, the first term should be deleted. Finally, we should consider the case where batches are formed when the jobs are ready to be processed. Then if the utilization factor for the job type is high, the batch formation time would be greatly reduced since formation would occur naturally while jobs are waiting in the queue. To approximate this situation, we multiply the first term (the batch formation time) by the factor $(1-u(B)^2)$ where the utilization factor refers only to one specific job type if there were multiple types.

7.3.2 Departure Process for Batch Service

The modeling difficulty for batch service arises when the batch is unloaded and the individual items are moved into their subsequent workstations. Without a branching split after the batch processing workstation, all of the batched items would proceed on to the same next workstation. Then this workstation would see individual items arriving but with unusual inter-arrival time characteristics. To illustrate this point, consider that a batch workstation processing batches of 4 items and that the random inter-departures times for three specific batches are T_1, T_2, and T_3. The next workstation sees, for these three batches, individual items with the following sequence of inter-arrival times: $T_1, 0, 0, 0, T_2, 0, 0, 0, T_3, 0, 0, 0$. This sequence does not possess the same inter-arrival time characteristics as the batch process itself. In fact the individual items are not independently arriving units. This situation for the receiving workstation is actually the batch move model as developed in Sect. 7.1.

The *i.i.d.* approximation for the squared coefficient of variation (SCV) equation for the departure of individuals from a batch service workstation is

$$C^2[T_d(I)] = kC^2[T_d(B)] + k - 1\ , \tag{7.13}$$

which can be written in terms of the basic workstation characteristics for batches.

Property 7.7. *Assume a batch service with the same notation as in Property 7.6. The squared coefficient of variation of the inter-departures of individuals from the workstation is approximated by*

$$C_d^2(I) = k[(1-u^2)C_a^2(B) + u^2 C_s^2(B)] + k - 1\ .$$

Although Property 7.7 gives an SCV for the renewal process (Definition 5.1) approximating the departure process, observe that the process formed by individual departures of a batch operation is clearly not a renewal process so that modeling the next workstation could be problematic. This is the topic of the next section.

Property 7.7 illustrates that after a batch service, the process of separating the batch into individual items causes the squared coefficient of variation to increase significantly as a function of the batch size parameter k. One would expect the multiplication of $C^2[T_d(B)]$ by the batch size k since this reverses the batching process adjustment. However, the additional factor $k-1$ indicates that the batch process changes the system's flow characteristics significantly. Again this SCV approximation ignores any dependencies in the inter-departure stream and treats each item's inter-departure time as independent and identically distributed.

Example 7.3. Consider a batch-processing workstation in which arrivals to the workstation are from another batch server so that the arrivals occur in batches of size 5 with a mean rate of 3 batches per hour and an SCV of batch inter-arrival times of 0.75. The SCV of the batch service time is also 0.75 with a workstation load factor or utilization of 84%. The average time each job spends in the workstation is thus given by

$$CT = 16.8 + \frac{0.75+0.75}{2} \times \frac{0.84}{1-0.84} \times 16.8 = 82.95 \text{ min} .$$

Note that 16.8 is the service time given in minutes and is obtained by dividing the utilization factor by the arrival rate. In addition, the first term in the cycle time equation of Property 7.6 was not used since the arrivals were already in batches.

Since the SCV of the inter-arrival times and the service times are the same, it is also the SCV of the inter-departure times for batches. The approximation for the inter-departure SCV of individuals is given by

$$C_d^2(I) = 5\left[(1-0.84^2)(0.75) + 0.84^2(0.75)\right] + 4 = 7.75.$$

□

Although the SCV calculation of 7.75 of inter-departures of Example 7.3 may give an accurate representation of the actual departure stream, there are major problems when using this value in a cycle time calculation for the downstream workstation. The next section discusses how to better utilize batch output when modeling downstream workstations.

- *Suggestion: Do Problems 7.4–7.6.*

7.4 Modeling the Workstation Following a Batch Server

Since the unbatching process after a batch-service workstation does not produce a renewal process (i.e., a stream of inter-departure times that are independent and identically distributed), it is prudent to model the recipient workstation as having a batch arrival process. This approach captures the true behavior of the arrival stream including the inter-dependence between arrivals, whereas the *i.i.d.* SCV approximation does not. The analysis of the workstation depends on whether the jobs leaving the batch processor are sent directly to the next workstation or if a probabilistic branch follows the batch server.

7.4.1 A Serial System Topology

To illustrate the difficulties inherent in modeling the workstation following a batch service operation, reconsider Example 7.3 and assume that the workstation of that example feeds into a workstation that processes jobs one-at-a-time. This downstream workstation has a service process described by an exponential distribution with a mean of 3 minutes. The workstation of Example 7.3 is feeding batches of size 5 to our workstation at a mean rate of 3 batches per hour with an SCV of batch inter-arrival times of 0.75 and an SCV of 7.75 of individuals. The workstation utilization factor is $u = 5 \times 3 \times 0.05 = 0.75$. Then using the standard approximation for individuals, the system cycle time for this workstation is

$$CT = \frac{7.75+1}{2} \times \frac{0.75}{1-0.75} \times 0.05 + 0.05 = 0.706 \text{ hr} . \tag{7.14}$$

Simulating this situation yielded a cycle time estimate of 0.496 hours based on just over 135,000 simulated services. The resultant inter-arrival stream characteristics were measured as $E[T_a(I)] = 0.333$, with $C^2[T_a(B)] = 0.753$. The workstation utilization factor was measured to be 0.749. This data was from a simulation run length of 10,000 hours with a statistics reset after 1,000 hours. So even though the first two moments of the inter-arrival time distribution were very accurate, the cycle time approximation was off by 42%. Why? The answer is in the arrival stream's characteristics. The first two moments of the inter-arrival time distribution does not capture the grouping behavior observed in the batch arrival process. For example, it is quite possible to have an arrival stream sequence of batches of size 2 with exactly the same mean and SCV; thus, it is clear that the first two moments alone cannot adequately describe an arrival process of individuals that arise from batches.

To properly take the arrival stream's characteristics into account, the next workstation after the batch server workstation should be modeled as a batch arrival station, using the modeling approach detailed in Sect. 7.1 dealing with batch moves, specifically, Property 7.2 should be used. This approach treats the system as a batch arrival and batch server system for the queue time estimate and then adds the individual unit's service time plus the extra waiting time due to each unit waiting its turn for processing within the batch.

The cycle time given by Eq. (7.14) can be re-calculated using Property 7.2 as follows:

$$CT = \frac{5 \times 0.75 + 1}{2} \times \frac{0.75}{1 - 0.75} \times 0.05 + \frac{5+1}{2} \times 0.05 = 0.506 \text{ hr} .$$

This result differs by 2% from the simulated value. Thus, the batch modeling approach is much closer to the observed cycle time since this model has inherent in it the behavior of items arriving for service from the batch server workstation.

7.4.2 Branching Following a Batch Server

From previous discussion (Sect. 7.4.1), we know that the proper method of modeling a workstation following a batch service process is to treat the output process as a batch move. Thus, units coming out of the batch server move to the next workstation in a fashion identical to the batch move model (Sect. 7.1). The question arises as to the appropriate modeling method if the departures from the batch server follow a probabilistic branch that separates individual items and distributes them to different workstations.

Consider for illustration purposes a batch server (Workstation 1) with a batch size of 4. Let p be the probability that an individual item from a batch goes next to Workstation 2 and let $q = 1 - p$ be the probability that an individual job goes to Workstation 3. Consider further a batch that is just exiting the batch server and the batch is immediately broken into individual items that are randomly branched to either Workstation 2 or Workstation 3, with random variables N_2 and N_3 denoting the number of jobs sent to the two workstations. Thus, $N_2 \in \{0,1,2,3,4\}$ and $N_3 = 4 - N_2$. If Workstation 2 receives 1 job then Workstation 3 receives 3 jobs from this batch. In other words, the workstations receiving output from a batch service operation followed by a probabilistic split see random sized batches. Figure 7.2 illustrates this idea with items being represented by small circles stacked on one another to indicate that they arrived at the same time.

A specific batch (of size 4) split between the two workstations can take any of the configurations shown in Table 7.2. So in essence, each workstation sees a binomial distribution of random batch sizes (refer to p. 14), in this example ranging from 1 to 4. Note that a zero batch size means that no arrival occurs for at least one more batch service.

The modeling approach is again to treat the arrival pattern as batches but now of random sizes, $N \in \{1, 2, \cdots, k\}$. However, because the workstation accepting the batches does not "see" batches of size zero, the distribution of batches sizes must be a conditional binomial distribution given that the random variable is not zero. Thus, the probability density function for the random batch size N is

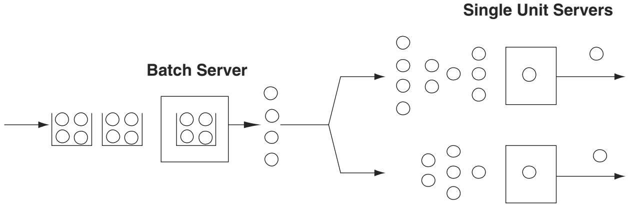


Fig. 7.2 Illustration of a batch service followed by individual (random) branching to subsequent workstations, where stacked jobs denote batches with a size equal to the number of stacked items

Table 7.2 Distribution of batch sizes to the workstations

Probability	WS 2	WS 3	Probability
$\binom{4}{1} p^1 q^3$	o	ooo	$\binom{4}{3} q^3 p^1$
$\binom{4}{2} p^2 q^2$	oo	oo	$\binom{4}{2} q^2 p^2$
$\binom{4}{3} p^3 q^1$	ooo	o	$\binom{4}{1} q^1 p^3$
$\binom{4}{4} p^4 q^0$	oooo		$\binom{4}{0} q^0 p^4$
$\binom{4}{0} p^0 q^4$		oooo	$\binom{4}{4} q^4 p^0$

$$\Pr\{N=n\} = p_n = \binom{k}{n} \frac{p^n q^{k-n}}{(1-q^k)} \quad \text{for } n = 1, \cdots, k\,, \tag{7.15}$$

where k is the fixed processing batch size and $q = 1 - p$. The mean and SCV for this conditional binomial distribution are not too difficult to determine and are given by

$$E[N] = \frac{kp}{1-q^k} \quad \text{and} \tag{7.16}$$

$$C^2[N] = \frac{q\left(1 - kq^{k-1} + (k-1)q^k\right)}{kp}\,. \tag{7.17}$$

As with fixed batch sizes, the cycle time model is separated into the cycle time in the queue for batches (working their way up to the server) and the average service delay for individuals within the batch. The service delay consists of the item's service time T_s plus the service time of all items in the batch that are processed before this specific individual. These times need to be averaged over all possible positions for items within the batch. These two components are addressed individually.

7.4.2.1 Cycle Time in the Queue for Random Sized Batches

A queued batch can be viewed as seeing batches ahead of it being served as a whole even though the service mechanism operates on individual items taken from a batch. The expected delay time is computed for a batch to move up in the first-come-first-serve queue until it (any of its items) becomes the batch being served. (This is the same model form used for the Batch Move Model of Sect. 7.1). The service time for each batch is a random sum of *i.i.d.* random variables which was described in Sect. 1.6.2. Thus, the measures for the batch service time are obtained through Property 1.9 and are given as

$$\begin{aligned} E[T_s(B)] &= E[N]E[T_s(I)] \quad \text{and} \\ C^2[T_s(B)] &= C^2[N] + \frac{C^2[T_s(I)]}{E[N]}\,, \end{aligned} \tag{7.18}$$

where N is the random variable denoting the batch size (Eq. 7.15).

The arrival rate of individual items at the workstation following a batch service with branching probability p is a function of the arrival rate of individuals into the batch service workstation. For notational clarification denote the batch server workstation as #1 and the recipient workstation as #2. Let $\lambda_1(I)$ be the arrival rate of individuals into the batch workstation and let $\lambda_2(I)$ be the arrival rate of individuals into #2 after the branch. Then the relationship between these rates is

$$\lambda_2(I) = p\lambda_1(I)\,. \tag{7.19}$$

However, the batch arrival rate is not quite as straight forward because of the possibility that a batch is of size zero. Assume that #1 operates on batches of size k, then the probability that a batch of size zero is "sent" to #2 is q^k where $q = 1 - p$. Therefore, the probability that a batch of size greater than zero departs from #1 is $1 - q^k$ so that the batch arrival rate to #2 is given by

$$\lambda_2(B) = \left(1 - q^k\right)\frac{\lambda_1(I)}{k}\,, \tag{7.20}$$

and the expected batch size is given by Eq. (7.16).

The squared coefficient of variation of the inter-arrival time into #2 is related to the squared coefficient of variation of the inter-departure time for #1. Since the departures are in terms of batches, the batch inter-arrival time's squared coefficient of variation into #2 is

$$C_B^2[T_a(2)] = (1 - q^k)C_B^2[T_d(1)] + q^k\,, \tag{7.21}$$

where the subscript B is used to indicate that the C^2 is for batches. Recall that the factor q^k is the probability that no units from the batch are directed to #2.

Using the standard Kingman approximation, the cycle time in the queue, CT_q, for the batches is given as

$$\begin{aligned} CT_q(2) &= \left(\frac{C_B^2[T_a(2)] + C^2[T_s(B)]}{2}\right)\left(\frac{u_2}{1-u_2}\right) E[T_s(B)] \qquad (7.22) \\ &= \left(\frac{C_B^2[T_a(2)] + C^2[N] + C^2[T_s(I)]/E[N]}{2}\right)\left(\frac{u_2}{1-u_2}\right) E[N]E[T_s(I)]\,, \end{aligned}$$

where the utilization factor is computed by $u_2 = \lambda_2(I)E[T_s(I)]$.

- *Suggestion: Do Problem 7.13.*

7.4.2.2 Average Service Delay Times for Random Sized Batches

Once a batch has worked its way through the batch queue and finally has command of the server, the server will be busy for the specific number of service times equal to the number of items in the batch. The delay time associated with individual items within the batch varies since processed items leave the workstation immediately upon completion of their turn in the server. Thus, an average delay is computed by taking into account the delay associated with each position, with respect to the order that items are served, within the batch. This average delay has two components, the service time of the individual and the average delay waiting for other items positioned ahead of that individual unit in the batch.

We follow the same logic here that was used in Sect. 7.1.3. The random variable D represents the total delay experienced by all jobs within a batch; thus from Eq. (7.5), it follows that

$$E[D|N=n] = \frac{n(n-1)}{2}E[T_s(I)]\,,$$

where the random variable N is the size of the batch. Since $E[E[D|N]] = E[D]$ (see Property 1.8), the service time delay, st, is obtained as follows

$$\begin{aligned} st &= \frac{E[D]}{E[N]} + E[T_s(I)] = \frac{E[N(N-1)]}{2E[N]}E[T_s(I)] + E[T_s(I)] \qquad (7.23) \\ &= \frac{E[N(N+1)]}{2E[N]}E[T_s(I)] = \frac{E[N^2] - E[N]^2 + E[N]^2 + E[N]}{2E[N]}E[T_s(I)] \\ &= \left(\frac{E[N]+1}{2} + \frac{V[N]}{2E[N]}\right)E[T_s(I)] = \left(\frac{1+E[N]+E[N]C^2[N]}{2}\right)E[T_s(I)]\,. \end{aligned}$$

Notice that for deterministic batches, Eq. (7.23) is identical to Eq. (7.6).

7.4.2.3 Cycle Time in the Workstation for Random Sized Batches

The cycle time in a workstation that directly follows a batch server and receives only a proportion of the individual items can be obtained by combining the two major pieces of the previous two sections yielding the following property.

Property 7.8. *Consider a workstation that processes items one-at-a-time with a mean and SCV of the (individual) processing time given by $E[T_s]$ and C_s^2. Jobs arrive to the workstation in batches of random size denoted by N. The times between batch arrivals have a mean and SCV of $E[T_a(B)]$ and $C_B^2[T_a]$ yielding a mean batch arrival rate of $\lambda(B) = 1/E[T_a(B)]$. The mean system cycle time per item at the workstation is approximated by*

$$CT_s = \left(\frac{E[N]\,C_B^2[T_a] + E[N]\,C^2[N] + C_s^2}{2}\right)\left(\frac{u}{1-u}\right)E[T_s]$$
$$+\left(\frac{1+E[N]+E[N]\,C^2[N]}{2}\right)E[T_s]$$

where the utilization factor is $u = E[N]\,\lambda(B)\,E[T_s]$.

Example 7.4. Consider a workstation that processes 5 units simultaneously ($k = 5$). Let the departure rate from this batch workstation be 3 batches per hour with a squared coefficient of variation of the inter-departure times of 0.75. After the batch leaves the workstation, it is broken into individual units and each item has a 25% chance of being sent to the second workstation. The second workstation processes items one-at-a-time according to an exponential distribution with mean of 12 minutes. We would like to analyze the second workstation.

The probability of #2 not receiving any units from a particular batch that finished processing at #1 is $0.75^5 = 0.2373$. Thus, the arrival rate of of batches (of any size) to #2 is $\lambda_a(B) = 3(1-0.2373) = 2.288$ per hour (see Eq. 7.20), or equivalently, $E[T_a(B)] = 0.437$ hours. Note that the arriving batch can be of any size from 1 to 5 units depending on the probabilistic results from individual unit branching. The mean and SCV for the batch size (Eqs. 7.16 and 7.17) are $E[N] = 1.639$ and $C^2[N] = 0.220$. The mean batch size (1.639) together with the batch arrival rate (2.288/hr) and the mean service time (0.2 hr) results in a utilization factor of $u = 0.75$.

The arrival rate of individual units is the average batch size (1.639) times the batch arrival rate (2.288/hr) yielding 3.75/hr which is also equal to the branch probability (0.25) times the individual departure rate from the first workstation. The SCV of the inter-arrival times of batches is determined from the departure process according to Eq. (7.21):

$$C_B^2[T_a] = (1-0.2373)0.75 + 0.2373 = 0.809.$$

Property 7.8 can now be used to determine the average time spent within the workstation for an arbitrary job:

$$CT = \left(\frac{1.639 \times 0.809 + 1.639 \times 0.220 + 1}{2}\right)\left(\frac{0.25}{1-0.25}\right)0.2 + \left(\frac{1 + 1.639 + 1.639 \times 0.220}{2}\right)0.2 = 1.106 \text{ hr}\,.$$

The approximations agree quite well with simulated results for this system. Table 7.3 contrasts several of these computations with the measured results from a simulation study. The simulation study consists of the 20 replications of individual simulations 10,000 hours in length, with a statistical reset at 1,000 hours. Each replication results in 20,600 to 20,700 processed batches (or more than 400,000 batches).

□

Table 7.3 Comparison of analytical approximation and simulation results for Example 7.4

	$E[T_a(B)]$	$C_B^2[T_a]$	$E[T_s(B)]$	$C^2[T_s(B)]$	CT_q	CT_s
Analytical Mean	0.437	0.809	0.300	0.830	0.806	1.106
Simulated Mean	0.437	0.813	0.301	0.835	0.805	1.106
Simulated Std.Dev.	0.003	0.011	0.002	0.009	0.044	0.045

The conclusion is that this model is the appropriate method for modeling the downstream server from a batch workstation with individual unit branching to the next workstation. This approach is considerably better than using an *i.i.d.* coefficient of variation formula to compute the individual inter-arrival time parameters and then applying a model for individual cycle times. The down side of this approach is that it is harder to incorporate into a network model because of the more complex connections between workstations and if the workstation has multiple inflows the blending of the streams becomes considerably more complicated.

- *Suggestion: Do Problems 7.7 and 7.8.*

7.4.2.4 Arrival SCV of Individuals after a Random Branch

If it is necessary to compute the squared coefficient of variation of the arrival stream of individuals coming from a batch service process, then the best one can do is an *i.i.d.* approximation for the SCV. This treatment considers all the individuals as independent arrivals and merely computes the associated SCV of the individual inter-arrival times.

To effect this computation, we repeat Eq. (7.21) from Sect. 7.4.2.1 that gives the relationship between departing batches from one workstation to arriving batches to the next workstation:

$$C_B^2[T_a(2)] = (1-q^k)C_B^2[T_d(1)] + q^k \ ,$$

where p is the branching probability with $q = 1-p$. Equation (7.13) gives the conversion from a batch to individual jobs, so by taking a weighted average of the squared coefficients of variation, we have

$$C_I^2[T_a(2)] = E[N]C_B^2[T_a(2)] + E[N] - 1 \ ,$$

where N is the random variable of the resulting batch size after the probabilistic branching with distribution given by Eq. (7.15). Using the mean value from Eq. (7.16), the SCV of the inter-arrival times of individually arriving jobs is calculated as

$$C_I^2[T_a(2)] = \frac{kp\left[(1-q^k)C_B^2[T_d(1)] + q^k + 1\right]}{1-q^k} - 1 \ , \tag{7.24}$$

where k is the batch size of the departing batches from the first workstation before branching occurs.

Note that a better approximation occurs if a single unit processing workstation that follows a batch server is modeled using the Batch Move Model that resulted in Property 7.8. Equation 7.24 is given for the situation where it is difficult to model the next workstation with the batch move approach and/or there are several sources of inflow into this workstation that must be combined.

Example 7.5. To illustrate obtaining the inter-arrival time SCV for individuals randomly branched to a workstation from a batch service workstation, consider that the batch workstation has an SCV for inter-departure times of batches given as $C_B^2[T_d(1)] = 0.8$, and let the branching probability be 1/2 for individual units from batches of size 4. Then $q^4 = 0.5^4 = 0.0625$ and

$$C_I^2[T_a(2)] = \frac{4(1/2)\,[0.9375(0.8) + 0.0625 + 1]}{0.9375} - 1 = 2.8667.$$

This agrees quite well with the simulated result of 2.87. □

7.4.2.5 Departures from the Workstation Following Batch Service

The mean and squared coefficient of variation of an arrival process sometimes does not adequately capture the arrival stream's characteristics from an accurate modeling prospective. This is particularly true for batch arrival streams. The batch process cycle time is relatively easy to characterize but the output process from the batch service workstation cannot be adequately characterized with only the mean and SCV parameters. The batch arrival phenomenon to a single unit service workstation requires a separate model (the Batch Move Model of Sect. 7.1), and the random batch size extension given in Property 7.8. Individual units depart these workstations and merge with other inflow streams to subsequent workstations. So the question of an adequate model for approximating the outflow or departure stream of individuals needs to be addressed. Curry and Deuermeyer [3] show that a simple extension to Property 7.3 yields a relatively accurate approximation for a workstation downstream from a batch processor that has probabilistic branches.

Property 7.9. *Consider a workstation with batch arrivals that processes items one-at-a-time. Using the same notation as in Property 7.8, the squared coefficient of variation of the inter-departures of individuals from the workstation is approximated by*

$$C_d^2(I) = (1-u^2)E[N]\,C_B^2[T_a] + (E[N]-1)\,(1-u)^2 + u^2 C_s^2\ .$$

Example 7.6. We return to Example 7.4 and determine the characteristics of the departure process from the second workstation (i.e., the workstation that was accepting 25% of the items departing from the batch processor). The batch size characteristics were computed to be $E[N] = 1.639$ and $C^2[N] = 0.220$ and the batch arrival rate was determined to be $\lambda_a(B) = 2.288$/hr. Therefore the individual arrival rate and thus the departure rate of items from the workstation is $1.639 \times 2.288 = 3.75$/hr which yields

$$E[T_d(I)] = 16 \text{ min}\ .$$

We also have from Example 7.4 that the SCV for the batch arrival process to the workstation was 0.8093; therefore the SCV of the inter-departure times of individual items from the workstation is

$$C_d^2(I) = 1.639(1-0.75^2)0.8093 + (1.639-1)(0.25)^2 + 0.75^2(1) = 1.183\ .$$

□

In a simulation study of the departures from this random batch arrival system with individual service, a set of 13 simulations with random batch sizes resulting from a service batch of size 5, a 25% change of individuals being routed to the workstation being studied and, $C_a^2(B)$ and $C_s^2(I)$ both ranged over 3/4, 1, and 3/2. Each of these simulations consisted of 100,000 simulated hours. The average absolute error between the theoretical estimate and the simulation estimate for $C_d^2(I)$ for these 13 studies was 1.80% with a maximum error of 3.03%. Although this study also was not over the whole range of values for utilization, and inter-arrival and service time SCV's, it does indicate that the *i.i.d.* approximation given in Property 7.9 for the SCV of departures is a viable approach for modeling purposes.

- *Suggestion: Do Problem 7.11(a).*

7.5 Batch Network Examples

Modeling the flow of jobs within a factory in which batching occurs can complicate the methodology considerably. To help in incorporating batches within your models, we devote this final section to the analysis of two different factories with batching. The first example includes all three major batch types and without any feedback paths. The second example includes a more complex branching structure that has reentrant flows. The main concept that is demonstrated with these examples is that the formulas from the various properties cannot be blindly applied — they often must be adjusted slightly to fit different situations. However, the bottom line is that systems with batching can be reasonably well analyzed if the various models discussed in this chapter are used wisely.

7.5.1 Batch Network Example 1

Consider Fig. 7.3 that has three workstations each of which operates using a different form of the batch service models studied in this chapter. The first workstation is a setup-batch processing workstation, the second workstation uses oven-batch processing, and the third workstation is a single unit server with batch arrivals (the batch move model). Inflow into the system is in terms of individual units with a Poisson arrival rate with a mean of 5 units per hour. These individuals are immediately batched into groups of $k = 3$ and transported into the first workstation. The batch forming time in this analysis will be added to the cycle time for Workstation 1. The data for each particular workstation is given as it is needed in the solution process.

Note in Fig. 7.3 that once batches are formed, they remain batches until they either exit the system following processing at Workstation 2 or they make it to Workstation 3. The jobs are large and require a forklift for transportation between workstations, thus all probabilistic branches are made on batches and not individual jobs. Specifically, 2/3 of the batches leaving Workstation 1 are routed to Workstation 2 and 1/3 go to Workstation 3. From Workstation 2, 1/4 of the batches are finished (leave the system) and the remaining 3/4 are routed to Workstation 3. At Workstation 3, the items are then separated once the batch enters service and, subsequently, they leave as individuals. The goal of this analysis is to obtain the expected cycle time and throughput rate for the system as a whole.

Workstation 1 Including Batch Forming Time

The arrival rate of individuals to the first workstation is according to a Poisson process with a mean rate of 5 per hour. The average batch forming time, BT, to be associated with each individual item is determined by the equation

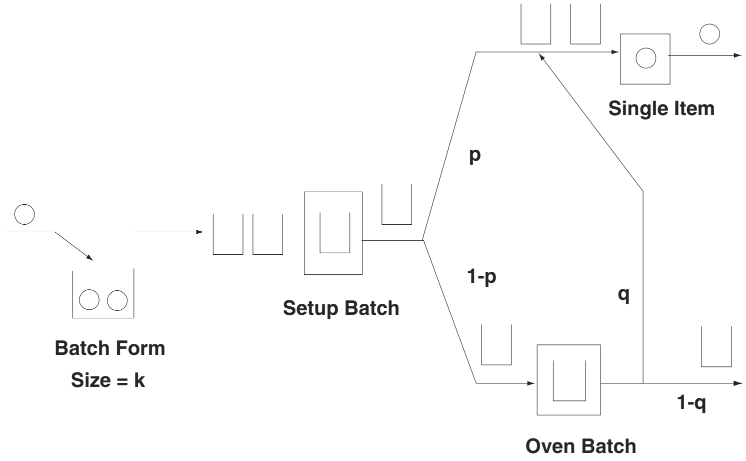


Fig. 7.3 Example manufacturing system where each workstation in the facility uses a different form of the batch processing models

$$BT_{\text{start}} = \frac{(3-1)}{2}\frac{1}{5} = 0.2 \text{ hr} ,$$

where the 1/5 hour is the mean inter-arrival time. The arrival rate of batches to Workstation 1 is the individual arrival rate divided by the batch size yielding $\lambda_1(B) = 5/3$ per hour. Since the external arrival process is Poisson, the squared coefficient of variation of the arrival stream of individuals is 1. Thus, the SCV for the inter-arrival time of batches is

$$C_B^2[T_a(1)] = \frac{1}{3} .$$

The first workstation is a setup batch system where a setup is required for every three jobs. The setup time has a mean of 12 minutes and a variance of 1080 minutes^2. After the setup, jobs are processed one-at-a-time with a mean processing time of 6 minutes and a variance of 240 minutes^2. Thus the characteristics for processing the entire batch is given as

$$\begin{aligned} E[T_{s,1}(B)] &= 12 + 3 \times 6 = 30 \text{ min} = 0.5 \text{ hr} \\ V[T_{s,1}(B)] &= 1080 + 3 \times 240 = 1800 \text{ min}^2 = 0.5 \text{ hr}^2 \\ C_{s,1}^2(B) &= \frac{0.5}{0.5^2} = 2 . \end{aligned}$$

The workstation utilization is $u_1 = \lambda_1(B) \times E[T_{s,1}(B)] = 0.8333$.Before using Property 7.4 to determine the average cycle time within the first workstation, we need to add a batch forming time after processing. Because all jobs are moved between workstations by a batch move, we add the batch forming time to the end of this cycle time. (Although the formula of Property 7.1 includes the batch forming time as part of the next workstation, it easier to include with the first workstation's cycle time because batches to the third workstation come from two different sources.) The batch forming time after processing is given by

$$BT_{\text{fin}} = \frac{(3-1)}{2}0.1 = 0.1 \text{ hr} ,$$

where 0.1 refers to the individual mean processing time. Thus, using Property 7.4, the cycle time for per job in the first workstation is computed as

$$CT(1) = BT_{\text{start}} + \frac{1/3+2}{2}\frac{0.8333}{1-0.8333}0.5 + \frac{3+1}{2}0.1 + 0.2 + BT_{\text{fin}} = 3.616 \text{ hr} .$$

The departing squared coefficient of variation from Workstation 1 (in terms of batches) is determined by the standard approximation (Property 5.2)

$$C^2_{d,1}(B) = (1-0.833^2)\left(\frac{1}{3}\right) + 0.833^2(2) = 1.491 .$$

The proportion of this output stream of batches that goes to Workstation 2 is 2/3 while 1/3 goes to Workstation 3. Thus, the two branches from Workstation 1 will have the following characteristics (Property 5.6) as arrival streams to the other two workstations:

$$\begin{aligned} \lambda_{1\to 2}(B) &= \frac{2}{3} \times \frac{5}{3} = 1.111/\text{hr} \\ C^2_{a,1\to 2}(B) &= \frac{2}{3}(1.491) + \frac{1}{3} = 1.327 \quad \text{and} \end{aligned}$$

$$\begin{aligned} \lambda_{1\to 3}(B) &= \frac{1}{3} \times \frac{5}{3} = 0.556/\text{hr} \\ C^2_{a,1\to 3}(B) &= \frac{1}{3}(1.491) + \frac{2}{3} = 1.164 . \end{aligned}$$

Workstation 2 Oven Batch Processing

The second workstation is an oven batch service process with a mean time of 48 minutes and a service SCV of 0.75. The only jobs coming into Workstation 2 come from Workstation 1, so the arrival process characteristics are those calculated previously from the Workstation 1 departure stream; thus, we have $E[T_{a,2}(B)] = 0.9$ hours and $C^2_{a,2}(B) = 1.327$. The utilization for the workstation is $u_2 = 1.111 \times 0.8 = 0.889$. (Do not forget to make units consistent by converting 48 minutes to 0.8 hours.) The formula of Property 7.6 is used after deleting the first term since the batch forming occured and was counted in Workstation 1; thus, the cycle time calculation is

$$CT(2) = 0.8 + \frac{(1.327+0.75)}{2}\,\frac{0.889}{1-0.889}\,(0.8) = 7.454 \text{ hr} .$$

The inter-departure time SCV for Workstation 2, again in terms of batches, is

$$C^2_{d,2}(B) = (1-0.889^2)\times 1.327 + 0.889^2 \times 0.75 = 0.871 .$$

The proportion of this departure stream that is branched (again full batch branching) is 3/4; thus,

$$\begin{aligned}\lambda_{2\to 3}(B) &= \frac{3}{4}\times 1.111 = 0.833/\text{hr}\\ C^2_{a,2\to 3}(B) &= \frac{3}{4}(0.871) + \frac{1}{4} = 0.903 .\end{aligned}$$

Workstation 3 Batch-Arrival Individual-Service

The arrival of batches into Workstation 3 comes from both Workstations 1 and 2; therefore, the total mean arrival rate is given by

$$\lambda_3(B) = \lambda_{1\to 3}(B) + \lambda_{2\to 3}(B) = 0.556 + 0.833 = 1.389/\text{hr} .$$

The SCV of the arrival stream is approximated by a weighted average of the two streams that merge (Property 5.5) yielding

$$\begin{aligned}C^2_{a,3}(B) &= \frac{0.556}{1.389}\times C^2_{a,1\to 3}(B) + \frac{0.833}{1.389}\times C^2_{a,2\to 3}(B)\\ &= 0.4\times 1.164 + 0.6\times 0.903 = 1.007 .\end{aligned}$$

The service process at Workstation 3 is for individual items; hence, the Batch Move Model of Sect. 7.1 is used to determine cycle time. The mean and standard deviation of the individual processing times are 12 and 8.458 minutes, respectively. The utilization factor for the workstation is $u_3 = 3\times 1.389\times 0.2 = 0.833$/hr, and the application of Property 7.2 yields the mean time that a job spends within Workstation 3 as

$$CT(3) = \frac{3\times 1.007 + 0.5}{2}\,\frac{0.833}{1-0.833}\,0.2 + \frac{3+1}{2}\,0.2 = 2.156 \text{ hr} .$$

Note that the SCV for the service time is the square of the standard deviation divided by the mean.

System Measures

The throughput rate of individual items for this system has to equal the arrival rate of 5 jobs per hour. The cycle time for the system including batches that exit from

Workstation 2 is determined by computing the work-in-process (WIP) at each workstation, then summing to obtain the system WIP. From the system WIP, using Little's Law ($WIP = th \times CT$), the cycle time for individuals is determined. Note that the WIP is to be computed in individual units so that the cycle time for individuals in the system can be computed. The data needed for this analysis and the results are contained in Table 7.4.

Table 7.4 WIP calculations for the example of Sect. 7.5.1

Workstation i	$\lambda_i(I)$	$CT(i)$	$WIP(i)$
1	5/hr	3.616 hr	18.08
2	3.333/hr	7.454 hr	24.84
3	4.167/hr	2.156 hr	8.98

Table 7.5 Transportation time calculations for the example of Sect. 7.5.1

From/To	Move Rate	Travel Time	WIP
Entrance to WS 1	5/hr	5 min	0.417
WS 1 to WS 2	3.333/hr	8 min	0.444
WS 1 to WS 3	1.668/hr	9 min	0.250
WS 2 to WS 3	2.499/hr	6 min	0.250

Thus, the total system WIP is 51.9 jobs and the average cycle time for individual jobs through this system regardless of their exit point is 51.9/5 = 10.38 hours. These calculations ignored all transportation times. If we assume a sufficient number of forklifts so that there is no waiting when a batch is ready to be moved, it is relatively easy to include the time necessary for batch moves. Table 7.5 shows the data and the calculations needed to include the transportation time needed for the forklifts to move the various jobs between workstations.

From the analysis contained in Table 7.5, we have that there is an average of 1.36 jobs within the transportation system of the factory. Thus, the total WIP in the factory is 53.26 jobs and the mean cycle time, including move times, is 10.65 hours.

7.5.2 Batch Network Example 2

Consider the network given in Fig. 7.4. The first workstation has two processing machines, the second workstation has one machine, and the third workstation has an oven process that serves three units simultaneously. The arrival rate of jobs into this system is a Poisson process with jobs entering at Workstation 1 at a mean rate of 10 jobs per hour. The data for the three workstations are given in Table 7.6.

Because there are reentrant flows within this factory, the total arrivals rates must be determined using a routing matrix. These probabilities given in Fig. 7.4 and result

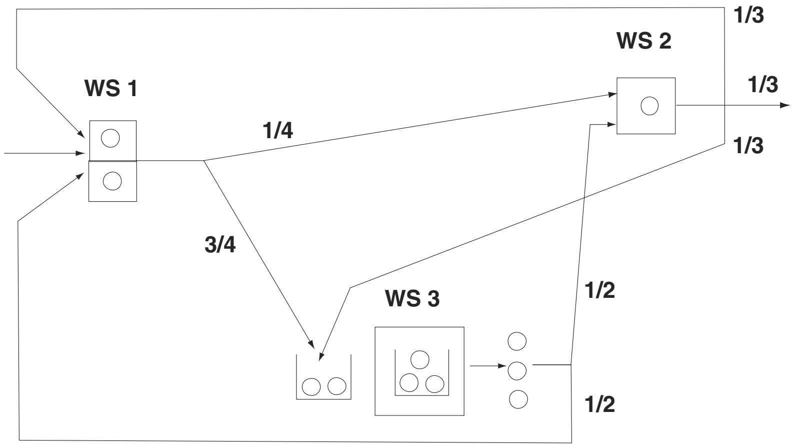


Fig. 7.4 Manufacturing system with batch processing at Workstation 3; batches are formed in front of Workstation 3 and individual items are shipped out from 3; Workstation 1 has two machines and Workstation 2 has one machine

Table 7.6 Data for the example of Sect. 7.5.2

Workstation	External Arrival Rate	Number of Machines	Batches Size for Processing	$E[T_s]$	$C^2[T_s]$
1	10/hr	2	1	2.4 min	1
2	0	1	1	1.714 min	1
3	0	1	3	3.75 min	1

in the following:

$$P = \begin{bmatrix} 0 & 0.25 & 0.75 \\ 0.333 & 0 & 0.333 \\ 0.5 & 0.5 & 0 \end{bmatrix} .$$

The mean total arrival rate into each workstation, λ_i for $i = 1,2,3$, are determined from the following system of equations (Property 5.7):

$$\begin{aligned} \lambda_1 &= 10 + \lambda_2/3 + \lambda_3/2 \\ \lambda_2 &= \lambda_1/4 + \lambda_3/2 \\ \lambda_3 &= 3\lambda_1/4 + \lambda_2/3 \, . \end{aligned}$$

The solution to this system is $\boldsymbol{\lambda} = (40/\text{hr}, 30/\text{hr}, 40/\text{hr})$. Using these inflow rates, the utilizations for the three workstations are determined from

$$\begin{aligned} u_1 &= \frac{40}{2} \times 0.04 = 0.800 \\ u_2 &= 30 \times 0.0286 = 0.858 \\ u_3 &= \frac{40}{3} \times 0.0625 = 0.833 \, . \end{aligned}$$

The squared coefficient of variations of the arrival streams into the three workstations are also determined by solving a system of linear equations. These equations are considerably more complex to develop than the arrival rate equations since they are a combination of the departure SCV's for each workstation, and the branch and merging mechanisms for network traffic streams as in Property 5.8. Now we will need to make further modifications to Properties 5.8 and 5.9 due to batching.

Because of reentrant flows, Workstations 1 and 2, those stations could be modeled using the approach taken in Sect. 7.4.2 using the random batch size methodology as in Property 7.8; however, this will lead to a relatively complex system of equations, and in fact, most of the batch sizes will be of size one. Therefore, we will take the approach of using the *i.i.d.* departure stream SCV approximation of Eq. (7.24) for individual departures. Note that (7.24) is in terms of departures from Workstation 1 going to Workstation 2. Before using this equation in our system to define the arrival stream SCV's, we must rewrite (7.24) in terms of the arrivals to Workstation 1 (namely, we use the standard relationship given in Property 5.2 adjusted for batch arrivals). This yields the following equation for the SCV of the inter-arrival times to Workstation 2 following a batch operation of size k in Workstation 1

$$C^2_{a,2}(I) = \frac{kp\left[(1-q^k)\left((1-u_1^2)(C^2_{a,1}(I)/k)+u^2C^2_{s,1}(B)\right)+q^k+1\right]}{1-q^k} - 1\,, \quad (7.25)$$

where the subscript indicates the appropriate workstation.

The modification to the equation of Property 5.8 that is necessary is the inclusion of (7.25) whenever the subscript of the summation refers to the third workstation. The resulting system of equations to be solved to obtain $C^2_a(i)$ for $i = 1,2,3$, are the following:

$$\begin{aligned}
C^2_a(1) &= \frac{10}{40}(1) + \frac{30/3}{40}\left[\frac{1}{3}\left\{(1-0.858^2)C^2_a(2)+0.858^2(1)\right\}+\frac{2}{3}\right] \\
&\quad + \frac{40/2}{40}\left[\frac{1.5\left[(7/8)\left((1-0.833^2)C^2_a(3)/3+0.833^2(1)\right)+9/8\right]}{7/8}-1\right] \\
C^2_a(2) &= \frac{40/4}{30}\left[\frac{1}{4}\left\{(1-0.8^2)C^2_a(1)+0.8^2\frac{1+\sqrt{2}-1}{\sqrt{2}}\right\}+\frac{3}{4}\right] \\
&\quad + \frac{40/2}{30}\left[\frac{1.5\left[(7/8)\left((1-0.833^2)C^2_a(3)/3+0.833^2(1)\right)+9/8\right]}{7/8}-1\right] \\
C^2_a(3) &= \frac{40(3/4)}{40}\left[\frac{3}{4}\left\{(1-0.8^2)C^2_a(1)+0.8^2\frac{1+\sqrt{2}-1}{\sqrt{2}}\right\}+\frac{1}{4}\right] \\
&\quad + \frac{30/3}{40}\left[\frac{1}{3}\left\{(1-0.858^2)C^2_a(2)+0.858^2(1)\right\}+\frac{2}{3}\right]\,.
\end{aligned}$$

The solution to this system of equations can be solved directly as a system of linear equations. Or, if a matrix inverse routine is not available, an iterative procedure can be used where all the C_a^2's are initialized to 1 and the above equations are used to obtain an updated estimate. This process, using the updated estimates, is repeated several times until the C_a^2 values do not change to whatever degree of accuracy that you deem necessary. This iterative process converges to the unique solution to this system of linear equations. Using this iterative process, the solution to three decimals places repeats itself after the fifth iteration. Thus, the sixth iteration yields the solution

$$C_a^2(1) = 1.589, \quad C_a^2(2) = 1.780, \quad C_a^2(3) = 1.137\ .$$

The workstation performance measures of cycle time, $CT(i)$, and $WIP(i)$ can now be estimated. These are

$$\begin{aligned} CT(1) &= \frac{(1.589+1)}{2}\left(\frac{2.4}{60}\right)\frac{0.8^{\sqrt{2(3)}-1}}{2(1-0.8)} + \frac{2.4}{60} = 0.134\ \text{hr} \\ WIP(1) &= 0.134(40) = 5.347 \end{aligned}$$

$$\begin{aligned} CT(2) &= \frac{(1.780+1)}{2}\left(\frac{1.714}{60}\right)\frac{0.858}{1-0.858} + \frac{1.714}{60} = 0.267\ \text{hr} \\ WIP(2) &= 0.267(30) = 8.006 \end{aligned}$$

$$\begin{aligned} CT(3) &= \frac{(3-1)}{2}\frac{1}{40} + \frac{(1.137/3+1)}{2}\left(\frac{3.75}{60}\right)\frac{0.833}{1-0.833} + \frac{3.75}{60} = 0.303\ \text{hr} \\ WIP(3) &= 0.303(40) = 12.118\ . \end{aligned}$$

Note that Workstation 3 has the batch forming time included in the cycle time.

The total system performance measures are 10 jobs per hour for throughput (what comes in must go out in steady-state), a total work-in-process of

$$WIP_s = WIP(1) + WIP(2) + WIP(3) = 25.472\ \text{jobs},$$

and a mean cycle time per job of 25.472/10 = 2.547 hr (using Little's Law).

- *Suggestion: Do Problems 7.10, 7.11(b) and 7.12.*

Bibliographical Note

The batch move (Sect. 7.1) and setup batch (Sect. 7.2) cycle time models follow the development of Hopp and Spearman [5]. The random batch arrival and unit service model for $M/G/1$ systems is developed in Cooper [2], and the generalization for the $G/G/1$ case developed herein agrees with his result for Poisson arrivals. The renewal process approximations for the departure SCV's from the various batch service processes are developed in Curry and Deuermeyer [3]. The development approach is an extension of the $G/G/1$ departure process analysis of Buzacott and Shanthikumar [6]. A more general batching rule is contained in [4] where instead of using a fixed batch size, a minimum size and a maximum size are established so that processing would begin whenever the minimum size is available but if more that the maximum number of items are queued at the end of an operation, only the maximum would be allowed in the processor.

Problems

7.1. Consider a system with a single workstation that processes jobs one at a time. Jobs arrive to the factory at a rate of one per hour. An analysis of the arrival data indicates that these inter-arrival times have a squared coefficient of variation (SCV) of 1.5. The service time mean is 0.75 hours with an SCV of 2. The company policy is to work on orders k at a time. That is, orders are held until there are k jobs, then this group of jobs is released into the factory for processing. Since there is no physical reason for holding the incoming work and forcing it into groups, what is the impact on cycle time of this ”batching” operation for specified k values?
(a) $k = 2$.
(b) $k = 3$.
(c) $k = 4$.
(d) $k = 5$.

7.2. Consider a factory that has a single workstation that processes parts individually. These parts are quite heavy and the company policy is to palletize incoming parts into groups of k items for ease of transportation. These batches are then released into the factory for processing. These k items are processed at the machine and again placed back on the pallet. When the pallet is full, the k items have been processed, the pallet is transported to shipping.
(a) Neglecting the actual transportation time, what is the equation for cycle time of individual parts for this factory. This cycle time includes the waiting time for all batching operations. Compare the batch movement cycle time with that of a system that does not need to batch these items for movement within the factory. How much extra time does an average item incur due to batching for movement purposes?
(b) Assume it takes an average of t_1 to move a pallet from the unloading dock to the workstation and an average of t_2 to move a pallet from the machine to the next workstation. Assuming no waiting for a forklift to move the pallet, add the transportation time to the model.

7.3. Consider a factory that processes a single job type. Orders are processed one at a time in a serial processing configuration. One of the workstations requires a machine cleaning operation periodically. This workstation has only one machine. No more than 9 jobs can be processed between the cleaning of this machine. To insure proper cleaning of the machine, management has jobs batched at the machine in the specified group size, and then the operator cleans the machine before processing each batch. Jobs arrive at this workstation at a rate of four per hour. An analysis of the arrival data indicates that these inter-arrival times have a squared coefficient of variation (SCV) of 1.5. The service time for individual-unit processing has a mean of 0.15 hours with an SCV of 0.75. The cleaning operation takes a mean time of one-quarter of an hour with an SCV of 1.5. Management would like to know the impact on cycle time and the departure SCV for this machine (workstation) for the various batch sizes that are feasible between 1 and 9.

7.4. Consider a workstation with a batch server of capacity 4. Jobs arrive at the workstation individually at a rate of 6 jobs/hour and an inter-arrival time $C_a^2(I)$ of 3. Only full batches are processed at the workstation. The batch mean service time is 0.6 hours with $C_s^2(B) = 0.8$. Find:
(a) The cycle time for this workstation including the batch forming time.
(b) The expected number of batches waiting to be processed.
(c) What is the mean and $C_d^2(B)$ of the batch inter-departure times?
(d) Considering that the batch is immediately broken into individual jobs on completion of service, what are the mean and $C_d^2(I)$ of the individual inter-departure times?

7.5. Consider the output from an oven batch server with batch size k and $C_d^2(B)$. The batch is immediately broken into the individual items. Compute the missing items in the following table (using Property 7.7): the individual item's departure SCV, $C_d^2(I)$, and the individual item's arrival SCV, $C_a^2(I)$ at the next workstation after random branching of individuals with probability p.

k	$C_d^2(B)$	$C_d^2(I)$	p	$C_a^2(I)$
4	0.8		1/4	
4	0.8		1/2	
4	0.8		3/4	
5	2.0		1/4	
5	2.0		1/2	
5	2.0		3/4	

7.6. An oven-type processing workstation processes two products. The products are processed separately, not mixed, in the oven. The oven holds 5 units of both products. The products enter the workstation as individual units (not batches) and leave the workstation as individual units.

Product	λ_i	$C_a^2(i)$	$E[S_i]$	$C_s^2(i)$	batch size
1	5	1.5	0.50	Poisson	5
2	4	2.2	0.35	Erlang-3	5

(a) Compute the workstation average cycle time, $CT_s(avg)$.
(b) Compute the cycle time for Product 1, $CT(1)$.

(c) Compute the cycle time for Product 2, $CT(2)$.
(d) Compute the $C_d^2(B)$ for outgoing batches.
(e) Compute the $C_d^2(I)$ for outgoing individual jobs.

7.7. Consider a workstation that processes 4 jobs simultaneously $(k = 4)$. The departure stream from this workstation has a mean rate of 4 batches/hour with an SCV of 1.5. After leaving the workstation, individuals are randomly branched to other workstations for further processing. Two-thirds of the units are branched to Workstation Q, as its only arrival stream, that has service time characteristics (for individuals) of $E[T_s(I)] = 0.08$ and $C^2[T_s(I)] = 1.3$. Determine the expected cycle time for Workstation Q.

7.8. Consider a workstation that processes 5 units simultaneously $(k = 5)$. The departure stream from this workstation has a mean rate of 4 batches/hour with an SCV of 1.75. After leaving the workstation, individuals are randomly branched to other workstations for further processing. Sixty percent of the units are branched to Workstation G, as its only arrival stream, that has service time characteristics (for individuals) of $E[T_s(I)] = 0.07$ and $C^2[T_s(I)] = 1.7$. Determine the expected cycle time for Workstation G.

7.9. Reconsider the batch service network example illustrated in Fig. 7.3. Reanalyze this network with the following data rather than the data used in the example. The network structure is identical to the example, but all of the numerical data have been changed, including the branching probabilities. Obtain the system throughput, cycle time and work-in-process. The problem data by workstation follows.

Batch forming and external arrival data:

$$\gamma_1 = 6,\ C_{a0}^2(1) = 1,\ k = 4.$$

Workstation 1 data (setup batching):

$$\begin{aligned}
E[T_{si}(I)] &= 1/15, \\
V[T_{si}(I)] &= 1/10, \\
E[R] &= 1/4, \\
V[R] &= 4/10. \\
p &= 1/4, \\
1 - p &= 3/4\ .
\end{aligned}$$

Workstation 2 data (oven batching):

$$\begin{aligned}
E[T_s(B)] &= 0.8, \\
C^2[T_s(B)] &= 3/4,
\end{aligned}$$

$$q = 1/3,$$
$$1 - q = 2/3\,.$$

Workstation 3 (batch move):

$$E[T_s(I)] = 1/5,$$
$$C^2[T_s(I)] = 1/2\,.$$

7.10. Consider a factory with a batch server. Let the network structure be the same as that of Fig. 7.4, except that the branching probabilities are different. The data for the workstations and the branching probabilities are given below. Develop the workstation and system performance measures of throughput, cycle time and work-in-process. The external arrival process is assumed to be Poisson. Note: use (7.25) in the $C_a^2(I)$ term.

Workstation	Inflow	Machines	Batches	$E[T_s]$	$C^2[T_s]$
1	10	2	1	1/25	2
2	0	1	1	1/35	2
3	0	1	4	1/10	2

From/To	1	2	3
1	0	1/3	2/3
2	1/3	0	1/3
3	4/10	6/10	0

7.11. Consider a network of four workstations with the data given in the following tables. Draw the network diagram and develop the workstation and system performance measures of throughput, cycle time and work-in-process. The external arrival process is assumed to be Poisson.
(a) Use Property 7.7 in the $C_a^2(I)$ computations.
(b) Use Eq. (7.25) in the $C_a^2(I)$ computations.

Workstation	Inflow	Machines	Batch Size	$E[T_s]$	$C^2[T_s]$
1	2.5	1	1	0.10	2.00
2	1	2	1	0.26	1.50
3	0	1	1	0.13	0.75
4	0	1	4	0.64	3.00

From/To	1	2	3	4
1	0	2/4	1/4	1/4
2	1/3	0	1/3	1/3
3	4/8	1/8	0	1/8
4	0	0	2/3	0

7.12. A manufacturer produces two products in a three-workstation facility. The products are similar and both use an identical heat-treatment process. Thus, these products can be indiscriminately mixed for this oven process, that can process six items simultaneously. Both external arrival processes are Poisson distributed. The factory capacity consists of 5 identical machines in Workstation 1, 4 identical machines in Workstation 2, and one oven for the heat-treatment process in Workstation 3 (with a batch capacity of 6 jobs). Using the product branching probabilities and processing time data listed below, compute the factory cycle time, work-in-process and throughput. All times are in hours. Note: use Eq. (7.25) in the $C_a^2(I)$ computations.

		Product 1		
Workstation	Inflow	Batch Size	$E[T_s]$	$C^2[T_s]$
1	2	1	0.20	1.00
2	0	1	0.30	1.50
3	0	6	0.40	1.75

	Product 1		
From/To	1	2	3
1	0	1/2	1/2
2	1/4	0	3/4
3	1/3	1/3	0

		Product 2		
Workstation	Inflow	Batch Size	$E[T_s]$	$C^2[T_s]$
1	3	1	0.30	2.00
2	0	1	0.35	1.80
3	0	6	0.40	1.75

	Product 2		
From/To	1	2	3
1	0	2/3	1/3
2	1/3	0	2/3
3	3/5	0	0

7.13. Re-derive the batch service time process characteristics $C^2[T_s(B)]$ (Eq. 7.18) using Property 1.9 for the sum of random variables.

7.14. Team Project Problem. The Southwestern Specialties Company has a line of four products that they produce in their factory located in Houston, Texas, working 24 hours per day. The company is soliciting bids from consulting firms for the analysis of their current and future factory performances. The company currently has contracts with several national retail companies, such as Wal-Mart, Kmart, and Target, to produce specific quantities of each of their four products. The initial project phase is to develop a model of their current factory and develop cycle time estimates for each product. The second phase of the project will be to predict the impact of a new marketing strategy based on E-Commerce using the World-Wide-Web. Several consulting companies have been selected to perform the first phase of the project (current factory performance modeling) and the best among those will be selected for the future phase. Only after successfully demonstrating your consulting firms capabilities, will the company authorize the release to the consulting firm the nature of the second phase of the modeling and analysis project.

First Phase Information

The Southwestern Specialties Company's Houston factory consists of three workstations (called Workstations 1, 2 and 3). Workstation 3 is an oven heat-treatment facility. The four products take different routes through the three workstations and have different numbers of processing steps. There currently are three machines in Workstation 1 and one machine each in Workstations 2 and 3. The machine (oven) in Workstation 3 has the capacity to process up to 4 units simultaneously, but it is currently operated with a fixed batch size of 3 units. Engineering has spent considerable design and analysis time over the years to develop a processing procedure that allows all four of the products to be processed in the oven with the same time and temperature settings. Therefore, the factory operations personnel can form an oven batch from any combination of the four product types.

Orders are released to the factory according to a Poisson process at a mean rate of 7.68 orders per day. The current distribution of order releases by product type is (20%, 30%, 25%, 25%) for Products, 1, 2, 3 and 4, respectively.

Engineering has developed standard times for each of the processing steps for each product and these "mean" times are listed below. Their analysis has revealed the surprising fact that the distribution of processing times for each and every process is very accurately approximated by an Erlang Type-2 distribution. The workstations' sequence for each product is:

Products	1	2	3	4	5
1	3	1	2	1	
2	1	2	3	2	1
3	2	1	2	3	1
4	3	1	2		

The mean processing time by product and processing step, in hours, are:

Products	1	2	3	4	5
1	8	6	1.7	6	
2	5	1.6	8	1.5	5
3	1.9	4	2.2	8	4
4	8	3	2.2		

The average cycle time for all products is approximately 80 hours. The consulting firms will be selected to continue into the second modeling and analysis phase based on their answer to the question: What is the mean cycle time by product? Of course, all relevant data concerning your firms answer to this question must be provided.

Second Phase Information

Your consulting firm has been selected to analyze the new company strategy for the Southwestern Specialties Company. The company has decided to no longer use

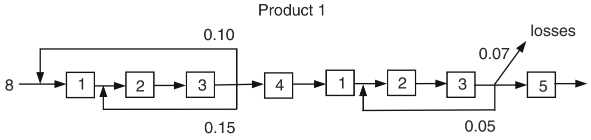


Fig. 7.5a Process flow diagram for Product 1 of Problem 7.15

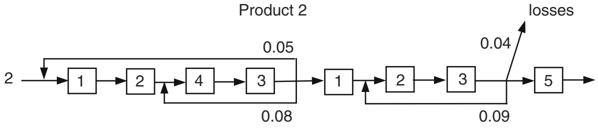


Fig. 7.5b Process flow diagram for Product 2 of Problem 7.15

fixed contracts. They have decided that they would do better by selling the four products over the internet. The research staff has determined that the sales rates are functions of the product cycle times and they estimate these functions as:

Product 1: $r(1) = 0.1088 - 0.0006 \times CT(1)$,
Product 2: $r(2) = 0.1632 - 0.0007 \times CT(2)$,
Product 3: $r(3) = 0.1360 - 0.0006 \times CT(3)$,
Product 4: $r(4) = 0.1360 - 0.0009 \times CT(4)$.

The company will allow *two* new machines to be purchased of any type (excluding ovens). The company wants answers to the following questions:
(a) What should be the company capacity structure?
(b) What is the projected company sales rates and cycle times for the four products?
(c) Are the sales consistent and stable? If not what can be done to make them stable?
(d) Is the company in a good or bad situation?

7.15. Team Project Problem. Quality Products Inc., a company that manufactures high-quality heat-pumps for the housing industry, has a local manufacturing facility. This plant only produces heat-pumps and until recently there was only one basic production process for these items. With the new environmental concerns and government regulations, they have designed and recently brought into production a second product line of heat-pumps. The production processes for the old and new product lines are similar, using the same equipment, but they have slightly different processing sequences and processing times. The product processing sequences are illustrated in Figs. 7.5a and 7.5b. Product 1 is the old heat-pump process and Product 2 is the production sequence for the new line of heat-pumps. This first quarter of 2002 the daily releases of products to be manufactured is 8 units on the product 1 and 2 units of Product 2. The average cycle time for the facility is in the neighborhood of 6.2 days.

Quality Products Inc. would like to have a consulting team perform a systems analysis for each quarter of the year. They expect that the total of units manufactured to remain at a demand level of 10 per day but the product mix will change each quarter. Their quarterly demand forecasts for daily demands by product type are:

	First Quarter	Second Quarter	Third Quarter	Fourth Quarter
Product 1	8	6	4	2
Product 2	2	4	6	8

The company is concerned about their machining capacities as demands change over time. They would also like to know what the impacts will be on their cycle times and they want to estimate the cycle times for the individual products as well as the facility average. If new machines are needed, they want to get these ordered and installed so that they will not suffer a short-fall in production output versus demand.

The top-planning engineer for Quality Products Inc. has developed the data for the old and new product processing steps. The mean processing times (in days) are:

Means	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Product 1	0.008	0.120	0.070	0.070	0.075	0.100	0.070	0.180
Product 2	0.002	0.100	0.090	0.070	0.080	0.080	0.070	0.100

SCV's	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Product 1	1.5	1.2	1.3	1.0	1.6	0.9	1.3	0.5
Product 2	1.0	1.0	1.0	1.3	1.0	1.0	1.3	1.0

There are currently three machines in Workstation 2 and two machines at Workstation 5. The other workstations seem to be operating okay with a single processing machine. Workstation 3 is a heat-treatment process and the current capacity of the machine is two jobs at a time.

The product demand data is currently 8 units of Product 1 per day with a SCV of 1.5. Product 2 has a mean demand rate of 2 units per day with an SCV of 0.75. As demand shifts from being predominantly Product 1 to mostly Product 2, the company does not anticipate a change in the SCV's for the individual products.

All of the machines in the factory have a 95% availability factor. The mean repair times are, respectively, 0.2, 0.3, 0.4, 0.35, 0.5, days. All repair times are exponentially distributed.

7.16. Team Project Problem. The MicroTex Corporation makes special purpose microprocessors that are used in a variety of machines. The company produces two products as variants from the same processing procedure. The products are distinguished after one layer or single sequence through the processing steps. After the first layer has been completed, the wafers go through a test operation; wafers are characterized as worthless (waste), bad and in need of rework, good wafers but low cycle speeds, and excellent with high cycle speeds. Product 1 is made from the low cycle speed processors that are immediately packaged and shipped. The high-speed units are processed further by a second sequence through the basic operational steps (using the same machines as previously) and then a final test is performed. Units again are characterized as waste, rework and completed units (no low speed units can come out of the second test). The completed units are then packaged and shipped as the company's high-grade product.

Microprocessor chips are produced by a process that starts with pure silicon wafers that are fragile, flat, thin circular objects that look similar to glass. Patterns are placed on the wafers by covering them with a photo-resist material and then exposing the images onto the resist by shining light through a template or mast of the desired image. The images are hardened by baking the wafer in an oven. A pattern of holes is then etched into the mask layer by removing the exposed material. This allows the dopants to be diffused into selected areas of the wafer. Specific ion atoms (dopants) are implanted on the exposed surface (boron, phosphorous, and arsenic) by diffusion processes. These processes are repeated hundreds of times to produce a state of the art microprocessor. Then a wafer probe is used to functionally test the individual processors on the wafer and characterize their performance potential. The completed wafers are diced into single chips with a diamond saw and then attached via glue to a package. The package provides the contact leads to the chip. Wire bonding, generally with gold leads, is used to connect the package leads to the wire connections within the chip. Then the package containing the chip is encapsulated with a plastic coating for mechanical and environmental protection.

The MicroTex wafer fab is a state-of-the-art pilot facility with the latest cluster tool technology. These fabrication processes are performed in three basic steps. The first set of processing steps is performed in Workstation 1 with a pair of identical single wafer processing equipment (machines). At the second workstation, the remaining set of operations is performed in a batch mode using a single large capacity (up to eight lots simultaneously) processor (similar to an oven operation). A third workstation contains a single testing machine used to determine the wafer performance characteristics. Workstation 4 consists of the packaging operations also performed using a single piece of equipment. All of the completed products received the same general processing using the same equipment, although the processing times vary for the second production pass. The wafer units that have graded out as high quality and speed are processed further.

To prepare for the second layer, the second time through the processing steps, a separate distinct set of processing must be performed. This preparation processing step is distinct from previous processing and, therefore, these operations are performed on a separate machine in Workstation 5. When completed, these units are sent back through the first three workstations for another sequence of processes. The second sequence of processing has distinct times from those of the first sequence, except for the batch operations of Workstation 2. This is fortunate, allowing batching at Workstation 2 to be indiscriminate of the type of wafers being processed. That is, batches can consist of either or both types of wafers. If the high quality product wafers grade out acceptably they are also then packaged at Workstation 4 and shipped.

Phase I

MicroTex management would like your consulting team to develop a model of their facility and help them answer questions concerning potential areas of improvement. The first phase of the project is to utilize their best guess data, compiled by their lead engineers, and develop a preliminary model of their company. If they find this result acceptable, they will allow your team access to the actual company proprietary data from which accurate and meaningful data can be developed. This refined model will then be used to develop strategies for future company improvement and new product development. These types of facility are extremely expensive to build, frequently costing from one to two billion dollars for a full scale facility. Thus, continuous operation of the facilities is maintained at all times; 24 hours a day, seven days a week.

The work release rate for the pilot facility is one job (lot of 24 wafers) per hour. All times are given in lot units. The mean times for the first three processing steps are estimated to be 1.15, 2 and 0.25 hours, respectively. The test operation on average finds that 10% of the processed wafers are scrapped and 15% can be reworked and are thus reprocessed at Workstation 2. Of the acceptable units, only 1/3 grade out as high quality and speed and go through further processing. The packaging operation for the low or first level product takes 45 minutes while the high quality product takes 54 minutes. For the high quality product, the unique first additional step takes 2 hours and 15 minutes. The company policy is that when (on the first trip through only) a lot is scrapped, it is replaced with a new lot start. The second trip through workstation one takes 75 minutes and in Workstation 3 the second trip requires an additional 3 minutes over the first processing time. The batch size used in Workstation 2 is a fixed quantity of four jobs (four lots of wafers). This is a carryover from a previous production line where the machine capacity was limited to four lots. For the pilot factory, the number of machines in Workstation 1 is two, the number of ovens in Workstation 2 is one, and the number of machines in Workstation 3 is one. For this pilot system analysis, we can assume that the order release process (external arrival process) and all service processes have squared coefficients of variation with values of 1.

MicroTex wants a short written report of your consulting team's preliminary model to determine if your consulting team will be continued into the actual factory analysis phase.

Phase II

The MicroTex Corporation accepts your design team as the company's consulting team for the wafer fabrication pilot facility study. During your preliminary analysis period, the company has had a team of industrial engineering coop students collecting time study data for all the machines used in the facility. The coop group finished the analysis on four of the five machine types and furnished the following table:

Processing Step	$E[S]$	C_s^2
1	1.15 hr	3
2 (oven)	2 hr	2
3 (test)	0.25 hr	1
4 (package)	0.75 hr	4
5 (special)	2.25 hr	3.05
6 (1)	1.25 hr	3
7 (oven)	2 hr	2
8 (test)	0.30 hr	1
9 (packaging)	0.90 hr	4

The current pilot facility has a cycle time around 65-70 hours. Management has a quality improvement program in place and they predict that scrap losses can be reduced to 5%, rework can be reduced to 10%. In addition, engineering believes that the processing times variations can be reduced across the board by 50%. Can the management goal of a cycle time of less than 35 hours be reached?

Engineering is always working to improve the high speed wafer yield percentage. These units are worth considerably more and have an unlimited market. Engineering feels that this yield percentage can be drastically improved, but maintaining the cycle time goal of 35 hours will be impossible. Management has, therefore, agreed to allow one more machine (of any type) to be placed in the pilot facility if necessary. What is the maximum high speed wafer yield percentage that can be accommodated within the 35 hour cycle time guideline?

The real goal of the pilot facility is to determine what facility configuration is necessary for a full scale facility with a release rate of 10 lots per hour. We can assume that all of the learning with respect to yields, variation reductions, etc., carry over to the new facility. Assuming that the best yield results for the pilot facility can be maintained in the new plant, what is the machine configuration and estimated cycle time for this facility?

References

1. Albin, S.L, and Kai, S. (1986). Approximation for the Departure Process of a Queue in a Network. *Naval Research Logistics Quarterly*, **33**:129–143.
2. Cooper, R.B. (1990). *Introduction to Queueing Theory*, Third Edition. The MacMillan Company, New York.
3. Curry, G.L., and Deuermeyer B.L. (2002). Renewal approximations of the departure processes of batch systems. *IIE Transactions*, **34**:95–104.
4. Curry, G.L., and Feldman, R.M. (1985). An M/M/1 Queue with a General Bulk Service Rule. *Naval Research Logistics Quarterly*, **32**:595–603.
5. Hopp, W.J., and Spearman, M.L. (1996). *Factory Physics: Foundations of Manufacturing Management*. Irwin, Chicago.
6. Buzacott, J.A., and Shanthikumar, G.J. (1993). *Stochastic Models of Manufacturing Systems*. Prentice-Hall, Englewood Cliffs, New Jersey.

Chapter 8
WIP Limiting Control Strategies

Many companies find themselves with too much work-in-process. The disadvantages of high levels of *WIP* are numerous, and many of the disadvantages cannot be directly measured economically. Two major disadvantages of high *WIP* levels that are difficult to economically evaluate are not being able to respond to demand changes quickly and the potential to build a considerable quantity of poor quality stock before realizing that there is a quality problem. To help control inventory within production and manufacturing facilities, *WIP* limiting production procedures are frequently used.

The just-in-time production approach attempts to control product releases based on factory conditions. The production release approach studied to this point is based on a schedule or "push" approach. The "pull" production strategy, generally associated with the Toyota production controls, was originally based on a card or kanban system. (The term kanban, borrowed from the Japanese language, originally referred to the use of cards to control the movement of parts; however, today many things other than cards might be used including simply an empty cart or even golf balls.) The general concept is to release work only when something has left the system (area or range of oversight control). This approach is thus some form of a *WIP* limiting process. One of the simplest approaches is to limit the total *WIP* and not be concerned with allocation restrictions within the facility itself. This approach, popularized under the term CONWIP from CONstant WIP (see Hopp and Spearman [8]), puts a new job into the system whenever a job leaves the system after the system has reached its CONWIP level. Once the system is loaded to the desired limit, this approach maintains a constant *WIP* in the system. More detailed controls can be accomplished by restricting the *WIP* available in regions of the factory and even down to the workstation level. When the CONWIP approach is implemented for each workstation, it becomes conceptually equivalent to the kanban approach.

Two different approaches for facility control are studied in this textbook: CONWIP and kanban control policies. This chapter considers a total *WIP* limit approach via the mathematical methodology of closed queueing networks. In Chap. 9, *WIP* limits at individual workstations or kanban control are studied. These two approaches lead to different analytical models for predicting the system performance

G.L. Curry, R.M. Feldman, *Manufacturing Systems Modeling and Analysis*, 2nd ed.,
DOI 10.1007/978-3-642-16618-1_8, © Springer-Verlag Berlin Heidelberg 2011

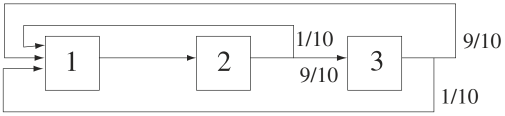


Fig. 8.1 Closed queueing network example

measures. The CONWIP method is the simpler mathematically as well as the simpler to implement and can be studied by the well developed approximation area called mean-value analysis for closed queueing networks. Mean value analysis is a computationally simple approach that was developed for exponential service time models. An approximation for general service distributions called extended mean value analysis is also discussed.

This chapter is concluded with a case study of the impacts of CONWIP control along with several job sequencing algorithms for selecting the next job from the queue for processing. This study is presented to familiarize the reader with the potential impact that scheduling rules, other than just push or pull strategies, can have on factory performance.

8.1 Closed Queueing Networks for Single Products

In Chap. 5, queueing networks were used to represent a factory. These were open queueing networks because jobs arrived from a source external to the network and jobs departed from the network. We will now change this approach and use closed queueing networks to model the factory.

Definition 8.1. A *closed queueing network* is a network of queues in which no arrivals are possible from outside the network and no jobs within the network can leave.

The network displayed in Fig. 8.1 is an example of a closed queueing network. A closed queueing network is a representation of a constant *WIP* controlled system where the total *WIP* is set at a specified limit, say $w_{\max}$. When a job completes service, it is counted and a new job is entered into the system immediately. This is mathematically equivalent to branching the completed job back to the starting workstation. For this representation of a constant *WIP* system, there are no external flows into the system and really no exiting flows from the system. Job completions are counted by recognizing that the rate of "good" jobs leaving the last workstation is equivalent to a job completion. The term "good" implies that a proportion of the jobs leaving the last workstation could be defectives that are not counted as completed jobs and these may be branched back for rework or, if scrapped, then a totally new job is started in the defective job's place.

Example 8.1. Consider a three-workstation factory where all jobs leaving Workstation 1 are sent to Workstation 2. From Workstation 2, 10% must be reworked and are returned to Workstation 1 and 90% are sent to Workstation 3. From Workstation 3, 10% are defective and are again sent back to Workstation 1 and 90% are good and shipped to the customer. A control will be placed on this factory so that there will always be exactly 25 jobs in the system. To implement this policy, a job will be started whenever a finished job is shipped to a customer. In such a situation, jobs are actually flowing into the system and out of the system, but mathematically, it is equivalent to the closed system shown in Fig. 8.1. The throughput rate of this system is the flow rate of jobs along the upper path (the path indicating a 9/10 probability branch) leaving Workstation 3 and returning to Workstation 1. □

The mathematical analysis of a closed queueing network starts with solving for the flows between workstations. It is assumed throughout the chapter that there are n workstations. A slight problem exists for closed-queueing networks in that there is no longer a unique set of flow rates that describe the system. This is not surprising if one considers that the flow rates are dependent on the number of jobs allowed in the system, $w_{\max}$. To illustrate this point, again consider Fig. 8.1. The arrival rates to each workstation must satisfy the following

$$\begin{aligned}
\lambda_1 &= 0.1\lambda_2 + \lambda_3 \\
\lambda_2 &= \lambda_1 \\
\lambda_3 &= 0.9\lambda_2
\end{aligned}$$

It is now easy to verify that the solution

$$(\lambda_1, \lambda_2, \lambda_3) = (1, 1, 0.9)$$

satisfies the flow requirements of Fig. 8.1. But it is also true that

$$(\lambda_1, \lambda_2, \lambda_3) = (2, 2, 1.8)$$

also satisfies the flow requirements. In fact, any multiple of the vector (1,1,0.9) would satisfy the above equation for the three rates, so obviously a unique set of flow rates cannot be found. But what can be found are the relative flow rates, call these the vector (r_1, r_2, r_3), that give the rates with respect to each other. For the above example, these rates are (1, 1, 0.9), based on the flow for either of the first two workstations. These relative rates are (1/0.9, 1/0.9, 1) if they are computed relative to the flow of the third workstation.

Before developing a method to obtain the relative flow rates, consider the difficulty in attempting to perform the standard flow rate analysis (see Sect. 5.4.1). In general, a solution to the following system of equations is required

$$\boldsymbol{\lambda} = P^T \boldsymbol{\lambda} + \boldsymbol{\gamma},$$

where $\boldsymbol{\lambda}$ is the vector of unknown internal flow rates, $\boldsymbol{\gamma}$ is the vector of known rates of arrivals from an external source, and P is the routing matrix giving the branching probabilities. Since there are no external inflows for closed queueing networks (namely, $\boldsymbol{\gamma} = \mathbf{0}$), the system can be rewritten as

$$(I - P^T)\boldsymbol{\lambda} = \mathbf{0}\,,$$

where I is an identity matrix (see Property 5.7).

Now if the inverse of $(I - P^T)$ exists, the flow can only be zero ($\boldsymbol{\lambda} = \mathbf{0}$). By the illustration above, the flow rates are not zero, so one can conclude that $(I - P^T)^{-1}$ does *not* exist. In fact this is always true for a closed queueing network, i.e., the matrix $(I - P^T)$ is nonsingular. Therefore, a dependent system of equations results when the external flows are zero. For this situation, any one of the equations can be dropped from the system. (Technically, this is an eigenvector problem for a positive matrix whose row sums all add to one. The maximum eigenvalue is one and the solution is, therefore, unique up to a multiplicative constant. As long as no workstation or group of workstations is isolated from the other workstations, there will be exactly one redundant equation.) That is, any one of the relative flow rate factors r_i can be set to some positive constant and then the other flow rate factors can be obtained relative to this value. For example using Fig. 8.1, let $r_1 = 1$ and then solution is $(r_1, r_2, r_3) = (1, 1, 0.9)$; let $r_3 = 1$ and the relative flow rates are $(r_1, r_2, r_3) = (1/0.9, 1/0.9, 1)$.

With this dependency, the question arises as to the proper methodology for obtaining these relative flow rates. The above approach is used, where one of the r_i is set to a constant. Without loss of generality, r_1 will always be set to 1 and, thus, it is no longer necessary to include the unknown r_1 in the system of equations. To illustrate, notice that routing matrix for Fig. 8.1 is

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0.1 & 0 & 0.9 \\ 1 & 0 & 0 \end{bmatrix}.$$

Now, after eliminating the first equation and setting the first variable equal to 1, the system of equations becomes

$$\begin{aligned} r_1 &= 1 \\ \begin{pmatrix} r_2 \\ r_3 \end{pmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.9 & 0 \end{bmatrix} \begin{pmatrix} 1 \\ r_2 \\ r_3 \end{pmatrix} \end{aligned}$$

which can be written as

$$\begin{aligned} r_1 &= 1 \\ \begin{pmatrix} r_2 \\ r_3 \end{pmatrix} &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0.9 & 0 \end{bmatrix} \begin{pmatrix} r_2 \\ r_3 \end{pmatrix} \end{aligned}$$

This system becomes

$$r_1 = 1$$

$$\begin{pmatrix} r_2 \\ r_3 \end{pmatrix} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0.9 & 0 \end{bmatrix} \right)^{-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

yielding $(r_1, r_2, r_3) = (1, 1, 0.9)$.

This same logic can be extended to a general closed network resulting in the following general property.

Property 8.1. *Let P denote the routing matrix associated with a closed queueing network containing n workstations, and let Q denote the submatrix formed from P by deleting the first row and first column; that is, $q_{i,j} = p_{i+1,j+1}$ for $i, j = 1, \cdots, n-1$. Then the vector of relative arrival rates to each workstation, $\mathbf{r}$, is given by $r_1 = 1$ and*

$$\begin{pmatrix} r_2 \\ \vdots \\ r_n \end{pmatrix} = \left(I - Q^T\right)^{-1} \begin{pmatrix} p_{1,2} \\ \vdots \\ p_{1,n} \end{pmatrix}.$$

Notice that I is an identity matrix of dimension $n-1 \times n-1$ and the column vector on the right-hand side of the equation is the first *row* of the routing matrix minus the first element.

- *Suggestion: Do Problems 8.1 and 8.2.*

8.1.1 Analysis with Exponential Processing Times

In this section, a system of equations for determining the mean cycle time and the expected *WIP* in each workstation is developed under the assumption that all processing times are exponentially distributed. The approach used when all workstations have only one server and there is only one product within the factory serves as the building block for the more complicated cases; therefore we discuss the simplest case first and then extend those models to the other cases.

8.1.1.1 Single-Server Systems

The methodology used to determine mean cycle times within a closed network is called a *mean value analysis*. In the initial models of queueing systems, the approach was to obtain the probability distribution for the number of jobs within the system, and then from the number of jobs, the various measures were obtained. We now bypass the probabilities and determine the mean values directly. The results for the exponential cases are exact.

In order to use the same notation as in the previous chapters, the letter w will be used to denote the number of jobs within a system. Thus, we continue to use n for the total number of workstations and k as the workstation index. In later sections, i will denote the type and m will be the total number of types within the factory; however, for this section, $m = 1$ so we will not need to specify the job type. The idea behind the mean value analysis is that the mean values for a network with w jobs can be easily derived from a network with $w-1$ jobs; therefore, we will need to reference the various mean values by the number of jobs within the network. For example, $CT_k(w)$ will denote the mean cycle time at node k under the assumption that the closed network (factory) contains w jobs.

The first key relationship that is needed is the actual arrival rate into each workstation. From Property 8.1, we know the relative arrival rates. In other words, if $\lambda_k(w)$ is the actual arrival rate to Workstation k when there are w jobs within the network, then

$$\lambda_k(w) = x(w)\, r_k\,, \tag{8.1}$$

where $x(w)$ is some (unknown) value dependent on the number of jobs in the system. By Little's Law this leads to

$$WIP_k(w) = x(w)\, r_k\, CT_k(w)\,.$$

Because the total WIP must equal w, we sum over all workstations and solve for the unknown "arrival rate" constant

$$x(w) = \frac{w}{\sum_k r_k\, CT_k(w)}$$

which when combined with Eq. (8.1) leads to the following property.

Property 8.2. *Consider a closed network with n workstations containing w jobs with the relative arrival rates to the workstations given by the n-dimensioned vector* **r** *determined from Property 8.1. The arrival rate to Workstation k is*

$$\lambda_k(w) = \frac{w\, r_k}{\sum_{j=1}^{n} r_j\, CT_j(w)}\,.$$

The time spent within a workstation by a job equals the service time for that job plus the time all jobs in front of that job must spend on the server. The key concept that makes the mean value analysis possible was shown by Reiser and Lavenberg [11]; that is, the average number of jobs in front of an arriving job for a factory containing w jobs equals that workstation's WIP for a factory containing $w-1$ jobs. Thus, the following relationship holds for Workstation k:

$$CT_k(w) = E[T_s(k)] + E[T_s(k)]\,WIP_k(w-1)\ . \tag{8.2}$$

Notice that some care needs to be taken in interpreting the parameters correctly. Because the mean service time at a workstation does not depend on the number of jobs within the network, the parameter k within the expression $E[T_s(k)]$ refers to the workstation number. However, the parameter for the cycle time and WIP refer to the total number of jobs within the network. The first term in Eq. (8.2) is the time for processing the arriving job itself and the second term represents the processing time for all jobs within the workstation, including the job in service, at the arrival time.

Using Little's Law and Property 8.2, the $WIP_k(w-1)$ term in the above equation can be replaced by $CT_k(w-1)$ leading to the following Mean Value Analysis Algorithm.

Property 8.3. *Consider a closed network with n workstations containing $w_{\max}$ jobs. Each workstation has a single exponential server and the relative arrival rates to the workstations are given by the n-dimensioned vector* $\mathbf{r}$ *determined from Property 8.1. The following algorithm can be used to obtain the workstation mean cycle times.*

1. *Set $CT_k(1) = E[T_s(k)]$ for $k = 1, \cdots, n$ and set $w = 2$.*
2. *Determine $CT_k(w)$ for $k = 1, \cdots, n$ by*

$$CT_k(w) = E[T_s(k)] + E[T_s(k)]\,\frac{(w-1)\,r_k\,CT_k(w-1)}{\sum_{j=1}^{n} r_j\,CT_j(w-1)}\ .$$

3. *If $w = w_{\max}$, determine arrival rates from Property 8.2 and stop; otherwise, increment w by 1 and return to Step 2.*

Example 8.2. Consider the network given in Fig. 8.1. Let the mean processing times at the three workstations be 12 minutes, 30 minutes, and 30 minutes, respectively. In addition, 9/10 of the jobs leaving Workstation 3 are considered good and will be shipped to customers while 1/10 of the jobs are scrapped. Management has decided that the CONWIP level will be set to 5 jobs. The first step of the mean value analysis is to determine the relative arrival rates. As you recall, just before the statement of Property 8.1, we showed that

$$r_1 = 1/\text{hr}, \quad r_2 = 1/\text{hr}, \quad r_3 = 0.9/\text{hr}\ .$$

The steps of the algorithm yield Table 8.1 for the cycle time values at the various WIP levels, where all values are in hours. Once the cycle time values are obtained, the other standard workstation characteristics can be derived from Property 8.2 and Little's Law yielding Table 8.2.

Notice that each utilization factors is less than 1; this will always be the case because the service rate serves as the upper limit to the throughput for each workstation, and the utilization factor equals the arrival rate (throughput rate) times the

Table 8.1 Mean cycle time results (in hours) for Example 8.2

Iteration	$CT_1(w)$	$CT_2(w)$	$CT_3(w)$	$\sum r_j CT_j(w)$
$w=1$	0.200	0.500	0.500	1.150
$w=2$	0.235	0.717	0.696	1.578
$w=3$	0.260	0.955	0.897	2.021
$w=4$	0.277	1.208	1.099	2.475
$w=5$	0.290	1.477	1.299	2.936

Table 8.2 Workstation characteristics for Example 8.2 at a CONWIP level of 5

Measure	Workstation 1	Workstation 2	Workstation 3
$CT_k(5)$	0.290 hr	1.477 hr	1.299 hr
$\lambda_k(5)$	1.703/hr	1.703/hr	1.533/hr
$WIP_k(5)$	0.493	2.515	1.992
$u_k(5)$	0.341	0.852	0.767

service time. More generally, the utilization factor is the arrival rate times the mean service time divided by the number of machines at the workstation; that is,

$$u_k(w) = \lambda_k(w)\,E[T_s(k)]/c_k \,,$$

where c_k is the number of machines at Workstation k. Since 90% of the output from Workstation 3 is considered good, the throughput rate of this factory is 90% of the throughput rate of Workstation 3; thus, the mean number of jobs shipped from this factory is

$$th_s = 0.9 \times 1.533 = 1.38/\text{hr} \,.$$

Since the WIP level is always 5 for this example, the system mean cycle time is given by

$$CT_s = \frac{w_{\max}}{th_s} = \frac{5}{1.38} = 3.62 \text{ hr} \,.$$

□

- *Suggestion: Do Problems 8.3, 8.4, 8.5 (a,b,c), 8.6 (a,b), and 8.7 (a,b).*

8.1.1.2 Multi-Server Systems

The algorithm for the multi-server case will require evaluation of the marginal probabilities associated with each workstation and is called a Marginal Distribution Analysis. (The probabilities are marginal in that they refer to the number of jobs within a specific workstation and not the joint probability of the number of jobs in different workstations at the same time.) As long as the processing times at each workstation are exponential, the analysis will yield the correct mean values. It is very similar to the Mean Value Analysis in that the cycle time calculations for a network with a CONWIP level set to w individual jobs depends on the values calculated for a network with a CONWIP level of $w-1$ jobs; however, the major difference is that the marginal probabilities must be calculated first. These probabilities when there are w jobs within the network will be denoted by $p_k(j;w)$ for $j = 0, \cdots, w$ where the subscript k refers to the workstation number. Then the cycle time for w jobs in the network will be expressed in terms of the marginal probabilities for a network with $w-1$ jobs.

If Workstation k has c_k servers and each server has a mean service time of $E[T_s(k)]$, then the *rate* of service for the workstation with j jobs within it is equal to $\min\{j, c_k\}/E[T_s(k)]$. The key to obtaining an expression for the marginal probabilities is to observe that the rate of arrival into Workstation k containing j jobs with a total network population fixed at w jobs equals $\lambda_k(w)\, p_k(j-1;w-1)$ and the rate of leaving that node is $\min\{j, c_k\}\, p_k(j;w)/E[T_s(k)]$ (see [2, p. 373]). Equating these two probabilities using a similar approach to the rate balancing method of Sect. 3.2 yields an iterative expression for the probabilities and cycle times. The resulting Marginal Distribution Analysis Algorithm of the following property is similar to the algorithm contained in Buzacott and Shanthikumar [2, pp. 373–374].

Property 8.4. *Consider a closed network with n workstations containing $w_{\max}$ jobs. Workstation k has c_k servers with exponential processing time having a mean of $E[T_s(k)]$. The relative arrival rates to the workstations are given by the n-dimensioned vector* **r** *determined from Property 8.1. The following algorithm can be used to obtain the workstation mean cycle times.*

1. *Set $p_k(0;0) = 1$ for $k = 1, \cdots, n$, and set $w = 1$.*
2. *Determine $CT_k(w)$ for $k = 1, \cdots, n$ by*

$$CT_k(w) = E[T_s(k)] \sum_{j=0}^{w-1} \frac{j+1}{\min\{j+1, c_k\}}\, p_k(j;w-1) \,.$$

3. *Define the workstation arrival rates, for $k = 1, \cdots, n$, by*

$$\lambda_k(w) = \frac{w\, r_k}{\sum_{i=1}^{n} r_i CT_i(w)} \,.$$

   If $w = w_{\max}$, stop; otherwise, proceed to the next step.

4. *Determine $p_k(j;w)$ for $k = 1, \cdots, n$ and $j = 1, \cdots, w$ by*

$$p_k(j;w) = \frac{\lambda_k(w)\, E[T_s(k)]}{\min\{j, c_k\}}\, p_k(j-1;w-1) \,.$$

5. *Set $p_k(0;w) = 1 - \sum_{j=1}^{w} p_k(j;w)$ for $k = 1, \cdots, n$.*
6. *Increment w by 1 and return to Step 2.*

Since the mean cycle time of a workstation does not require the marginal probabilities of other workstations, the two algorithms can be used together. In other words, if some workstations have only one server, the easier algorithm of Property 8.3 can be used for those workstations while the algorithm of Property 8.4 is used for those workstations with multi-servers.

Example 8.3. We shall increase the capacity of the factory used in Example 8.2 (from Fig. 8.1) by adding two machines to Workstation 2 and adding one machine to Workstation 3; thus, we have $c_1 = 1$, $c_2 = 3$, and $c_3 = 2$. All other characteristics will stay the same. Although Workstation 1 has only one server, we shall used the Marginal Distribution Analysis Algorithm to demonstrate is equivalence to the Mean Value Analysis Algorithm. Both algorithms give the same value for $w = 1$ since the mean cycle time must equal the mean service time if only one job is in the network; thus,

$$CT_1(1) = 0.2 \text{ hr}, \quad CT_2(1) = 0.5 \text{ hr}, \quad CT_3(1) = 0.5 \text{ hr} .$$

Recall that $\mathbf{r} = (1,1,0.9)$ so that $\sum_{k=1}^{n} r_k CT_k(1) = 1.15$ and thus the arrival rates to the stations are

$$\lambda_1(1) = 0.8696/\text{hr}, \quad \lambda_2(1) = 0.8696/\text{hr}, \quad \lambda_3(1) = 0.7826/\text{hr} .$$

The main extra work necessitated by the multiple servers is the calculation of the marginal probabilities. From Step 4 of the algorithm, we have

$$p_1(1;1) = 0.1739/\text{hr}, \quad p_2(1;1) = 0.4348/\text{hr}, \quad p_3(1;1) = 0.3913/\text{hr} ,$$

and from Step 5, we have

$$p_1(0;1) = 0.8261/\text{hr}, \quad p_2(0;1) = 0.5652/\text{hr}, \quad p_3(0;1) = 0.6087/\text{hr} .$$

The next iteration begins with $w = 2$. The calculations for cycle time yield

$$CT_1(2) = 0.2348 \text{ hr}, \quad CT_2(2) = 0.5 \text{ hr}, \quad CT_3(2) = 0.5 \text{ hr} .$$

Notice that with two jobs being maintained in the system, the cycle time in the second and third workstations is the same as when the CONWIP level was set to one. Because there are at most two jobs in system, there cannot be a queue at Workstations 2 and 3 due to the number of servers at these workstations. Using the same logic, we would expect the mean cycle time at the second station to remain a 0.5 hr with a CONWIP level of 3 since there are three separate processors at the second workstation.

The sum after the second iteration is $\sum_{k=1}^{n} r_k CT_k(2) = 1.1848$ so that the arrival rates to the stations are

$$\lambda_1(2) = 1.6881/\text{hr}, \quad \lambda_2(2) = 1.6881/\text{hr}, \quad \lambda_3(2) = 1.5193/\text{hr} .$$

Each time the number of jobs within the factory increases, the number of calculations for the marginal probabilities also increase. The probability calculations are shown in Table 8.3.

Table 8.3 Marginal probabilities for $w = 2$ in Example 8.3

	Workstation 1	Workstation 2	Workstation 3
$p_k(1;2)$	0.2789	0.4771	0.4624
$p_k(2;2)$	0.0587	0.1835	0.1486
$p_k(0;2)$	0.6624	0.3394	0.3890

The next iteration begins with $w = 3$. The calculations for cycle time yield

$$CT_1(3) = 0.2793 \text{ hr}, \quad CT_2(3) = 0.5 \text{ hr}, \quad CT_3(3) = 0.5372 \text{ hr} .$$

The sum is $\sum_{k=1}^{n} r_k CT_k(3) = 1.2627$ so that the arrival rates to the stations are

$$\lambda_1(3) = 2.3758/\text{hr}, \quad \lambda_2(3) = 2.3758/\text{hr}, \quad \lambda_3(3) = 2.1382/\text{hr} .$$

Notice that the numerical value for $CT_1(3)$ is the same whether the algorithm of Property 8.4 or 8.3 is used, as long as the values for $CT_2(3)$ and $CT_3(3)$ come from Property 8.4. However, it does serve to check for numerical carelessness, so we will continue calculating the probabilities associated with Workstation 1. The probability calculations are shown in Table 8.4.

Table 8.4 Marginal probabilities for $w = 3$ in Example 8.3

	Workstation 1	Workstation 2	Workstation 3
$p_k(1;3)$	0.3147	0.4032	0.4159
$p_k(2;3)$	0.1325	0.2834	0.2472
$p_k(3;3)$	0.0279	0.0727	0.0794
$p_k(0;3)$	0.5248	0.2407	0.2575

The next iteration begins with $w = 4$. The calculations for cycle time yield

$$CT_1(4) = 0.3327 \text{ hr}, \quad CT_2(4) = 0.5121 \text{ hr}, \quad CT_3(4) = 0.6015 \text{ hr} .$$

The sum is $\sum_{k=1}^{n} r_k CT_k(4) = 1.3862$ so that the arrival rates to the stations are

$$\lambda_1(4) = 2.8856/\text{hr}, \quad \lambda_2(4) = 2.8856/\text{hr}, \quad \lambda_3(4) = 2.5971/\text{hr} .$$

The final probability calculations are shown in Table 8.5.

Table 8.5 Marginal probabilities for $w = 4$ in Example 8.3

	Workstation 1	Workstation 2	Workstation 3
$p_k(1;4)$	0.3029	0.3474	0.3344
$p_k(2;4)$	0.1816	0.2909	0.2700
$p_k(3;4)$	0.0765	0.1363	0.1605
$p_k(4;4)$	0.0161	0.0349	0.0516
$p_k(0;4)$	0.4229	0.1905	0.1835

Since our factory has a CONWIP level set at $w_{\max} = 5$, the final iteration involves only the first three steps of the algorithm. Then after the final cycle times and arrival rates are calculated, Little's Law can be used to obtain the workstation average *WIP* levels. These final results are contained in Table 8.6.

Table 8.6 Workstation characteristics for Example 8.3

	Workstation 1	Workstation 2	Workstation 3
$CT_k(5)$	0.392 hr	0.534 hr	0.686 hr
$\lambda_k(5)$	3.238/hr	3.238/hr	2.914/hr
$WIP_k(5)$	1.269	1.730	2.000
$u_k(5)$	0.648	0.540	0.729

The rate at which jobs can be shipped out of this system is

$$th_s = 0.9 \times 2.914 = 2.62/\text{hr}\;;$$

thus, the extra machines produced an increase of almost 90% in output. Notice also for both Example 8.2 and 8.3, the workstation *WIP* values sum to the CONWIP level, which is another convenient check on the numerical accuracy of data entries. At the CONWIP level of 5, the mean cycle time through for the manufacturing process is

$$CT_s = \frac{5}{2.62} = 1.91\ \text{hr}\,.$$

□

- *Suggestion: Do Problems 8.6 (c), 8.7 (c), 8.8, and 8.9.*

8.1.2 Analysis with General Processing Times

The Mean Value Analysis Algorithm (Property 8.3) is based on the fact that when a job arrives to a workstation within a closed network containing w total jobs, the average number of jobs ahead of the arriving job will equal the average *WIP* of that workstation for a closed network containing $w-1$ jobs. This fact is based on the exponential assumption, so that for networks containing workstations that have non-exponential processing times, an iterative method like the Mean Value Analysis Algorithm is no longer exact. Another aspect of the move from exponential service times to general distributions for service is that we need to consider the remaining processing time for the job in service at the point in time when an arrival occurs. Previously the remaining service time was not considered because of lack of memory of the exponential distribution (see the discussion around Eq. 1.16).

As an approximation, we shall continue to assume that an arriving job sees the number of jobs ahead of it based on a network with one less job. Another important assumption that is possible with exponential processing times is its memoryless property. In other words, consider a processor with first and second moments given by $E[T_s]$ and $E[T_s^2]$. Assume that a job is undergoing processing and we pick an arbitrary point in time and would like to determine the mean remaining time until processing is finished for that part. For an exponential processor, the mean remaining time is $E[T_s]$ based on the memoryless property. For a non-exponential processor, the mean remaining time is given by $E[T_s^2]/(2E[T]^2)$ (see [6]). Thus, we develop a modified Mean Value Analysis Algorithm by using the appropriate form for the remaining time for the job in process as seen at an arbitrary point in time.

As you recall, Eq. (8.2) was the basis for the Mean Value Analysis Algorithm and it is composed of three parts: (*i*) the remaining processing time for the job being serviced (if any), (*ii*) a full service time for each job in the queue when the job under consideration arrives, and (*iii*) a full service time for the arriving job. Since the utilization factor is the probability that the processor is busy, Eq. (8.2) can be written as

$$\begin{aligned} CT_k(w) &= E[T_s(k)] + E[T_s(k)]\,(WIP_k(w-1) - u_k(w-1)) \\ &\quad + u_k(w-1)\,E[T_s(k)^2]/(2E[T_s(k)])\,, \end{aligned} \tag{8.3}$$

where $u_k(w-1)$ is the utilization factor for Workstation k when there are a total of $w-1$ jobs in the network, and from Property 8.2 we have

$$u_k(w) = \frac{w\,r_k\,E[T_s(k)]}{\sum_{j=1}^{n} r_j\,CT_j(w)}\,. \tag{8.4}$$

The first step in combining Eqs. (8.3) and (8.4) is to use Little's Law to replace $WIP(w-1)$ with $CT(w-1)$ in Eq. (8.3). The utilization factor is then eliminated in Eq. (8.3) by using Eq. (8.4). Finally, we use the fact

$$E[T_s(k)^2] = E[T_s(k)]^2\,(C_s^2(k) + 1) \tag{8.5}$$

since our data usually include the SCV instead of the second moment. After simplifying, the following property is obtained that modifies the mean value analysis to an approximation procedure for non-exponential service times.

Property 8.5. *Consider a closed network with n workstations containing $w_{\max}$ jobs. Each workstation has a single processor with processor characteristics given by $E[T_s(k)]$ and $C_s^2(k)$ and the relative arrival rates to the workstations are given by the n-dimensioned vector* **r** *determined from Property 8.1. The following algorithm can be used to obtain the workstation mean cycle times.*

1. *Set $CT_k(1) = E[T_s(k)]$ for $k = 1, \cdots, n$ and set $w = 2$.*
2. *Determine $CT_k(w)$ for $k = 1, \cdots, n$ by*

$$CT_k(w) = E[T_s(k)] + \frac{(w-1)\,r_k}{\sum_{j=1}^{n} r_j\,CT_j(w-1)} \times \left[E[T_s(k)]\,CT_k(w-1) + \frac{E[T_s(k)]^2\,(C_s^2(k)-1)}{2} \right] .$$

3. *If $w = w_{\max}$, determine arrival rates from Property 8.2 and stop; otherwise, increment w by 1 and return to Step 2.*

Example 8.4. We shall illustrate this modified Mean Value Analysis Algorithm using an example problem taken from [2, Example 8.5, p. 382] where the results are acceptable but certainly not exact.

The problem has four workstations with branching probabilities from each workstation being 1/3 for each of the other workstations. Since all states are equivalent in terms of branching probabilities, the relative arrival rates to each state are the same; thus, $\mathbf{r} = (1,1,1,1)$. Throughput is counted based on entries into Workstation 1. The processing time data (namely, the mean and SCV) are given in Table 8.7.

Table 8.7 Service time characteristics for Example 8.4

Workstation k	$E[T_k]$	$C^2[T_k]$	$E[T_k^2]$
1	1.25 hr	0.25	1.953 hr^2
2	1.35 hr	1.00	3.645 hr^2
3	1.45 hr	1.00	4.205 hr^2
4	1.25 hr	0.50	2.344 hr^2

The number of jobs allowed in the closed queueing network is 15. Thus, the algorithm will take 15 iterations to reach that number. The first 4 and the last iteration values are displayed in Table 8.8, where all values are in hours.

Table 8.8 Mean cycle time results for Example 8.4

Iteration	$CT_1(w)$	$CT_2(w)$	$CT_3(w)$	$CT_4(w)$	$\sum r_j CT_j(w)$
$w = 1$	1.250	1.350	1.450	1.250	5.3
$w = 2$	1.434	1.694	1.847	1.471	6.446
$w = 3$	1.624	2.060	2.281	1.699	7.664
$w = 4$	1.815	2.438	2.745	1.929	8.927
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$w = 15$	3.609	6.738	9.231	4.081	23.659

Since the relative arrival rates are the same (i.e., $r_k = 1$, for $k = 1,\cdots,4$), the arrival to each state is the same and thus from Property 8.2 the system throughput is

$$th(15) = \lambda_1(15) = \frac{15 \times r_1}{\sum_{j=1}^{4} r_j CT_j} = \frac{15}{23.659} = 0.634/\text{hr} .$$

The throughput estimation for this system agrees to three decimal places with the simulation results of [2]. The cycle time results, however, are not as impressive. If we assume that the simulation results contained in [2] for this example are the exact values, then the percent errors in the mean cycle time estimates for the four workstations are -13.0%, 4.9%, 5.1%, and -4.6%. Thus, for non-exponential service times, the algorithm of Property 8.5 yields acceptable but far from perfect results. However, if we naively used the exponential assumption by following Property 8.3, the errors for the four workstations would be 30.3%, 15.4%, 18.7%, and 26%; thus, if the service time SCV is not 1, it is best to take advantage of the modified version of the Mean Value Analysis. □

- *Suggestion: Do Problems 8.10 and 8.11.*

8.2 Closed Queueing Networks with Multiple Products

It is not too difficult to extend the (single-server) Mean Value Analysis Algorithm to account for multiple products; however, the implementation of the algorithm becomes intractable with more than just a couple of products and modest CONWIP levels.

As in Chap. 6, notation will become more cumbersome since there are more quantities that must be reflected in the notation. For the most part, we will be able to use similar notation as was used in Chap. 6; namely, the index i will be used for the job type (product) and it will often be written as a superscript. The total number of job types will be m.

Each job type will have its own routing matrix and thus its own relative arrival rates which will be denoted by the vector $\mathbf{r}^i = (r_{i,1}, \cdots, r_{i,n})$. The value of $\mathbf{r}^i$ is determined by Property 8.1, where the matrix P and submatrix Q of the property are replaced by the routing matrix P^i and submatrix Q^i that describe the switching probabilities associated with Job Type i.

With multiple job types, a separate CONWIP level must be specified for each type. In other words, when a Type i job is finished, another Type i will be started. Since we assume that there are m different job types, the CONWIP level is a vector called $\mathbf{w} = (w^1, \cdots, w^m)$. The vector $\mathbf{e}_i$ is used to specify the unit vector with a one in the i-position and zeros elsewhere. The unit vector is used to indicate a decrease (or increase) of one unit of a specified job type. For example, the vector $\mathbf{w} - \mathbf{e}_1$ represents a CONWIP level specified by $\mathbf{w}$ except with one less of Type 1; thus, $\mathbf{w} - \mathbf{e}_1 = (w^1 - 1, w^2, \cdots, w^m)$.

In the next section, the Mean Value Analysis Algorithm will be extended and a small example will be used to demonstrate its implementation. Then in Sect. 8.2.2 an approximation will be derived that gives a much easier implementation with reasonable results as long as the total CONWIP level is not too small. Finally, in Sect. 8.2.3

the approximation will be extended to non-exponential processing times, although in such cases the approximation is not as accurate as for the single-product algorithm.

8.2.1 Mean Value Analysis for Multiple Products

There are two key concepts used in the Mean Value Analysis Algorithm. The first is that an arriving job must wait for all jobs within the workstation to be processed plus its own processing. The second is that the number of jobs seen by an arriving job in a system containing w jobs is the mean WIP for a system with $w-1$ jobs. In other words, the arriving job cannot be behind itself and, thus, sees only $w-1$ other jobs. These concepts are still true except that now we need to consider all other job types. Thus, the main relationship is

$$CT_k^i(\mathbf{w}) = E[T_s(i,k)] + \sum_{\ell=1}^{m} E[T_s(\ell,k)]\, WIP_k^\ell(\mathbf{w}-\mathbf{e}_i)\ . \tag{8.6}$$

The arrival rate of Job Type i into the Workstation k is also the same as in Property 8.2 except that it must be determined separately for each type.

Property 8.6. *Consider a closed network with n workstations and m job types. The vector* $\mathbf{w}$ *designates the total number of jobs in the network of the various types, and the relative arrival rates to the workstations for Job Type i are given by the n-dimensioned vector* $\mathbf{r}^i$ *determined from Property 8.1 adjusted by using the routing matrix* P^i*. The arrival rate to Workstation k for Job Type i is*

$$\lambda_{i,k}(\mathbf{w}) = \frac{w^i\, r_{i,k}}{\sum_{j=1}^{n} r_{i,j}\, CT_j^i(\mathbf{w})}\ ,$$

where w^i *is the* i^{th} *component of the vector* $\mathbf{w}$ *(i.e., the total number of Type i jobs in the network) and* $r_{i,k}$ *is the* k^{th} *component of* $\mathbf{r}^i$*.*

With an expression for the arrival rate, we can use Little's Law to replace the WIP term in Eq. (8.6); however, care must be taken because the term $\mathbf{w}-\mathbf{e}_i$ stays the same as the index of summation varies in the expression $WIP_k^\ell(\mathbf{w}-\mathbf{e}_i)$ for Eq. (8.6). The reason for this is that the number of Type i jobs that a Type i job "sees" is w^i-1; whereas the number of Type ℓ jobs that a Type i job sees is w^ℓ (we do not subtract a 1) for $\ell \neq i$. Thus, when applying Little's Law to Eq. (8.6), the term for $\ell = i$ will need to be listed separately as shown in the following iterative equation for the mean cycle time of Job Type i in Workstation k

$$CT_k^i(\mathbf{w}) = E[T_s(i,k)] + E[T_s(i,k)]\,\frac{(w^i-1)\,r_{i,k}\,CT_k^i(\mathbf{w}-\mathbf{e}_i)}{\sum_{j=1}^n r_{i,j}\,CT_j^i(\mathbf{w}-\mathbf{e}_i)} + \sum_{\substack{\ell=1\\ \ell\neq i}}^{m} E[T_s(\ell,k)]\,\frac{w^\ell\, r_{\ell,k}\,CT_k^\ell(\mathbf{w}-\mathbf{e}_i)}{\sum_{j=1}^n r_{\ell,j}\,CT_j^\ell(\mathbf{w}-\mathbf{e}_i)}\,, \tag{8.7}$$

where w^ℓ is the total number of Type ℓ jobs within the closed network, and the expression that is written as a ratio is evaluated to zero if the numerator is zero, even though the denominator will also be zero. Conceptually, Eq. (8.7) is very similar to the iterative expression in Property 8.3; however, implementation is significantly worse because of the necessity to determine the cycle time for all combinations of the vector $\mathbf{w}$ whose individual components are less that their maximum value. In developing the next algorithm, we shall let $\mathbf{w}_{\max}$ denote the vector $(w_{\max}^1,\cdots,w_{\max}^m)$ so that it is possible for each component to have its own maximum value. We also let $\mathbf{0}$ denote an m-dimensioned vector of all zeros. Finally, we let $|\mathbf{w}| = \sum_{i=1}^m |w^i|$.

Property 8.7. *Consider a closed network with n workstations, m job types, and $\mathbf{w}_{\max}$ designating the total number of jobs in the network of the various types. Each workstation has a single exponential server and the relative arrival rates to the workstations for Job Type i are given by the n-dimensioned vector $\mathbf{r}^i$. The following algorithm can be used to obtain the mean cycle times for Type i jobs at Workstation k.*

1. *Set $CT_k^i(\mathbf{0}) = 0$ and $CT_k^i(\mathbf{e}_i) = E[T_s(i,k)]$ for $k = 1,\cdots,n$ and $i = 1,\cdots,m$. Set $W = 2$.*
2. *For each $\mathbf{w}$ such that $|\mathbf{w}| = W$ and each $w^i \le w_{\max}$, determine $CT_k^i(\mathbf{w})$ for $i = 1,\cdots,m$ and $k = 1,\cdots,n$ from Eq. (8.7).*
3. *If $W = |\mathbf{w}_{\max}| = \sum_{i=1}^m w_{\max}$, determine all arrival rates from Property 8.6 with $\mathbf{w} = \mathbf{w}_{\max}$ and stop; otherwise, increment W by 1 and return to Step 2.*

Once the arrival rates have been determined, the WIP in each station for each job type can be determined and as a check for numerical accuracy, the sum of the WIP should equal $|\mathbf{w}_{\max}|$. Step 2 contains several “sub-steps” since there will be several vectors $\mathbf{w}$ that satisfy the stated condition, and the order in which the cycle times are evaluated is important. Whenever a value of $CT_k^i(\mathbf{w})$ is to be calculated, it is important that the value of $CT_k^i(\mathbf{w}-\mathbf{e}_i)$ has already been determined. To insure this, the algorithm should proceed through the possible values of $\mathbf{w}$ in either lexicographical order or reverse lexicographical order. (Lexicographical order is the order in a dictionary; thus (2,5,3,0) comes before (3,0,2,7) in lexicographical order.)

Example 8.5. Consider a manufacturing facility that has three workstations, with a single machine in each workstation. There are two products that are produced simultaneously in the factory. The workstation flow diagram for each product are

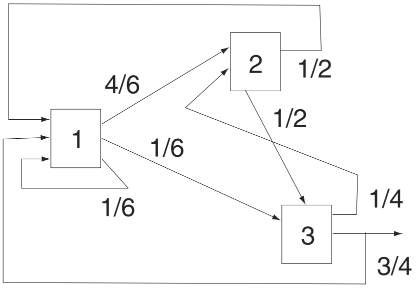


Fig. 8.2a Workstation flow diagram for Product 1 of Example 8.5

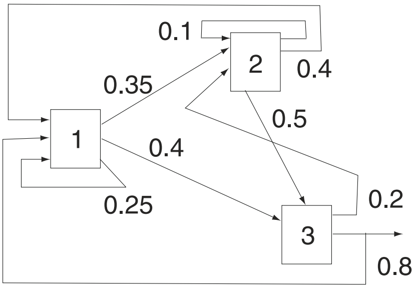


Fig. 8.2b Workstation flow diagram for Product 2 of Example 8.5

given in Figs. 8.2a and 8.2b. There are 5 Type 1 jobs allowed in the system and 8 Type 2 jobs. The processing times are all assumed to be exponentially distributed with mean values given in Table 8.9.

Table 8.9 Mean processing times for Example 8.5

		Workstation	
Product	$E[T_s(i,1)]$	$E[T_s(i,2)]$	$E[T_s(i,3)]$
$i = 1$	0.25 hr	0.50 hr	1.0 hr
$i = 2$	1.20 hr	0.75 hr	0.3 hr

The two flow diagrams of the figures yield two routing matrices, P^1 and P^2, which in turn yield two vectors giving the relative arrival rates to each workstation for the two job types. These vectors are obtained by using the matrices P^1 and P^2 in place of P in Property 8.1:

$$\begin{aligned} \mathbf{r}^1 &= (1.0, 0.810, 0.571) \quad \text{and} \\ \mathbf{r}^2 &= (1.0, 0.538, 0.669)\,. \end{aligned} \tag{8.8}$$

The first several iterations obtained when the algorithm of Property 8.7 is applied to this problem gives the results displayed in Table 8.10. Notice that for each iteration of Step 2, the values of **w** are ordered lexicographically.

Table 8.10 Mean cycle time values in hours for Example 8.5

	For Product 1				For Product 2			
	WS 1	WS 2	WS 3		WS 1	WS 2	WS 3	
$\mathbf{w}$	$CT_1^1(\mathbf{w})$	$CT_2^1(\mathbf{w})$	$CT_3^1(\mathbf{w})$	$\sum r_k^1 CT_k^1$	$CT_1^2(\mathbf{w})$	$CT_2^2(\mathbf{w})$	$CT_3^2(\mathbf{w})$	$\sum r_k^2 CT_k^2$
(0,1)	0	0	0	0	1.2	0.75	0.3	1.8042
(1,0)	0.25	0.5	1	1.226	0	0	0	0
(0,2)	0	0	0	0	1.998	0.918	0.333	2.715
(1,1)	1.048	0.668	1.033	2.179	1.251	0.915	0.766	2.256
(2,0)	0.301	0.665	1.466	1.677	0	0	0	0
(0,3)	0	0	0	0	2.966	1.023	0.349	3.75
(1,2)	2.016	0.773	1.049	3.241	1.986	1.038	0.639	2.972
(2,1)	1.036	0.788	1.339	2.438	1.29	1.071	1.298	2.735
(3,0)	0.34	0.821	1.998	2.146	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
(5,8)	9.601	1.062	1.379	11.249	9.378	1.407	0.848	10.701

With the cycle time calculations complete, the arrival rates (and thus throughput rates) at each workstation by job type can be calculated using Property 8.6. Then, with the arrival rates the *WIP* calculations are possible through Little's Law. These results are contained in Table 8.11.

Table 8.11 Arrival rate and *WIP* for Example 8.5 at its CONWIP levels

	For Product 1			For Product 2		
	WS 1	WS 2	WS 3	WS 1	WS 2	WS 3
$\lambda_{i,k}$	0.444/hr	0.36/hr	0.254/hr	0.748/hr	0.402/hr	0.5/hr
WIP_k^i	4.268	0.382	0.35	7.01	0.566	0.424

Returning to Figs. 8.2a and 8.2b, we see that 75% of the Type 1 jobs that leave Workstation 3 are considered finished product and 80% of the Type 2 jobs that leave Workstation 3 are considered finished; thus the product throughput rates for this factory are

$$\begin{aligned} th_s^1 &= 0.75 \times 0.254 = 0.1905/\text{hr} \\ th_s^2 &= 0.80 \times 0.5 = 0.4/\text{hr} . \end{aligned}$$

By summing the individual workstation *WIP* levels by product, we obtain the pre-established CONWIP levels of 5 and 8 which then yield cycle times by product of

$$\begin{aligned} CT_s^1 &= \frac{5}{0.1905} = 26.25 \text{ hr} \\ CT_s^2 &= \frac{8}{0.4} = 20.0 \text{ hr} . \end{aligned}$$

The total factory throughput is the sum of the throughputs for the two products yielding

$$th_s = 0.1905 + 0.4 = 0.5905/\text{hr} ,$$

and the cycle time for an arbitrary job is

$$CT_s = \frac{13}{0.5905} = 22.02 \text{ hr} .$$

□

- *Suggestion: Do Problems 8.12 and 8.16 (a,b).*

8.2.2 Mean Value Analysis Approximation for Multiple Products

It is obvious that the algorithm of Property 8.7 will result in too many calculations when there are several different job types and high level of CONWIP control. However, this is not a problem is such cases because of the availability of a reasonable approximation. If the total number of jobs within the closed network is large, then removing one item from the factory will not change the cycle time significantly. This fact would indicate that the cycle time expression found on the left and right hand side of Eq. (8.7) are approximately the same and we can drop the cycle time dependence on the vector $\mathbf{w}$. This leads to the following recursive system of equations for $k = 1, \cdots, n$ and $i = 1, \cdots, m$ the defines (approximately) the mean cycle time at Workstation k for Job Type i:

$$\begin{aligned} CT_k^i &= E[T_s(i,k)] + E[T_s(i,k)] \frac{(w_{\max}^i - 1)\, r_{i,k}\, CT_k^i}{\sum_{j=1}^n r_{i,j}\, CT_k^i} \\ &+ \sum_{\substack{\ell=1 \\ \ell \neq i}}^{m} E[T_s(\ell,k)] \frac{w_{\max}^{\ell}\, r_{\ell,k}\, CT_k^{\ell}}{\sum_{j=1}^n r_{\ell,j}\, CT_j^{\ell}} , \end{aligned} \tag{8.9}$$

where $w_{\max}^i$ is the total number of Type i jobs within the network.

Because it is a recursive equation that is also a contraction mapping, it is relatively easy to write an iterative procedure that will yield estimates for the cycle times.

Property 8.8. *Consider a closed network with n workstations, m job types, and* $\mathbf{w}_{\max}$ *designating the total number of jobs in the network of the various types. Each workstation has a single exponential server and the relative arrival rates to the workstations for Job Type i are given by the n-dimensioned vector* $\mathbf{r}^i$. *The following algorithm can be used to approximate the mean cycle times for Type i jobs at Workstation k.*

1. *Set* $CT^i_{k,old} = E[T_s(i,k)]$ *for* $k = 1, \cdots, n$ *and* $i = 1, \cdots, m$.
2. *For each* $k = 1, \cdots, n$ *and* $i = 1, \cdots, m$, *obtain values for* $CT^i_{k,new}$ *by using Eq. (8.9) with the* $CT^i_{k,old}$ *values used for the right-hand side cycle time values and the* $CT^i_{k,new}$ *values are from the left-hand side.*
3. *Let the error term be defined as* $\max_{i,k}\{|CT^i_{k,new} - CT^i_{k,old}|\}$, *and if the error term is less than* 10^{-5} *(or other chosen limit), stop; otherwise, let the* $CT^i_{k,old}$ *values become the* $CT^i_{k,new}$ *values and repeat Step 2.*

Example 8.6. Consider again the manufacturing facility of Example 8.5 and illustrated with Figs. 8.2a and 8.2b. Using the values from Table 8.9 and Eq. (8.8) we get the following iteration, where all values are in hours.

Table 8.12 Mean cycle time values in hours for Example 8.6

	For Product 1				For Product 2			
	WS 1	WS 2	WS 3		WS 1	WS 2	WS 3	
Iteration #	CT^1_1	CT^1_2	CT^1_3	$\sum r^1_k CT^1_k$	CT^2_1	CT^2_2	CT^2_3	$\sum r^2_k CT^2_k$
1	0.25	0.5	1	1.226	1.2	0.75	0.3	1.8042
2	6.839	2.5026	3.1299	10.6533	7.0419	2.75	2.8623	10.4362
3	7.3696	1.7311	2.1114	9.9774	7.6704	1.97	1.5241	9.7498
4	8.5411	1.4333	1.7343	10.6924	8.7317	1.672	1.1238	10.3831
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
15	9.6747	1.0166	1.3911	11.2925	9.7679	1.2567	0.7477	10.9443

Table 8.13 Arrival rate and *WIP* for Example 8.5 at its CONWIP levels of 5 and 8

	For Product 1			For Product 2		
	WS 1	WS 2	WS 3	WS 1	WS 2	WS 3
λ^i_k	0.4428/hr	0.3586/hr	0.2528/hr	0.731/hr	0.3933/hr	0.489/hr
WIP^i_k	4.2837	0.3646	0.3517	7.1401	0.4942	0.3656

Since 75% of the Type 1 jobs that leave Workstation 3 are considered finished product and 80% of the Type 2 jobs that leave Workstation 3 are considered finished, the product throughput rates for this factory are

$$\begin{aligned} th^1_s &= 0.75 \times 0.2528 = 0.1896/\text{hr} \\ th^2_s &= 0.80 \times 0.489 = 0.3912/\text{hr}\,. \end{aligned}$$

By summing the individual workstation *WIP* levels by product, we obtain the pre-established CONWIP levels of 5 and 8 which then yield cycle times by product of

$$CT_s^1 = \frac{5}{0.1896} = 26.37 \text{ hr}$$
$$CT_s^2 = \frac{8}{0.3912} = 20.45 \text{ hr}$$

The total factory throughput is the sum of the throughputs for the two products yielding

$$th_s = 0.1896 + 0.3912 = 0.5808/\text{hr} ,$$

and the cycle time for an arbitrary job is

$$CT_s = \frac{13}{0.5808} = 22.38 \text{ hr} .$$

Thus, the approximation from the algorithm of Property 8.8 yielded system estimates within 1.6% of the actual values. □

- *Suggestion: Do Problems 8.13, 8.14, 8.15, and 8.16 (a,c,d,e).*

8.2.3 General Service Time Approximation for Multiple Products

The extension of the multiple product mean value analysis to non-exponential servers is conceptually the same as for the single product system. Thus, our approach in this section is to combine the methodology of Sect. 8.1.2 with the exact mean value analysis methodology of Sect. 8.2.1. We then form the approximation using the same logic as in Sect. 8.2.2; that is, we assume enough jobs within the network so that removing one job will not make a significant difference in the cycle time values.

We first extend Eq. (8.3) to include the multi-product case in an analogous equation to that of (8.6); namely, the mean cycle time at Workstation k for a Type i job is

$$CT_k^i(\mathbf{w}) = E[T_s(i,k)] + \sum_{\ell=1}^{m} \Big\{ E[T_s(\ell,k)] \left(WIP_k^\ell(\mathbf{w} - \mathbf{e}_i) - u_{\ell,k}(\mathbf{w} - \mathbf{e}_i) \right) + u_{\ell,k}(\mathbf{w} - \mathbf{e}_i) \frac{E[T_s^2(\ell,k)]}{2E[T_s(\ell,k)]} \Big\} . \tag{8.10}$$

The utilization factor for Job Type i at the single-server Workstation k is the arrival rate times the mean service time, or

$$u_k^i(\mathbf{w}) = \lambda_{i,k}(\mathbf{w}) E[T_s(i,k)] = \frac{w^i r_{i,k} E[T_s(i,k)]}{\sum_{j=1}^{n} r_{i,j} CT_j^i} , \tag{8.11}$$

where w^i is the amount of Type i jobs in the network. The manipulation of Eq. (8.10) is now very similar to the process used to derive the equation of Property 8.5. We use Eq. (8.11) to eliminate the utilization factor, use Little's Law to eliminate the *WIP* terms, and finally use Eq. (8.5) to replace the second moment by the SCV term; thus, we have

$$
\begin{aligned}
CT_k^i(\mathbf{w}) = E[T_s(i,k)] &+ \frac{(w^i - 1)\, r_{i,k}}{\sum_{j=1}^n r_{i,j}\, CT_j^i(\mathbf{w} - \mathbf{e}_i)} \\
&\times \left[E[T_s(i,k)]\, CT_k^i(\mathbf{w} - \mathbf{e}_i) + \frac{E[T_s(i,k)]^2\, (C_s^2(i,k) - 1)}{2} \right] \\
&+ \sum_{\substack{\ell=1 \\ \ell \neq i}}^{m} \frac{w^\ell\, r_{\ell,k}}{\sum_{j=1}^n r_{\ell,j}\, CT_j^\ell(\mathbf{w} - \mathbf{e}_i)} \\
&\times \left[E[T_s(\ell,k)]\, CT_k^\ell(\mathbf{w} - \mathbf{e}_i) + \frac{E[T_s(\ell,k)]^2\, (C_s^2(\ell,k) - 1)}{2} \right] ,
\end{aligned}
$$

where $C_s^2(i,k)$ is the squared coefficient of variation of the service times for Job Type i at Workstation k. If we assume sufficient jobs within the network so that the cycle time is approximately the same when one job is removed, we have the following equation that can be used in our approximation algorithm for non-exponential, multi-product closed networks.

$$
\begin{aligned}
CT_k^i = E[T_s(i,k)] & \\
&+ \frac{(w_{\max}^i - 1)\, r_{i,k}}{\sum_{j=1}^n r_{i,j}\, CT_j^i} \times \left[E[T_s(i,k)]\, CT_k^i + \frac{E[T_s(i,k)]^2\, (C_s^2(i,k) - 1)}{2} \right] \\
&+ \sum_{\substack{\ell=1 \\ \ell \neq i}}^{m} \frac{w_{\max}^\ell\, r_{\ell,k}}{\sum_{j=1}^n r_{\ell,j}\, CT_j^\ell} \times \left[E[T_s(\ell,k)]\, CT_k^\ell + \frac{E[T_s(\ell,k)]^2\, (C_s^2(\ell,k) - 1)}{2} \right] .
\end{aligned}
\tag{8.12}
$$

The resulting algorithm does not yield as accurate results as one would like. It does, however, produce usable results and can serve as a starting point for further approximation developments.

Property 8.9. *Consider a closed network with n workstations, m job types, and* $\mathbf{w}_{\max}$ *designating the total number of jobs in the network of the various types. Each workstation has a single processor with processor time and SCV for Job Type i begin given by* $E[T_s(i,k)]$ *and* $C_s^2(i,k)$*, respectively, and with the relative arrival rates to the workstations for Job Type i are given by the n-dimensioned vector* $\mathbf{r}^i$*. The following algorithm can be used to approximate the mean cycle times for Type i jobs at Workstation k.*

1. *Set* $CT_{k,old}^i = E[T_s(i,k)]$ *for* $k = 1, \cdots, n$ *and* $i = 1, \cdots, m$.

2. *For each $k = 1, \cdots, n$ and $i = 1, \cdots, m$, obtain values for $CT^i_{k,new}$ by using Eq. (8.12) with the $CT^i_{k,old}$ values used for the right-hand side cycle time values and the $CT^i_{k,new}$ values are from the left-hand side.*
3. *Let the error term be defined as $\max_{i,k}\{|CT^i_{k,new} - CT^i_{k,old}|\}$, and if the error term is less than 10^{-5} (or other chosen limit), stop; otherwise, let the $CT^i_{k,old}$ values become the $CT^i_{k,new}$ values and repeat Step 2.*

The resulting model is not as accurate as one would like. It does, however, yield usable results and can serve as a starting point for further approximation developments.

Example 8.7. Consider a two-product three-workstation problem with a limit of 7 and 6 jobs in the system for the two products. The workstations flow probabilities for the two products and the relative arrival rates are given in Table 8.14. Notice that the rates in Table 8.14 (namely, $\mathbf{r}^1$ and $\mathbf{r}^2$) are computed using the job type specific switching probabilities contained in the table and then applying Property 8.1. The workstations processing time data (means and SCV's) are displayed in Table 8.15.

Table 8.14 Flow probabilities and relative arrival rates for Example 8.7

Product	From/To	1	2	3	Arrival Rates
1	1	0	0.3	0.7	1/hr
	2	0.1	0	0.9	0.3/hr
	3	1	0	0	0.97/hr
2	1	0	1	0	1/hr
	2	0.1	0	0.9	1/hr
	3	1	0	0	0.9/hr

Table 8.15 Processing time data for Example 8.7

			Workstation	
Product	Measure	$k = 1$	$k = 2$	$k = 3$
$i = 1$	$E[T_s(1,k)]$	0.60 hr	1.00 hr	0.50 hr
	$C_s^2(1,k)$	0.50	1.00	1.50
$i = 2$	$E[T_s(2,k)]$	0.20 hr	0.60 hr	0.50 hr
	$C_s^2(2,k)$	1.50	1.00	0.75

The first few iterations of the algorithm from Property 8.9 are displayed in Table 8.16.

After the final iteration, the arrival rates can be determined by Property 8.6 and then used with the cycle times to determine *WIP* levels. The arrival rates are also

Table 8.16 Iterative results for cycle times (in hours) for Example 8.7

		For Product 1				For Product 2		
	WS 1	WS 2	WS 3		WS 1	WS 2	WS 3	
Iteration #	CT_1^1	CT_2^1	CT_3^1	$\sum r_k^1 CT_k^1$	CT_1^2	CT_2^2	CT_3^2	$\sum r_k^2 CT_k^2$
1	0.6	1	0.5	1.385	0.2	0.6	0.5	1.25
2	2.0097	4.0276	2.7582	5.8934	1.7646	3.5562	2.8195	7.8585
3	2.0131	3.8593	2.8709	5.9556	1.7562	3.3928	2.9503	7.8043
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
40	1.8748	3.3697	3.3409	6.1264	1.6002	2.9048	3.4162	7.5796

combined with the mean service times to obtain workstation utilizations. These workstation performance measures by job type are given in Table 8.17.

Table 8.17 Workstation performance measures by job type for Example 8.7

		For Product 1			For Product 2	
Measure	WS 1	WS 2	WS 3	WS 1	WS 2	WS 3
Arrival Rates	1.1426/hr	0.3428/hr	1.1083/hr	0.7916/hr	0.7916/hr	0.7124/hr
WIP	2.1422	1.1551	3.7028	1.2667	2.2995	2.4338
Utilization Factor	0.6856	0.3428	0.5542	0.1583	0.475	0.3562

The composite workstation measures for arrival rates, *WIP*, and utilization are obtained by summing across product types. The workstation cycle time is obtained using the combined *WIP* and arrival rates together with Little's Law. These measures are found in Table 8.18. We assume for this example that whenever a job leaves Workstation 3 it is finished and a new job starts in Workstation 1; therefore, the throughput for Workstation 3 is the factory throughput; thus the system estimates are

Table 8.18 Workstation characteristics for Example 8.7

Measure	WS 1	WS 2	WS 3
WIP_k	3.4089	3.4546	6.1366
λ_k	1.9342/hr	1.1344/hr	1.8207/hr
u_k	0.8439	0.8178	0.9104
CT_k	1.7624 hr	3.0453 hr	3.3705 hr

$$th_s = 1.8207/\text{hr} \quad \text{and}\; CT_s = \frac{13}{1.8207} = 7.1401 \text{ hr} .$$

A simulation model was written to give a feeling for the accuracy of this approximation. The simulation was run so that the half-width of all confidence intervals was less that 0.01 and the results are shown in Table 8.19. The simulated estimate for the mean system cycle time was 7.06 hr. The arrival rates and factory throughput are good. The cycle time measures are not as good, but they are still acceptable for many applications although there is clear room for improvement. □

Table 8.19 Simulation results for Example 8.7

Measure	WS 1	WS 2	WS 3
WIP_k	4.03	3.36	5.60
λ_k	1.96/hr	1.14/hr	1.84/hr
u_k	0.86	0.82	0.92
CT_k	2.06 hr	2.96 hr	3.04 hr

The algorithm based on Eq. (8.12) is not unique. Notice in Eq. (8.10), the remaining time for the job undergoing processing depends on the job type; however, it is also reasonable to replace those terms with the remaining time using the workstation service time averaged over all job types. Specifically, the workstation mean service time is given as

$$E[T_s(k)] = \frac{\sum_{\ell=1}^{m} \lambda_{\ell,k}(\mathbf{w}) E[T_s^\ell(k)]}{\sum_{\ell=1}^{m} \lambda_{\ell,k}(\mathbf{w})} , \tag{8.13}$$

where $\mathbf{w}$ specifies the fixed *WIP* level for each type job. (The value of $E[T_s(k)]$ clearly depends on *WIP* levels since these effect arrival rates; however, we shall ignore this in our notation as we try to keep notation as clean as possible.) The utilization factor for the workstation could then be approximated by $u_k = \sum_\ell \lambda_{\ell,k} E[T_s(k)]$. With this modified definition of utilization, Eq. (8.10) can be modified to be

$$\begin{aligned} CT_k^i &= E[T_s(i,k)] + \sum_{\ell=1}^{m} \lambda_{\ell,k} E[T_{\ell,k}] \left(CT_k^\ell - E[T_s(k)] \right) \\ &\quad + \left(\sum_{\ell=1}^{m} \lambda_{\ell,k} \right) E[T_s(k)] \times \frac{E[T_s^2(k)]}{2E[T_s(k)]} . \end{aligned} \tag{8.14}$$

Equation (8.14) can now be used with Property 8.9 for the single-server, multiple product case; however, the equation also is easily modified to be used as an approximation for the multi-server case. An extension for multiple servers per workstation suggested in Askin and Standridge [1] for the exponential approximation and by Buzacott and Shanthikumar [2] for the general model is to adjust the service times by dividing by the number of servers. Then the utilization factor u_k for a workstation is no longer the probability that the arriving job must wait and a better approximation is $(u_k/c_k)^{c_k}$, where c_k is the number of identical servers at workstation k. This results in the following recursive relationship:

$$\begin{aligned} CT_k^i &= E[T_s(i,k)] + \sum_{\ell=1}^{m} \lambda_{\ell,k} \frac{E[T_{\ell,k}]}{c_k} \left(CT_k^\ell - E[T_s(k)] \right) \\ &\quad + \left(\frac{E[T_s(k)]}{c_k} \sum_{\ell=1}^{m} \lambda_{\ell,k} \right)^{c_k} \times \frac{E[T_s^2(k)]}{2\,c_k E[T_s(k)]} . \end{aligned} \tag{8.15}$$

Thus, for the multi-server case, Property 8.9 could be used with Eq. (8.15) to yield approximate cycle times. The iterations are slightly more involved because the values for $\lambda_{i,k}$ and $E[T_s(k)]$ must be calculated after each iteration in order to obtain the next estimates for CT_k^i, but the extra calculations are not difficult.

Although the multi-server approximation based on Eq. (8.15) can give reasonable results, an iterative method proposed by Marie [9, 10] has been shown to often give superior results. Marie's method uses an aggregation technique that is beyond the scope of this text; however, the method is worth investigating for those interested in modeling multi-server networks with non-exponential processing times.

- *Suggestion: Do Problem 8.17.*

8.3 Production and Sequencing Strategies: A Case Study

In manufacturing systems analysis, the concept of just-in-time manufacturing has received significant interest in recent years. Based on Toyota of Japan's kanban control concept, the "pull" manufacturing strategy has evolved. This strategy is fundamentally different from the traditional MPR-type "push" release strategy. Pull versus push production release strategies can have a profound impact on the cycle time for products. This case study will investigate the differences of pull and push release strategies and the impact of scheduling rules on cycle time.

A push-release strategy is one where products are released to the manufacturing system based on a schedule. This schedule is usually derived from orders or order forecasts and the schedule developed based on typical or "standard" production cycle times (one such mechanism is the MRP strategy). A pull-release strategy, on the other hand, is one where orders are authorized for release into the shop based on the completion of processing within the shop. For example, one pull-based control policy is to have a fixed number of parts being manufactured within the shop at any one time, i.e., a CONWIP control system just analyzed. Hence, when a part is completed and ready for shipping, the next part is released to the shop. In this way, the *WIP* is controlled and the manufacturing flow times are reduced.

The combination of the job-release strategy (into the system) and the job-sequencing strategy (at a machine center) can have a significant impact on product cycle times. In this case study, these concepts are illustrated and it is shown how to model these schemes for the most complex manufacturing environment - the job shop. A job shop is a manufacturing system where production steps that require the same machinery are processed within the same production area called a machine center. Furthermore, different part types have different routings through the shop and can require multiple processing steps on the same machine. Thus, if the first, third and fifth processing steps require the same machine (processing times, of course, can vary by processing step), then the part is routed back to the same machine center for processing. Due to the re-entrant flow or feedback nature of the processing routings, this type of manufacturing system is more complex to design and control than the straight-through manufacturing design (called a flow-shop).

In this study, the simulation language used (MOR/DS) was developed by the authors [3]; however, the mechanics of simulation model itself are not discussed. Our concern is with the results of the study and their implications. The purpose of this section is two-fold. First, we have always assumed a FIFO (first-in first-out) queueing discipline, and this is not always the best possible. Therefore, this case study should serve to emphasize that different priority schemes can have a significant effect on cycle times and that a FIFO system is not always preferred. Although this text presents many mathematical models for manufacturing and production systems, there are still many problems, especially those dealing with sequencing issues, for which good mathematical approximations must be developed. Thus the second purpose of this section is to illustrate the importance of sequencing and encourage the development of analysis techniques in this area.

The job-release strategies and the processing sequencing strategies at the workstations in combination have been the subject of several research studies in recent years. In this case study, a job shop model is used in conjunction with the strategies reported in the papers by Wein [13, 14], Harrison and Wein [7], Spearman et al. [12], and Duenyas [5] to illustrate in impact that various control strategies can have on cycle times. We do not have the space in this textbook to discuss the modeling of sequencing in coordination with other control strategies, but it is important for the reader to have some understanding of the profound impact that sequencing can have on factory performance.

8.3.1 Problem Statement

The problem used for illustration purposes is from Wein [14]. This model consists of three single workstations and three part types to be produced. The part routings through the machine centers and their mean processing times are listed in Table 8.20. Figure 8.3 illustrates the product flow through the facility. All processing times are assumed to be exponentially distributed and the load on each machine center averages 89.4% utilization. The three products are to be produced in equal quantities with a required total output rate of 8.94 per hour.

Table 8.20 Three-product, three-workstation job shop data from Wein [14]

Product Type	Processing Route	Mean Times
1	$3 \to 1 \to 2$	6 min : 4 min : 1 min
2	$1 \to 2 \to 3 \to 1 \to 2$	8 min : 6 min : 1 min : 2 min : 7 min
3	$2 \to 3 \to 1 \to 3$	4 min : 9 min : 4 min : 2 min

Of interest are the effects of a variety of release and sequencing rules on the average cycle time for all products. For a workstation with a single machine and a

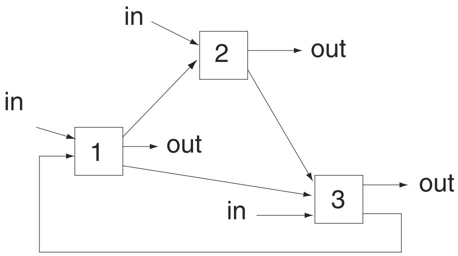


Fig. 8.3 Routing diagram for the problem of Sect. 8.3

fixed job set, sequencing the jobs according to the shortest expected processing time ($SEPT$) rule always yields the shortest cycle time; however, it is not necessarily optimal for a general job-shop. In this case study, various policies are compared among themselves and with a standard push or deterministic release schedule. The push variation of the job shop model is given first. The data from Table 8.20 and the release rates are incorporated into the model. For model specification, there is a release rule and a sequencing rule for selecting the next job to be processed at the machines. The standard first-in first-out sequencing rule is called the $FIFO$ rule. For job releases into the shop, note that the simulation uses time between releases and so a deterministic rate of 8.94 per hour results in the time between releases of 6.71 minutes. If the three job types are released separately, then the release time is 20.13 minutes for each job type since they are to be released in equal numbers. This specific model configuration is denoted as $Deter - FIFO$ (deterministic release, first-in first-out sequencing). Other job sequencing rules order jobs at the machines in some specified sequence; here the smallest number has the highest priority. The sequencing rules that will be considered are: $SEPT$ - smallest expected processing time at the current machine, $SRPT$ - smallest remaining expected processing time at all remaining machines, and $WBAL$ - a workload balance sequencing rule proposed by Harrison and Wein [7], which is essentially $SRPT$ by workstation (includes all remaining visitations to the current workstation).

8.3.2 Push Strategy Model

The general push processing release model is setup to run different sequencing rules for the deterministic release policy. The three different polices considered are the $SEPT$, $SRPT$, and $WBAL$ rules. The workload-balance sequencing rule ($WBAL$) assigns priorities to the jobs at each workstation in the processing sequence according to:

Workstation 1: type 2 - step 4, type 3 - step 3, type 1 - step 2, and type 2- step 1;
Workstation 2: type 1 - step 3, type 3 - step 1, type 2 - step 5, and type 2 - step 2 ;
Workstation 3: type 2 - step 3, type 3 - step 4, type 1 - step 1, and type 3 - step 2.

The workload balance priority sequencing rule (*WBAL*) gives the highest priority at Workstation *i* to the job with the smallest remaining expected processing time to be performed by Workstation *i* throughout the remainder of the job's processing sequence. This priority scheme is a variant of the SRPT rule with remaining processing time being restricted to the workstation in question. To illustrate this priority scheme, the data from Table 8.20 is used and the sequencing priority for Workstation 1 developed. There are four uses of Workstation 1, these are: product 1-step 2, product 2-steps 1 and 4, and product 3-step 3. The remaining processing times in Workstation 1 for these four visits are 4, 10, 2, and 4, respectively. Thus, in step 4 first priority is given to jobs of product type 2. The tie between product 1-step 2 and product 3-step 3 is broken arbitrary in favor of product 3-step 3. Lastly, product 2-step 1 is processed. The sequences for the other two workstations are computed similarly.

The push-release system is operated by fixing the time between product releases and staying with that schedule regardless of the number of jobs in the system. For this problem, each job type has a desired throughput rate of 3 jobs per hour. Thus, to obtain this throughput rate on a long-term basis, the release rates need be the same as the desired output rates. Each type of job can either be released at this fixed rate, or a job released to the system at three times that rate and then assigned a type once active.

The results for the four sequencing algorithms are displayed in Table 8.21. The *WBAL* algorithm yields a mean flow time that is 52% shorter than the worst algorithm, *SRPT*, and 18% better than the next best algorithm, *SEPT*. In general, the standard deviations of the flow times follow the same order as the means, except that the *FIFO* rule yields the lowest, 5.5% lower than the *WBAL* method.

Table 8.21 Push release policy results for the four job sequencing algorithms with the mean and standard deviations of the job flow times and throughput rates as given; the results are the average of 10 replications of length 22,000 with a statistical reset at 2,000

Sequencing Rule	Mean Cycle Time	Std.Dev. Cycle Time	Total Throughput
WBAL	104.3 min	87.7 min	8.88/hr
SEPT	127.1 min	130.7 min	8.88/hr
FIFO	175.3 min	82.9 min	8.88/hr
SRPT	219.3 min	240.6 min	8.82/hr

For push-release control, the work-balance algorithm performs the best of the four algorithms tested. The best push result, however, can be improved on by limiting the total number of jobs allowed in the system (a *CONWIP* control strategy).

8.3.3 CONWIP Strategy Model

In this analysis, we use a slightly different form for CONWIP control. Instead of establishing a limit for each separate product type, only a limit on the total work-in-process will be used. Thus, the total number of jobs allowed to be actively processing at any time is called the *CONWIP* number. Once the allowed number of jobs is in the shop, new releases require the completion of a job within the system.

The *CONWIP* policy results for the four sequencing algorithms are displayed in Table 8.22. The *WBAL* algorithm yields a mean flow time under the *CONWIP* policy that is 41% shorter than the worst algorithm, *SRPT*, and 30% better than the *FIFO* algorithm, and 13.5% better than *SEPT*. Again the *FIFO* rule yields the lowest standard deviation with the *WBAL* method second. A job selection policy that should be avoided in the *CONWIP* environment is to enter the next job into the system with the same job type as the one that just completed. This approach, although seemingly consistent with the desired output proportions, can lead to preferential production of the faster processing job types particularly in conjunction with *SEPT* sequencing algorithm.

Table 8.22 *CONWIP* control policy results with a cyclic release policy and four job sequencing algorithms with mean and standard deviations of the job flow times and throughput rates as given; the results are the average of 10 replications of length 22,000 with a statistical reset at 2,000

Sequencing Rule	CONWIP Limit	Mean Cycle Time	Std.Dev. Cycle Time	Total Throughput
WBAL	12	80.4 min	54.3 min	8.94/hr
SEPT	14	93.0 min	67.9 min	9.00/hr
FIFO	17	114.8 min	37.4 min	8.88/hr
SRPT	20	136.4 min	110.1 min	8.76/hr

To illustrate this potential problem, consider the best *CONWIP* quantity of 14 under the *SEPT* rule and sequential job selection. These policies yield a mean throughput rate 9 per hour with a perfect balance between the three job types of 3/hr each. Using the same total *CONWIP* limit and a selection method of the entering job type to be the same as the one that just completed, the total throughput quantity is 9.66/hr, but the distribution of the throughput by job type is now much higher for job type one (4.62/hr, 2.4/hr, 2.64/hr). This result is due to the type one jobs having a shorter number of processing steps and also having relatively fast processing times. This gives job type ones a slight advantage in processing rate that tends to build over time. Of course, with the faster turn around rate for job type ones, the total throughput rate is above the objective value of 9/hr. Reducing the *CONWIP* limit from 14 to 8 jobs results in a reasonable total throughput rate of 9.06/hr, but the job type completion rate distribution is not balanced with the first type job having more than twice the throughput as the other two types. The mean flow time is considerably lower but the rate of completed jobs is nowhere near the required distribution. This throughput rate imbalance is due to the use of a single total CONWIP number. The imbalance could be alleviated by using the CONWIP control separately for each job type, but of course it is more complicated to implement and to analyze.

For all machine-sequencing algorithms used with the *CONWIP* control policy, it is necessary to obtain the *CONWIP* limit that yields the desired throughput rate. Since this is a single value for the policies being considered herein, this parameter can be searched rather easily. One method that is easy to implement is to start at a rather low *CONWIP* limit and incrementally increase this value until the desired throughput rate has been obtained. To expedite this process, a large increment can be used first, then when the desired rate has been exceeded, the increment can be reduced and the process continued. This reduction is repeated until the correct *CONWIP* limit has been found. Several of *WBAL* results are tabulated in Table 8.22. These results are used to illustrate the search process. Starting with a *CONWIP* limit of 5, simulation obtained a total throughput rate of 7.32/hr. Recall that the desired throughput rate is 8.94/hr. Then incrementing the limit first by 5 units, the next throughput rate of 8.7/hr is obtained. This result is low and thus, the limit is incremented by 5 units and the system is again evaluated via simulation. The *CONWIP* capacity limit of 15 jobs yields a total throughput rate of 9.18/hr. This result exceeds the desired rate and, therefore, the process returns to the last lower limit (10 units) and begins incrementing by 1 unit each simulation evaluation. It is known that the value lies between 10 and 15 and, therefore, no more than four more evaluations will be necessary. The throughput rate for control limit values of 11, 12, and 13 units are evaluated. The limit of 11 jobs is slightly low and 13 jobs is slightly high. The best (3-digit) approximation is at a control limit of 12 units (Table 8.23).

Table 8.23 Search study to find the *CONWIP* limit with *WBAL* sequencing rule that yields to desired 8.94/hr total throughput rate; the results are the average of 10 replications of length 22,000 with a statistical reset at 2,000

CONWIP by 5's	*CONWIP* by 1's	Total Throughput	Mean Cycle Time	Std.Dev. Cycle Time
5		0.122/min	40.95 min	22.32 min
10		0.145/min	68.95 min	43.98 min
	11	0.148/min	73.94 min	48.12 min
	12	0.149/min	79.99 min	53.02 min
	13	0.150/min	86.57 min	60.43 min
	14	0.153/min	91.33 min	64.30 min
15		0.153/min	97.79 min	70.05 min
20		0.157/min	126.7 min	97.28 min

Appendix

The Mean Value Analysis Algorithm (Property 8.3) and its modification for non-exponential times (Property 8.5) as well as the multiple product approximations (Property 8.8 and 8.9) are easily evaluated using Excel. However, the Marginal Distribution Analysis Algorithm (Property 8.4) is more complicated and is best handled in Excel with VBA as is the algorithm of Property 8.7. In this appendix, we give the Excel formula needed for the Mean Value Analysis Algorithm and leave the extensions to the reader. The material in the Appendix of Chap. 3 can be used to find the relative arrival rates (Property 8.1) of a network. Given the vector **r**, the following can be used to obtain the cycle times for the network of Example 8.2.

The initial data of the problem is established by the following.

	A	**B**	**C**	**D**	**E**
1		E[Ts-1]	E[Ts-2]	E[Ts-3]	
2		0.2	0.5	0.5	
3		r-1	r-2	r-3	
4		1	1	0.9	

We skip a row and then setup for the algorithm.

	A	**B**	**C**	**D**	**E**
6	w	CT-1	CT-2	CT-3	Sum
7	1	=B2	=C2	=D2	

In Cell A8, type `=A7+1` and then copy Cell A8 down through Cell A20. In Cell E7, type

```
=SUMPRODUCT(B7:D7,$B$4:$D$4)
```

and copy Cell E7 down through Cell E20. Finally, the main iterative step is typed into Cell B8 as

```
=B$2+B$2*$A7*B$4*B7/$E7
```

and then Cell B8 is copied to the right through Cell D8 and then B8 is copied down through Cell D20. It is important when typing the various formulas that care is taken to type the dollar signs (\$) exactly as shown since at times the row indicator must be an absolute address and sometimes the column indicator must be an absolute address. The resulting spreadsheet should give the Mean Value Algorithm through a CONWIP level of 14.

Problems

8.1. Find the relative flow rates for the network displayed in Fig. 8.4.

8.2. Find the relative flow rates for the network displayed in Fig. 8.6.

8.3. Re-consider the Example 8.2 and find the workstation and system performance measures for a CONWIP level of
(a) 7 jobs, and
(b) 10 jobs.

Fig. 8.4 Network flows for Problem 8.1

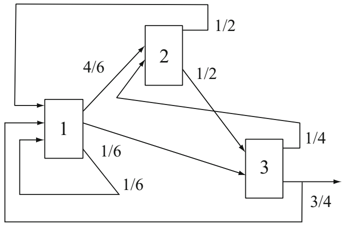
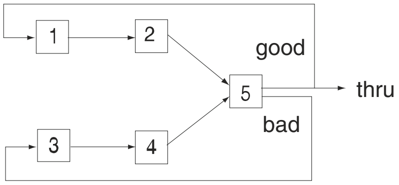


Fig. 8.5 Network flows for Problem 8.5



8.4. For the network of Problem 8.1, consider a CONWIP level of 5 jobs and assume that each workstation has only one processor. The means of the exponentially distributed service times at the three workstations are 15 minutes, 30 minutes, and 1 hour, respectively. Using the Mean Value Analysis Algorithm, find the expected cycle time in each workstation, the expected work-in-process in each workstation, the flow rate for each workstation, the total system throughput, and the system cycle time.

8.5. Consider the following closed queueing network made up of single server workstations with routing structure displayed in Fig. 8.5. Consider that each workstation has one machine with exponentially distributed processing times. Use the Mean Value Analysis Algorithm with $w_{\max} = 10$ to find the expected cycle time in each workstation, the expected work-in-process in each workstation, the mean throughput rate for each workstation, the total system throughput, and the system cycle time.

(a) Use the following data (based on an example in [4]):

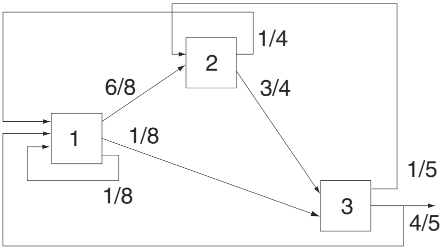
$$\begin{aligned} E[T_s] &= (1/2, 1/2, 1, 1, 1) \\ \Pr\{good\} &= 1/2, \\ \Pr\{bad\} &= 1/2. \end{aligned}$$

(b) Use the following data:

$$\begin{aligned} E[T_s] &= (1/3, 1, 3/2, 1/2, 2) \\ \Pr\{good\} &= 1/2, \\ \Pr\{bad\} &= 1/2. \end{aligned}$$

(c) Use the following data:

Fig. 8.6 Network flows for Problems 8.2 and 8.6



$$
\begin{aligned}
E[T_s] &= (1/3, 1, 3/2, 1/2, 2) \\
\Pr\{good\} &= 3/4, \\
\Pr\{bad\} &= 1/4.
\end{aligned}
$$

8.6. Consider the single product network model depicted in Fig. 8.6.
(a) Compute the relative flow rates $(r_1 = 1, r_2, r_3)$.
(b) Let the mean processing times be given by (4 hr, 2 hr, 3 hr) for the three workstations and assume that there is only one processor at each workstation. The total number of jobs kept in the system at all times is 10. Assuming that the cycle time estimates for the three workstations converge to 32.769 hr, 3.335 hr, and 6.421 hr, respectively, fill in the following information:

$$
\begin{array}{lll}
WIP_1 = ?, & WIP_2 = ?, & WIP_3 = 1.267 \\
\lambda_1 = ?, & \lambda_2 = 0.222, & \lambda_3 = 0.197 \\
WIP_s = ?, & th_s = ?, & CT_s = ?
\end{array}
$$

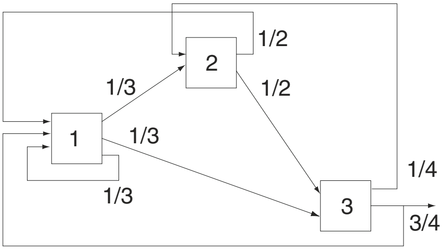
(c) Let Workstation 1 have two machines and re-compute the mean cycle time estimates for the three workstations as well as the mean throughput for the factory.

8.7. Consider the single product network model depicted in Fig. 8.7.
(a) Compute the relative flow rates $(r_1 = 1, r_2, r_3)$.
(b) Let the mean processing times be given by (2.5 hr, 3 hr, 5 hr) for the three workstations and assume that there is only one processor at each workstation. The total number of jobs kept in the system at all times is 8. Assuming that the cycle time estimates converge to (8.357 hr, 5.001 hr, 24.951 hr) for the three workstations, fill in the following information:

$$
\begin{array}{lll}
WIP_1 = ?, & WIP_2 = ?, & WIP_3 = 4.563 \\
\lambda_1 = ?, & \lambda_2 = 0.152, & \lambda_3 = 0.183 \\
WIP_s = ?, & th_s = ?, & CT_s = ?
\end{array}
$$

(c) Let Workstations 2 and 3 have two machines and re-compute the mean cycle time estimates for the three workstations as well as the mean throughput for the factory.

Fig. 8.7 Network flows for Problem 8.7



8.8. Resolve Problem 8.4, with an additional processor at Workstation 2.

8.9. Resolve Problem 8.5 (a) with 1, 2, 1, 1, 2 servers in the respective workstations.

8.10. Resolve Problem 8.3 except assume that the SCV for all service times is 0.25 with a CONWIP limit of 5 jobs.

8.11. Solve Problem 8.4 except assume that the processing times at the workstations have the following characteristics:

$$E[T_s] = (0.25, 0.50, 1.0) \text{ hr}$$
$$C_s^2 = (0.75, 1.25, 2.0) \ .$$

8.12. Consider a two product production facility with three, single-server workstations. The *WIP* limits for the products are 2 and 3 jobs. Assume that the processing times by product are exponentially distributed with mean times of

Product	$E[T_s(i,1)]$	$E[T_s(i,2)]$	$E[T_s(i,3)]$
$i = 1$	1.00 hr	2.00 hr	3.00 hr
$i = 2$	1.75 hr	2.50 hr	1.50 hr

The workstation transition probability matrices for the two products are:

$$P^1 = \begin{bmatrix} 1/5 & 3/5 & 1/5 \\ 1/5 & 1/5 & 3/5 \\ 2/5 & 2/5 & 1/5 \end{bmatrix} \quad \text{and } p^2 = \begin{bmatrix} 2/6 & 3/6 & 1/6 \\ 3/6 & 1/6 & 2/6 \\ 2/6 & 3/6 & 1/6 \end{bmatrix} .$$

(a) Determine the cycle times and *WIP*'s by product and workstation using the algorithm of Property 8.7.
(b) Compute the product and system performance measures given that the flow out of Workstation 3 back to Workstation 1 is considered good production for both products.

8.13. Consider Example 8.5 with two products containing routes according to Figs. 8.2a and 8.2b. Add a third product with its CONWIP level set at 7 jobs in this facility. The mean processing times (in hours) for the third product in the three workstations are $E[T_s(3,k)] = (0.6, 0.4, 0.5)$. The workstation transition probability matrix for this product is

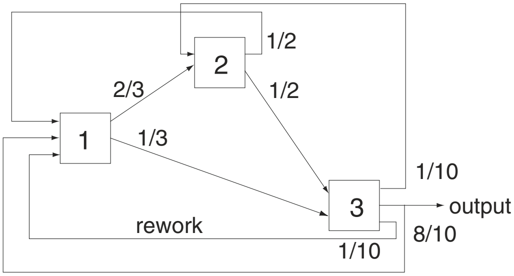


Fig. 8.8a Network flows for Product 1 of Problem 8.15

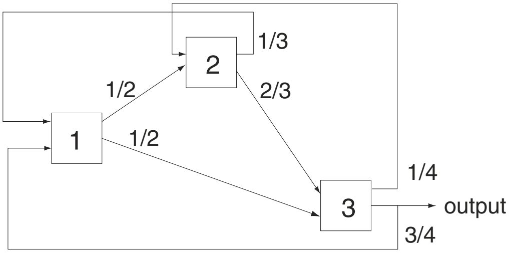


Fig. 8.8b Network flows for Product 2 of Problem 8.15

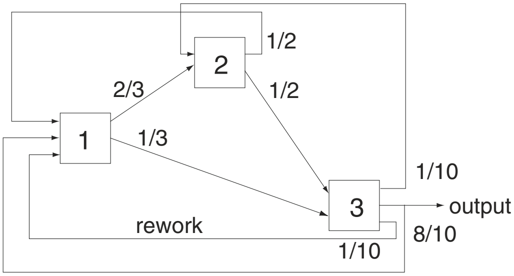
$$P^3 = \begin{bmatrix} 0 & 2/3 & 1/3 \\ 4/5 & 0 & 1/5 \\ 3/4 & 1/4 & 0 \end{bmatrix} .$$

Using the approximation algorithm of Property 8.8, determine the cycle times and WIP levels for each workstation by job type. Assume that the flow from Workstation 3 to Workstation 1 is good production and determine factory mean throughput and cycle time.

8.14. Reconsider the facility described in Problem 8.12 except let the WIP limits for the products be 8 and 10 jobs, respectively. (a) Determine the cycle times and WIP's by product and workstation using the algorithm of Property 8.8.
(b) Compute the product and system performance measures given that the flow out of Workstation 3 back to Workstation 1 is considered good production for both products.

8.15. A two-product factory is operated by releasing units only when a job is completed. The company policy is to maintain exactly 8 units of Product 1 and 5 units of Product 2 in the system at all times. So if a job of type i completes, another job of that type is immediately released into the factory. The two products have quite different processing sequences and times with routing structures as displayed in Figs. 8.8a and 8.8b . All processing times are exponentially distributed. The mean processing times by workstation and product are

Fig. 8.9a Network flows for Product 1 of Problem 8.16



Product	$E[T_s(i,1)]$	$E[T_s(i,2)]$	$E[T_s(i,3)]$
$i=1$	2.0 hr	3.0 hr	1.0 hr
$i=2$	1.5 hr	2.0 hr	3.50 hr

Using the provided data, obtain the following answers using the algorithm of Property 8.8.

(a) Find the relative arrival rates to the workstations.

(b) Write the equations for the workstation cycle times.

(c) Assume that the workstation cycle times are

	WS 1	WS 2	WS 3
CT_k^1	8.543 hr	16.855 hr	8.086 hr
CT_k^2	8.275 hr	16.439 hr	9.681 hr

and complete the following tables.

		WS 1	WS 2	WS 3
Product 1	WIP_k^1	2.566	3.730	1.704
	λ_k^1	?	0.221/hr	0.211/hr
Product 2	WIP_k^2	?	2.036	1.598
	λ_k^2	0.165/hr	0.124/hr	0.165/hr

(d) Give the workstation utilization factors.

(e) Give the system cycle times and throughputs for the two products.

8.16. A two-product factory is operated by releasing units only when a job is completed. The company policy is to maintain exactly 6 units of each product type in the system at all times. So if a job of type *i* completes, another job of that type is immediately released into the factory. The two products have quite different processing sequences and times with routing structures as displayed in Figs. 8.9a and 8.9b . All processing times are exponentially distributed. The mean processing times by workstation and product are

Product	$E[T_s(i,1)]$	$E[T_s(i,2)]$	$E[T_s(i,3)]$
$i=1$	15 min	30 min	45 min
$i=2$	60 min	42 min	24 min

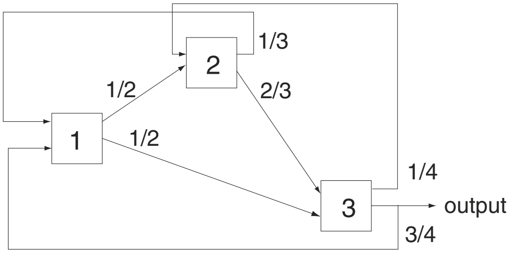


Fig. 8.9b Network flows for Product 2 of Problem 8.16

Using the provided data, obtain the following answers.
(a) Verify the following relative arrival rates.

Product	$r_{i,1}$	$r_{i,2}$	$r_{i,3}$
$i = 1$	1	0.737	0.702
$i = 2$	1	0.5	1.1111

(b) Using the algorithm of Property 8.7, determine the workstation cycle times if the CONWIP levels were set to $\mathbf{w}_{\max} = (1,1)$.
(c) Using the algorithm of Property 8.8, determine the workstation cycle times and workstation *WIP* levels by product for the CONWIP levels of $\mathbf{w}_{\max} = (6,6)$.
(d) Determine the workstation utilization factors.
(e) Determine the system performance measures.

8.17. Resolve Example 8.6 except assume that the service time distribution has a Gamma distribution with shape parameter $\alpha = 2$ (i.e., an Erlang-2 distribution) and with mean values as specified by Table 8.9.

References

1. Askin, R.G., and Standridge, C.R. (1993). *Modeling and Analysis of Manufacturing Systems*. John Wiley & Sons, New York.
2. Buzacott, J.A., and Shanthikumar, G.J. (1993). *Stochastic Models of Manufacturing Systems*. Prentice Hall, Englewood Cliffs, N. Y.
3. Curry, G.L., Deuermeyer, B.L., and Feldman, R.M. (1989). *Discrete Simulation: Fundamentals and Microcomputer Support*. Holden-Day, Inc., Oakland, CA.
4. Diagle, J.N. (1992). *Queueing Theory for Telecommunications*. Addison-Wesley Publishing Co., Reading, Mass.
5. Duenyas, I. (1994). A Simple Release Policy for Networks of Queues with Controllable Inputs, *Operations Research*, **42**:1162–1171.
6. Gross, D., and Harris, C.M. (1998). *Fundamentals of Queueing Theory*, Third Edition, John Wiley & Sons, New York.
7. Harrison, J.M., and Wein, L.M. (1990). Scheduling Networks of Queues: Heavy Traffic Analysis of a Two-Station Closed Network, *Operations Research*, **38**:1052–1064.
8. Hopp, W.J., and Spearman M.L. (1996). *Factory Physics: Foundations of Manufacturing Management*. Irwin, Chicago.

9. Marie, R.A. (1979). An Approximate Analytical Method for General Queueing Networks. *IEEE Transactions on Software Engineering*, **5**:530–538.
10. Marie, R.A. (1980). Calculating Equilibrium Probabilities for $\lambda(n)/C_k/1/N$ Queues. *Performance Evaluation Review*, **9**:117–125.
11. Reiser, M., and Lavenberg, S.S. (1980). Mean-Value Analysis of Closed Multichain Queuing Networks. *J. Association for Computing Machinery*, **27**:313–322.
12. Spearman, M.L., Woodruff, D.L., and Hopp, W.J. (1990). CONWIP: A Pull Alternative to Kanban, *International Journal of Production Research*, **28**:879–894.
13. Wein, L.M. (1990). Scheduling Networks of Queues: Heavy Traffic Analysis of a Two-Station Closed Network with Controllable Inputs, *Operations Research*, **38**:1065–1078.
14. Wein, L.M. (1992). Scheduling Networks of Queues: Heavy Traffic Analysis of a Multistation Network with Controllable Inputs, *Operations Research*, **40**:S312–S344.

Chapter 9
Serial Limited Buffer Models

Limited buffer capacity models can be used for the mathematical representations of a form of kanban control. There are two aspects of limited buffer systems studied in this chapter. First an approach for developing an analytical approximation model for serial flow systems is developed. Then the issue of how these buffer values can be set to yield an optimal system configuration is addressed.

The systems considered here consist of a set of workstations that have limits on the number of work-in-process units allowed to wait at each of the single-server processing stations. For serial systems, these workstations are connected in a serial configuration so that jobs flow from the first to the second workstation only, and then from the second to the third workstation, etc., until they exit the facility. Thus, all jobs have the same routing sequence. The workstations have a set number of jobs that are allowed into the workstation simultaneously and these limits need not be identical. Let w_k represent the work-in-process capacity limitation for Workstation k. This is the total number of jobs allowed in the workstation including the job being processed. Only single-server machines in each workstation are considered; the complexities of multiple servers in a limited-buffer capacity model is beyond the scope of this analysis. (Thus, Workstation k will process jobs on Machine k so the terms Workstation k and Machine k will be used interchangeably.)

There are several methods of operating a WIP-limiting control strategy. The major policy is that a job may not proceed to the next workstation until a space becomes available in that workstation. However, there are several ways the workstation can operate. The concept of process blocking after job completion is generally used in analytical models. That is, when a job is finished, it may not be removed from the machine until space is available in the next workstation for this job. This effectively blocks the machine from processing other jobs in its queue and is called *blocking after service*. Another variation is blocking before service, that is the machine cannot process the job until it has the authorization to move to the next workstation. A control procedure frequently used in practice processes queued jobs in the workstation until they all have been processed and the machine is forced to be idle due to the lack of unfinished inventory. In this chapter, only the blocking after processing strategy is implemented for modeling purposes. This strategy allows for the imme-

G.L. Curry, R.M. Feldman, *Manufacturing Systems Modeling and Analysis*, 2nd ed.,
DOI 10.1007/978-3-642-16618-1_9, © Springer-Verlag Berlin Heidelberg 2011

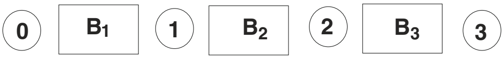


Fig. 9.1 Network structure for the kanban analysis

diate response of the system to congestion and does not delay the response until several more jobs have been processed.

This chapter deals with a finite *WIP* control approach where the limits are placed on the number of jobs allowed in each workstation rather than in the factory as a whole as was done with the *CONWIP* approach of Chap. 8. The general approach is to develop approximate probability distributions for the number of jobs in each workstation (somewhat) independently and then connect these to estimate factory performance. To facilitate the individual workstation models, general processing time distributions are approximated by easy to model exponential phases while maintaining the first two moments of the general service distributions. By assuming that all distributions to be modeled have SCV's greater than or equal to 1/2, Coxian (GE_2) process sub-models can be used and tractable steady-state queueing models result. An approximation methodology is developed for serially connected systems with finite buffers at each workstation. The methods of this chapter utilize a decomposition approach that make the resulting models computationally tractable.

9.1 The Decomposition Approach Used for Kanban Systems

The system being modeled is a series of workstations, or machines, connected by buffer spaces of varying capacities. Job releases into the facility are controlled by an initial machine with an unlimited backlog that continuously processes jobs and sends them into the first workstation as long as there is space for that job. When the job cannot proceed into the first workstation, the capacity there being full, then this "job release" machine is blocked using the same "blocked after processing rule" as all "real" workstation machines. The pre-release jobs are not considered as actual jobs and do not count as facility *WIP*. This initial process can be thought of as the preparation time necessary for a job release. Figure 9.1 illustrates the serial network structure being studied, where Machine 0 is a machine representing job releases to the system and there is a buffer of finite capacity between each machine. It is possible that job releases are simply due to an individual processing the order so "machine" may be a misnomer, but it is used simply for ease of reference.

The system can be modeled by developing the steady-state equations defining the proportion of the time that the system is in every possible state. This direct full scale modeling approach gets into computational difficulty very quickly because the number of states that have to be considered grows exponentially with the number of serial workstations. For example, if there can be 5 jobs in each workstation and there are 4 workstations in series, then each workstation would have states $0,\cdots,5$, and the total number of states necessary to model the network is 6^4 =1296; whereas

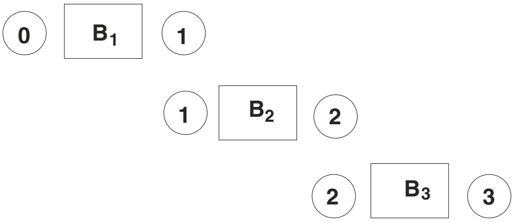


Fig. 9.2 Two-node decomposition of the serial system of Fig. 9.1 where each interior machine in the serial list serves as the arrival-machine in one subsystem and the service-machine in the next subsystem

there would be 60,466,176 states with 10 workstations. To overcome this explosive growth in the modeling representation, a decomposition approach is generally taken. In the decomposition approach, an attempt is made to isolate each workstation and obtain its steady-state probability distribution based on inflow and service distribution parameters. These parameters reflect the interaction between the workstations, and very good approximations to the system performance measures can often be obtained.

Most decomposition approaches isolate a single workstation at a time, but certainly the approach could utilize modeling pairs of workstations at a time or any computationally tractable number. The standard approach is one workstation at a time, that is the approach taken here. The popular approach, but certainly not the only modeling view, is to create subsystems composed of a workstation buffer and two servers. An upstream server is used to depict the time between job arrivals to the workstation and a down-stream server represents the workstation processing machine. Then as the analysis proceeds, each machine will play the role in one subsystem of the processing server and in the next downstream subsystem as the arrival-generating machine. There will actually be two distinct service distributions for each machine because of the distinction between these two roles. This is called a two-node decomposition approach. This two-node decomposition approach is used in many research papers and the books by Perros [13] and Altiok [1]. The modeling decomposition representation is illustrated in Fig. 9.2.

For discussion purposes, subsystems are numbered from left to right, such as Subsystem 1, 2, etc. up to the last subsystem represented by n. Each Subsystem k has an upstream server, denoted as Machine $(k-1)$ and a downstream server denoted as Machine k, with a buffer space for waiting jobs in between. The buffer and downstream machine correspond to Workstation k of the facility being modeled. The buffer-downstream machine combination has limit of w_k jobs. Note that a job being processed by the upstream (left-side) server is not counted against this limit because in reality it is still in the previous workstation. Note that the first workstation's upstream server (Machine 0) may not actually exist. The system's job generating or external inter-arrival mechanism is incorporated into the model and for notational convenience is labeled as Machine 0.

Now consider Subsystem k, where Machine k is processing jobs entered into the buffer by Machine $k-1$. In actuality, Machine k can be blocked after service because the buffer for Machine $k+1$ is full and it can also be starved (forced idle for the lack of jobs) because of upstream workstation behavior. The blocking of Machine k cannot be mimicked in the decomposition based on knowledge of the subsystem's state-probability distribution because of the disconnection between workstations in the decomposition methodology. Thus, to account for this potential delay, an appropriate delay as part of the job processing time is incorporated. Therefore, in the decomposition structure, the processing time at a server will be longer than the nominal processing time. This time will incorporate the probability of blocking due to the next subsystem being full and an appropriate delay time for completing processing at this server to relieve the blockage. Note that the downstream server in turn could be blocked by its downstream subsystem being full and also be forced to wait for a process completion there, and so forth all the way downstream until the last server has been reached. The last server in the serial system can never be blocked. It is assumed that completed jobs are immediately transported to shipping, or storage, etc., and hence leave the system immediately.

The blocking of an upstream server in a subsystem can be properly accounted for from the known probability distribution for the subsystem states. Hence, the upstream server's processing time does not have to account for downstream blocking. It will, however, be left periodically without a job to process (this situation is referred to as the machine being starved). The additional delay until a job becomes available for the upstream server to process must be incorporated into the upstream machine's processing time. This delay occurs with a known probability based on the upstream subsystem's steady-state probability distribution and the associated delay time is the remaining processing time for the current job on that machine. However, in turn, this workstation could also be starved and forced to wait on its upstream server, and so forth all the way back to the job generating Machine 0. This first machine can never be starved.

This discussion hopefully has instilled a feeling for the differences in the two service times for a given machine when it is playing the role of either the upstream or the downstream server in the decomposition procedure. One of the main tasks in the implementation of this decomposition procedure is the estimation of these two distinct service distributions for each machine. These processes are discussed in the next section.

9.2 Modeling the Two-Node Subsystem

The first aspect of modeling each subsystem is to describe each server within the two-machine subsystem. Initially, the machines will be modeled as exponential processes, then modeled by a mixture of generalized Erlangs with two or three phases (described in the next subsection), and finally approximated by a two-phase generalized Erlang. Once the machines are described, a state space will be developed using the general approach of Sect. 3.6.

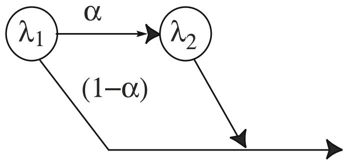


Fig. 9.3 A generalized Erlang with two phases (GE_2), where the first phase always occurs and has a mean rate λ_1 and the second phase occurs with probability α and has a mean rate λ_2

9.2.1 Modeling the Service Distribution

In the initial step for modeling the limited buffer subsystems, the model for a finite capacity exponential queueing process is needed. Such a system was previously analyzed as an example in Chap. 3 (see Eqs. 3.5 and 3.7) so we will not repeat the specific steps; however, the general solution is given in the following property so that it can be easily referenced when needed.

Property 9.1. *Consider a single-server queueing system with arrivals according to a Poisson process having mean rate* λ *and an exponential service time with mean* $1/\mu$. *The system can have at most* $w_{\max}$ *jobs in the system, counting all jobs in the queue plus the one in service. The probability distribution describing the number of jobs in the system in steady-state is*

$$p_i = \frac{(\lambda/\mu)^i}{\sum_{j=0}^{w_{\max}} (\lambda/\mu)^j} \ \textit{for}\ i = 0, \cdots, w_{\max} \, .$$

Each processing time is assumed to be exponentially distributed; however, because of the possibilities of blockage or starvation, the actual delay time cannot be modeled using the exponential distribution. For purposes of modeling the delay times within each server, we review some of the material of Sect. 3.6.3 where the generalized Erlang (GE) distribution was introduced. Figure 9.3 presents a graphical representation of the GE_2 distribution, where $1/\lambda_1$ is the mean time spent in the first phase, α is the probability that the second phase will be visited, $1-\alpha$ is the probability that only the first phase will be used, and $1/\lambda_2$ is the mean time spent in the second phase if it is visited. The GE_2 distribution is used because it is very versatile, being able to fit a distribution to any positive mean and any SCV greater than or equal to 1/2. For a given mean, $E[X]$, and SCV, $C^2[X]$, the following can be used to find the parameters of a GE_2 distribution [2, p. 54–56]:

If $C^2[X] > 1$,

$$\lambda_1 = \frac{2}{E[X]} \, , \qquad \lambda_2 = \frac{1}{E[X]C^2[X]} \, , \qquad \alpha = \frac{1}{2C^2[X]} \, ; \tag{9.1}$$

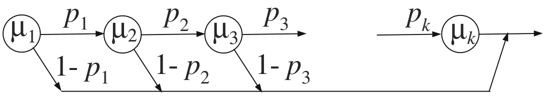


Fig. 9.4 A mixture of generalized Erlang distributions (MGE_k) with k-phases

and if $\frac{1}{2} \le C^2[X] \le 1$,

$$\lambda_1 = \frac{1}{E[X]C^2[X]} , \quad \lambda_2 = \frac{2}{E[X]} , \quad \alpha = 2(1 - C^2[X]) . \tag{9.2}$$

To represent a specific GE_2 distribution, we list its parameters as a three-tuple giving the rate of the first phase, then the probability associated with moving to the second phase and finally the rate of the second phase. Thus, the distribution of Fig. 9.3 is said to be a $(\lambda_1, \alpha, \lambda_2)$ GE_2 distribution.

A generalization of the *GE* distribution is a *mixture* of generalized Erlangs (*MGE*). Consider the diagram in Fig. 9.4. The diagram shows a *GE* distribution with k phases; however, we add to the fact that it is a mixture so that the process does not necessarily start at the first phase; thus, in addition to the parameters shown in the diagram, there is a vector of probabilities, denoted by $\boldsymbol{\alpha}$, that represents the starting phase; thus, α_i denotes the probability that the process will start in Phase i and then the process will proceed through the phases always going to the right or exiting the system.

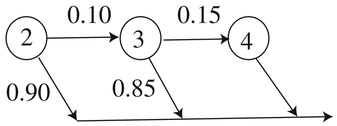
One method of describing an *MGE* process is to construct a so-called generator matrix for the process which is a matrix giving the transition rates of moving from state to state once the process starts. For example, the generator for the process of Fig. 9.4 is

$$G = \begin{bmatrix} -\mu_1 & p_1\mu_1 & 0 & & 0 \\ 0 & -\mu_2 & p_2\mu_2 & & 0 \\ 0 & 0 & -\mu_3 & & 0 \\ & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -\mu_k \end{bmatrix} . \tag{9.3}$$

For a generator matrix, the diagonal elements are always negative, since they represent leaving the state and the off-diagonal elements are always non-negative. The off-diagonal elements of a row must sum to a value less than or equal to the absolute value of the diagonal element and the difference between the absolute value of the diagonal element and the row sum of the off-diagonal elements is the rate at which the process terminates from that particular state.

The *MGE* distribution is an example of a more general type of distribution know as phase type distributions. These were popularized by Neuts and a relatively complete description of their use within a queueing context can be found in [11]. The moments of phase type distributions, and thus *MGE* distributions, are easily determined according to the following property.

Fig. 9.5 The MGE_3 service time process for Example 9.1



Property 9.2. *The moments for a random variable, T, having a phase-type distribution with generator G and initial probability vector $\boldsymbol{\alpha}$ are given by*

$$\begin{aligned} E[T] &= -\boldsymbol{\alpha} G^{-1}\mathbf{1} \\ E[T^2] &= 2\boldsymbol{\alpha} G^{-2}\mathbf{1} \\ E[T^3] &= -6\boldsymbol{\alpha} G^{-3}\mathbf{1}\,, \end{aligned}$$

where $\boldsymbol{\alpha}$ is a row vector, $\mathbf{1}$ is a column vector of all ones, and $G^{-n} = (G^{-1})^n$.

Example 9.1. Assume that the processing time for a workstation is described by an MGE_3 process where the mean sojourn times for the three phases are 1/2 hr, 1/3 hr, and 1/4 hr, respectively. Furthermore, there is a 90% chance that if the service starts in the first phase it will be finished after the first phase, and an 85% chance that if the process makes it to the second phase, that it will be finished after that phase (Fig. 9.5). In addition, there is a 90% probability that the process will start in the first phase and there is a 10% that it will start in the second phase. For this process, the generator matrix is

$$G = \begin{bmatrix} -2 & 0.2 & 0 \\ 0 & -3 & 0.45 \\ 0 & 0 & -4 \end{bmatrix}.$$

and $\boldsymbol{\alpha} = (0.9, 0.1, 0)$. To obtain the moments for the process, the inverse of the generator is needed, and this is

$$G^{-1} = \begin{bmatrix} -0.5 & -0.03333 & -0.00375 \\ 0 & -0.03333 & -0.00375 \\ 0 & 0 & -0.25 \end{bmatrix}.$$

Property 9.2 yields a mean of $E[T] = 0.5205$ hr and a second moment of $E[T^2] = 0.5339$ hr^2; thus, the SCV is $C^2[T] = 0.971$.

To approximate the MGE_3 distribution with a GE_2 distribution, Eq. (9.2) is used to fit the moments. This yields a (1.9792, 0.0586, 3.8425) GE_2 distribution. In other words, the approximation always starts in the first phase having a mean rate of 1.9792/hr, then with probability 5.86% it will enter a second phase having a rate of 3.8425/hr and with probability 94.14% it will finish after the first phase. □

- *Suggestion: Do Problems 9.1–9.3.*

9.2.2 Structure of the State-Space

Each subsystem of the serial decomposition consists of an arrival generating machine (called the arrival-machine), a workstation processing machine (called the service-machine), and a finite buffer of capacity $w_{\max} - 1$ jobs in between the two machines. A job in the service-machine counts as part of the work-in-process so the subsystem has a capacity of $w_{\max}$ jobs. The job being processed by the arrival-machine does not count against the subsystem capacity limit because the job being served there is physically located in the previous workstation. The intent of this section is the development of a queueing model of the steady-state occupancy probabilities for the subsystem. Each service mechanism will be modeled as a GE_2 distribution.

Since a job is assumed to be always available at the arrival-machine, the machine itself will either be processing a job in its first phase (remember, the machine is considered to be a GE_2 system), processing a job in its second phase, or be finished processing the job but the job is blocked because there is no room in the buffer. For modeling purposes, it is necessary to keep track of the arrival-machine status (i.e., either identify phase of processing or show the machine blocked), the service-machine status (either identify phase of processing or show the machine idle), and the number of jobs in the subsystem. Thus, a 3-tuple of information is needed to represent the subsystem status. The continuous existence of a unit in the arrival-machine does not match up with reality for the associated machine. The modeling approach, however, is to account for the idle time for this real machine in the processing time for the arrival-machine. Thus, this machine should be thought of as the delay time between appearances of a job (inter-arrival time) to the workstation under consideration. When the actual predecessor machine is idle, this time is part of the inter-arrival time for the arrival-machine.

The 3-tuple state indicator is a vector with the first element representing the status of the first node (arrival-machine), the second element defines the status of the service-machine, and the third element is the total number of jobs in the subsystem. As always, if at least one job is available for processing, the service-machine will be processing (not idle). Thus, the 3-tuple subsystem status vector is of the form

$$(a, s, w)$$

where the states for a are Phase 1, Phase 2 or completed processing but blocked denoted by $a \in \{1, 2, b\}$. The states for s are similarly Phase 1, Phase 2, or idle denoted by $s \in \{0, 1, 2\}$, and the states for the third element of the three-tuple (work-in-process) are $w \in \{0, 1, \cdots, w_{\max}\}$. Different subsystems are denoted by indexing the 3-tuple elements by the subsystem index k as in $(a_k, s_k, w_{\max,k})$.

For each state where the machines are fully operational, there are 4 states associated with each fixed work-in-process level. That is, since each machine can be in one of two states, there are four combinations resulting: $(1, 1, w)$, $(1, 2, w)$, $(2, 1, w)$, $(2, 2, w)$, for $0 < w < w_{\max}$. For the situation where the arrival-machine is blocked, the buffer must be full and the service-machine must be busy; therefore, the possi-

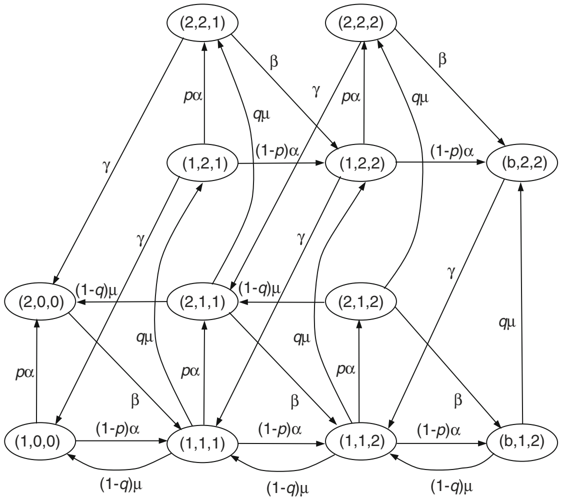


Fig. 9.6 Rate diagram for a two-node submodel with MGE_2 distributions: arrival node parameters (α, p, β) and service node parameters (μ, q, γ)

ble states are $(b, 1, w_{\max})$ and $(b, 2, w_{\max})$. Finally, if the subsystem is empty, then the arrival-machine cannot be blocked and will be in one of its two phases while the service-machine will be idle resulting again in two possible states: $(1,0,0)$ and $(2,0,0)$. Thus, there are a total of $4(w_{\max}+1)$ possible states for any subsystem capacity limitation of $w_{\max} \geq 1$. For example, a subsystem with a capacity limit of 2 units will have a state space of 12 possible states. These twelve states, by inventory level, are:

$$\begin{gathered}(1,0,0),(2,0,0),\\(1,1,1),(1,2,1),(2,1,1),(2,2,1),\\(1,1,2),(1,2,2),(2,1,2),(2,2,2),\\(b,1,2),(b,2,2).\end{gathered}$$

The movement of the subsystem from state to state is limited to adjacent inventory levels because of the single unit machine processing assumptions (that is, no batch arrivals or services are allowed). Figure 9.6 displays this 12 state example and associated flow rates. In Fig. 9.6, the arrival node's GE_2 distribution has parameters (α, p, β) and the service-machine's GE_2 distribution parameters are (μ, q, γ). As the number of units allowed in the subsystem increases, the diagram has more columns but the structure remains as illustrated.

9.2.3 Generator Matrix Relating System Probabilities

The steady-state probabilities for the subsystem states, v_i, are determined by solving the system of equations relating the flows between states. Here the index i represents a 3-tuple (a, s, w); a is the status of the arrival-machine, s is the status of the service-machine, and w is the number of jobs present. The steady-state equations relating these states are developed by equating the in-flow into any state with the out-flow from that state. (This procedure for obtaining the equations is called the isolation method in Sect. 3.6.2). Taking the states one at a time, a system of balance equations is derived. For the example illustrated in Fig. 9.6 there are twelve such equations (in 12 unknowns):

$$\begin{aligned}
\alpha\, v_{(1,0,0)} &= (1-q)\mu\, v_{(1,1,1)} + \gamma v_{(1,2,1)} \\
\beta\, v_{(2,0,0)} &= (1-q)\mu\, v_{(2,1,1)} + p\alpha\, v_{(1,0,0)} + \gamma v_{(2,2,1)}
\end{aligned}$$

$$\begin{aligned}
(\alpha+\mu)\, v_{(1,1,1)} &= (1-p)\alpha\, v_{(1,0,0)} + \beta\, v_{(2,0,0)} + \gamma v_{(1,2,2)} \\
&\quad + (1-q)\mu\, v_{(1,1,2)} \\
(\beta+\mu)\, v_{(2,1,1)} &= \gamma v_{(2,2,2)} + (1-q)\mu\, v_{(2,1,2)} + p\alpha\, v_{(1,1,1)} \\
(\alpha+\gamma)\, v_{(1,2,1)} &= q\mu\, v_{(1,1,1)} \\
(\beta+\gamma)\, v_{(2,2,1)} &= p\alpha\, v_{(1,2,1)} + q\mu\, v_{(2,1,1)} \\
(\alpha+\mu)\, v_{(1,1,2)} &= (1-p)\alpha\, v_{(1,1,1)} + \beta\, v_{(2,1,1)} + \gamma v_{(b,2,2)} \\
&\quad + (1-q)\mu\, v_{(b,1,2)} \\
(\beta+\mu)\, v_{(2,1,2)} &= p\alpha\, v_{(1,1,2)} \\
(\alpha+\gamma)\, v_{(1,2,2)} &= q\mu\, v_{(1,1,2)} + (1-p)\alpha\, v_{(1,2,1)} + \beta\, v_{(2,2,1)} \\
(\beta+\gamma)\, v_{(2,2,2)} &= p\alpha\, v_{(1,2,2)} + q\mu\, v_{(2,1,2)} \\
\gamma v_{(b,2,2)} &= \beta\, v_{(2,2,2)} + (1-p)\alpha\, v_{(1,2,2)} + q\mu\, v_{(b,1,2)} \\
\mu\, v_{(b,1,2)} &= \beta\, v_{(2,1,2)} + (1-p)\alpha\, v_{(1,1,2)}
\end{aligned}$$

To form the generator matrix for this system, the left hand coefficients will be the negative of the diagonal elements and the coefficients on the right-hand side will be the off-diagonal elements. The resulting generator is shown in Fig. 9.7, where blanks represent zeros. Since the v_i values for $i = 1, \cdots, 12$ must form a probability mass function, the norming equation (i.e., $\sum_i v_i = 1$) must be used also. Thus, the steady-state probabilities can be found by the following property.

Property 9.3. *Consider a process described by a generator matrix Q such that the sum of the off-diagonal elements of each row equals the absolute value of the diagonal element of that row. If the row vector* $\mathbf{v}$ *satisfies*

(1,0,0)	(2,0,0)	(1,1,1)	(1,2,1)	(2,1,1)	(2,2,1)	(1,1,2)	(1,2,2)	(2,1,2)	(2,2,2)	(b,1,2)	(b,2,2)
$-\alpha$	$p\,\alpha$	$(1-p)\,\alpha$									
	$-\beta$	β									
$(1-q)\mu$		$-(\alpha+\mu)$	$q\mu$	$p\alpha$		$(1-p)\alpha$					
γ			$-(\alpha+\gamma)$		$p\alpha$		$(1-p)\alpha$				
	$(1-q)\mu$			$-(\beta+\mu)$	$q\mu$	β					
	γ				$-(\beta+\gamma)$		β				
		$(1-q)\mu$				$-(\alpha+\mu)$	$q\mu$	$p\alpha$		$(1-p)\,\alpha$	
		γ					$-(\alpha+\gamma)$		$p\alpha$		$(1-p)\,\alpha$
				$(1-q)\mu$				$-(\beta+\mu)$	$q\mu$	β	
				γ					$-(\beta+\gamma)$		β
						$(1-q)\mu$				$-\mu$	$q\mu$
						γ					$-\gamma$

Fig. 9.7 *Q*-generator matrix associated with the rate diagram of Fig. 9.6

$$\mathbf{v}Q = \mathbf{0}$$
$$\sum_i v_i = 1\,,$$

then v_i denotes the steady-state probability of the process being in state i.

Notice that if the sum of the off-diagonal elements were less than the absolute value of the diagonal element, then the process would terminate after some period of time and no steady-state would exist. There is also one redundant equation within the system defined by $\mathbf{v}Q = \mathbf{0}$ so that to obtain the steady-state probabilities, one of the columns (it does not matter which one) from the generator must be deleted.

- *Suggestion: Do Problems 9.4 and 9.5.*

9.2.4 Connecting the Subsystems

Recall that in the decomposition procedure, the upstream and downstream processing times must be adjusted to account for machine starvation and machine blockage, respectively. The two-node submodel can account for blocking by blocking the arrival-machine in the submodel. The arrival-machine in the submodel cannot, however, be starved due to the structure of the submodel. But this machine in the real system can be starved. A similar situation exists for blocking of the downstream or subsystem service-machine. Hence for the decomposition method to give reasonable results, these elements of the problem are accounted for in the delay time associated with their respective services. The decomposition approach is to de-couple the subsystems as much as possible, and this is accomplished by approximating the subsystem interactions as probabilistically independent events. So the probability of being blocked by the downstream system is taken as the steady-state probability of that subsystem being full.

Since the behavior of each subsystem is a function of the behavior of its neighboring subsystems, these subsystems all need to be solved simultaneously. This of course, somewhat negates the concept of decomposing the problem into subsystems; however, an iterative solution can be structured where the previous iteration subsystem values are used to estimate the interactions of the current subsystem with its neighbors and the estimates improve with each iteration. This computational approach is the crux of the decomposition solution method. A variety of iteration schemes have been utilized in various decomposition approaches for problems of this nature and are summarized in the paper by Dallery and Frein [4].

The general approach for obtaining a solution to the decomposed subsystems is to initially set the service-time distribution to the nominal service-time distribution and the arrival generating process-time distribution to the predecessor nominal service-time distribution for each subsystem. Then starting with the first subsystem, the subsystems are solved sequentially in increasing order. This allows for succeeding subsystems to estimate the probability of starvation from previously analyzed subsystem's results. Note that on this forward pass, the downstream blockage probabilities are not improved and only the arrival generating service distributions are improved. Then a backward pass through the subsystems is performed, starting at the end subsystem and working backward to the first subsystem. This process allows for improved processing times for the machines because the previous subsystem blocking probabilities and associated processing times have been updated. After both the forward and backward passes have been completed (called an iteration) the two subsystem process-time distributions have been updated. This iterative process is repeated until convergence of the distributions occur. For the single parameter exponential service-time distribution the iterative solution scheme is a contraction and, hence, converges [4].

The blocking and starvation probabilities are not based on steady-state values but on the probabilities at the instance of a service completion and the instance of an arrival, respectively. The blocking probability of a completed job in a subsystem is equal to the probability that an arrival for the downstream subsystem finds that subsystem full at the instance of the arrival occurrence. Thus, that subsystem cannot be in the blocked state at that time (or the arrival would not have occurred because the blocked state means that the arrival process is temporally shut-off). This blocking probability is computed as the ratio of the subsystem full states probabilities multiplied by the arrival completion rates for the full states divided by the sum of all allowable state probabilities (this excludes the two blocked states) times their respective arrival completion rates. A similar computation is required to compute the starvation probability for the arrival-machine at the subsystem under consideration. This probability is based on the upstream-subsystem machine at completion of a service finding that subsystem empty. Thus, the next service time, the inter-arrival time for this subsystem, will include a delay associated with that subsystem waiting for an arrival before processing can commence. The probability that the upstream subsystem is empty at the instance of the departure of a job just completing ser-

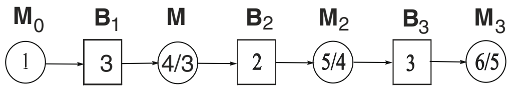


Fig. 9.8 Diagram representation of the example problem of Sect. 9.3 where circles represent machines with mean times listed and boxes represent *WIP* buffer areas with capacities listed

vice is computed based on the service completion rates for the various upstream subsystem states excluding the empty states (no service event can occur while that subsystem is empty and, therefore, not serving a job). The starvation probability for the subsystem in question is then the ratio of the sum of the probabilities that this subsystem has only one job in it times the job completion rates for those states and the sum of all nonzero states times their respective service-completion rates.

A detailed example is used to illustrate this iteration solution scheme. This simple three workstation serial system with exponential service time distributions converges to an acceptable accuracy level in 5 iterations. The errors as compared with a single long simulation run are in the neighborhood of 1% for the three performance measures of cycle time, throughput and work-in-process.

9.3 Example of a Kanban Serial System

A kanban system has been implemented within a facility that has a simple serial structure for its three workstations. The kanban limits are 4, 3, and 4, respectively, at the three workstations. All three workstations are single machine systems with exponential processing times have means of 80 min, 75 min, and 72 min, respectively. When there is room to begin a new job in the first workstation (i.e., when a new job is pulled into the factory), it takes an average of 60 minutes to gather the raw material and do the data entry necessary to begin the job within the factory. Notice that in Fig. 9.8 illustrating this factory, the capacity of the buffers is one less than the kanban limits due to a job possibly being in process.

Since the factory is using a kanban control system, each machine is blocked from starting service on a new job until the job just completed obtains space in the next workstation. The first or job arrival generating machine is always busy except when it has a completed job that is blocked from entering the first workstation. This situation blocks the machine from starting on a new job. Completed jobs at the last workstation leave the system with no blockage or delays.

Throughout the discussion of the algorithm, it will be important to keep in mind the distinction between the arrival-machine and the service-machine. The first subsystem contains the machine pair (0, 1) with Machine 0 being the arrival-machine and Machine 1 being the service-machine. The second subsystem contains the machine pair (1,2) with Machine 1 being the arrival-machine and Machine 2 being the service-machine. Finally, the third subsystem contains the machine pair (2,3) with Machine 2 being the arrival-machine and Machine 3 being the service-machine.

For notational purposes, $\bar{t}_0, \bar{t}_1, \bar{t}_2$, and $\bar{t}_3$ will denote the average time for the four machines. For our computations, we will use hours for time units; therefore, $\bar{t}_0 = 1$ hr, $\bar{t}_1 = 4/3$ hr, $\bar{t}_2$ =5/4 hr, and $\bar{t}_3 = 6/5$ hr. In addition, when formula are given in general, the total number of workstations will be denoted by n.

9.3.1 The First Forward Pass

The decomposition procedure makes a first pass through the subsystems starting with Subsystem 1 called the *forward pass*. The purpose of the forward pass is to update the arrival-machine processing time distribution. The basis for the update is an analysis of the departure characteristics of the previous subsystem. The difficulty with the arrival-machine is that it might become starved and so the inter-arrival time is longer than normal; thus, to update the inter-arrival times for Subsystem k, we need to determine the probability that a departure from Subsystem $(k-1)$ leaves that subsystem empty. Except for the first subsystem, the processing time for the arrival-machine is approximated by a GE_2 distribution and it will become important to know the phase of the arrival-machine when the departure occurs. For the first subsystem, the arrival-machine is always modeled with the exponential distribution and we denote the probability that a departure from Subsystem 1 leaves it empty by $p^0_{d,1}$. For Subsystem k $(k > 1)$, we denote the probability that a departure from that subsystem leaves it empty and that the arrival-machine is in Phase i at the time of departure by $p^{(i,0)}_{d,k}$.

Since Subsystem 1 has an arrival-machine that can never be starved, the analysis of the first subsystem does not update the arrival-machine, but it does determine the probability that a departure leaves the system empty.

9.3.1.1 First Forward Pass for Subsystem 1

Subsystem 1 has exponential inter-arrival times with mean of 1 hr and an exponentially distributed service time with mean time 4/3 hr and a system capacity of 4 units. The state space can be represented as $\{0,1,2,3,4,b\}$ due to the fact that the processing times of the machines are exponential and thus the state space does not need an indicator for the second phase of service. (An exponential process has only one phase). Using Property 9.1, the probabilities are easily computed and are shown in Table 9.1.

Table 9.1 Probabilities for Subsystem 1 — first forward pass

Jobs	0	1	2	3	4	b
Prob.	0.0722	0.0962	0.1283	0.1711	0.2281	0.3041

The probability that a departure leaves the subsystem empty is the probability that this job was the only job in the subsystem at the departure time. Note that no departure can occur if the subsystem is empty. Thus, this probability is computed as the conditional probability (Definition 1.3) that there is one job in the system given that system in not empty; that is,

$$p^0_{d,1} = \frac{0.0962}{1-0.0722} = 0.1037 \; .$$

The throughput rate for this subsystem is based on the steady-state probabilities and is computed from the mean arrival rate times the probability that the system is not blocked. Thus, the throughput rate for subsystem one is

$$th(1) = 1 \times (1 - 0.3041) = 0.6959/\text{hr} \; .$$

Summary: The first forward pass of Subsystem 1 will always use probabilities derived from Property 9.1. Assume these probabilities are denoted by v_i for $i = 0, \cdots, w_{\max}+1$, where $w_{\max}+1$ represents the blocked state. (For ease of notation, we shall let $v_{w_{\max}}$ be written as $v_{\max}$ and let $v_{w_{\max}+1}$ be written as v_b.) Then the probability that a departure will leave the system empty is

$$p^0_{d,1} = \frac{v_1}{1-v_0} \; , \tag{9.4}$$

and the mean throughput rate is given by

$$th(1) = \frac{1-v_b}{\bar{t}_0} \; , \tag{9.5}$$

where $\bar{t}_0$ is the mean time needed to release jobs into the factory (or the time to prepare jobs for processing) once there is room available.

9.3.1.2 First Forward Pass for Subsystem 2

The second subsystem has Machine 1 as the arrival-machine and Machine 2 as the service-machine. The buffer for this subsystem has a limit of 2 therefore its capacity is 3 units. For the first pass, the service distribution is exponential with mean time 5/4 hr. The inter-arrival time distribution used for the arrival-machine is the machine's nominal service time (exponentially distributed with mean time 4/3 hr or, equivalently, a mean rate of 0.75/hr) interspersed with periodic delays due to machine starvation. That is, periodically this arrival-machine (that is really the machine for Subsystem 1) does not have a job to process and must wait for its own arrival which is exponentially distributed with a mean time of 1 hr. Thus, when an arrival occurs to Subsystem 2, this job departs from Subsystem 1 and if it leaves that subsystem empty, the time until the next departure includes both the idle time of Machine 1 plus the service time of Machine 1. Therefore, the probability that

Subsystem 1 is empty at a departure time (namely, $p^0_{d,1} = 0.1037$) is used to activate this delay, and the phase-type inter-arrival time distribution for Subsystem 2 is an MGE_2 distribution with a generator matrix G

$$G = \begin{bmatrix} -1.0 & 1.0 \\ 0.0 & -0.75 \end{bmatrix}$$

and $\boldsymbol{\alpha} = (0.1037, 0.8963)$. This process has a mean time of $E[T_a(2)] = 1.4370$ hr and an SCV of $C^2_a(2) = 0.9561$.

The decomposition procedure is to always replace the arrival-machine processing time distribution with a GE_2 approximation. Thus, fitting the moments for the MGE_2 service process with the GE_2 distribution, we obtain parameters associated with the arrival-machine for the second subsystem of

$$(\alpha_2, p_2, \beta_2) = (0.7278, 0.0878, 1.3917)$$

using Eq. (9.2). Since the processing time distribution for the service-machine is exponential, the state space is not quite a large as the state space of Sect. 9.2.2. (Notice that the reason the service-machine has an exponential distribution is that we have not yet estimated the probability of blocking at the downstream subsystem since this is the first pass.) The state space for Subsystem 2 (first pass) is

$$\{(10), (20), (11), (21), (12), (22), (13), (23), (b3)\} .$$

We follow the same logic as the development of the generator matrix in Sect. 9.2.3 and develop the generator matrix for Subsystem 2 (first pass). The general form of the generator is given as follows where blanks in the matrix represent zeros:

$$Q = \begin{bmatrix}
-\alpha & p\alpha & (1-p)\alpha & & & & & & \\
 & -\beta & \beta & & & & & & \\
\mu & & -(\alpha+\mu) & p\alpha & (1-p)\alpha & & & & \\
 & \mu & & -(\beta+\mu) & \beta & & & & \\
 & & \mu & & -(\alpha+\mu) & p\alpha & (1-p)\alpha & & \\
 & & & \mu & & -(\beta+\mu) & \beta & & \\
 & & & & \mu & & -(\alpha+\mu) & p\alpha & (1-p)\alpha \\
 & & & & & \mu & & -(\beta+\mu) & \beta \\
 & & & & & & \mu & & -\mu
\end{bmatrix} ,$$

where (α, p, β) are the parameters (α_2, p_2, β_2) and $\mu = 1/\bar{t}_2$.

Property 9.3 can now be used to determine the steady-state probabilities for Subsystem 2. These are shown in Table 9.2.

We will need not only the probability that a departure from the subsystem leaves it empty, but also the joint probability for the state of the arrival process when the departure occurs. The probability that a departure from Subsystem 2 leaves the subsystem empty while its arrival-machine is in the first phase is

Table 9.2 Probabilities for Subsystem 2 — first forward pass

Phase of	Number of Jobs in System			
Arrival-Machine	0	1	2	3
1	0.2410	0.2192	0.1904	0.1659
2	0.0163	0.0091	0.0073	0.0048
b				0.1461

$$p_{d,2}^{(1,0)} = \frac{0.2192}{1-(0.2410+0.0163)} = 0.2951\ ,$$

and the probability that a departure from Subsystem 2 leaves the system empty while its arrival-machine is in the second phase is

$$p_{d,2}^{(2,0)} = \frac{0.0091}{1-(0.2410+0.0163)} = 0.0122\ .$$

Note that again the probability is conditioned on the subsystem not being empty since no departure can occur while it is empty.

The throughput of Subsystem 2 is the arrival rate (reciprocal of 1.4370 hr) times the probability that the subsystem is not blocked, which is

$$th(2) = 0.6959 \times (1-0.1461) = 0.5942/\text{hr}\ .$$

Summary: The first forward pass of Subsystem 2 involves using an MGE_2 distribution for the arrival-machine with generator

$$G = \begin{bmatrix} -1/\bar{t}_0 & 1/\bar{t}_0 \\ 0.0 & -1/\bar{t}_1 \end{bmatrix}$$

and $\boldsymbol{\alpha} = (p_{d,1}^0,\, 1-p_{d,1}^0)$. Using some matrix algebra, it is possible to find the mean and variance of this in closed form; therefore, the iteration does not need to express the matrix explicitly. Thus, the mean and variance of the the inter-arrival times to the second subsystem (first pass) are given by

$$\begin{aligned} E[T_a(2)] &= \bar{t}_1 + p_{d,1}^0\,\bar{t}_0 \quad \text{and} \\ \text{Var}[T_a(2)] &= \bar{t}_1^2 + p_{d,1}^0\,\bar{t}_0^2\left(2-p_{d,1}^0\right)\ . \end{aligned} \tag{9.6}$$

With the mean and SCV for the arrival process determined, Eq. (9.1) or (9.2) is used to obtain the parameters for the approximating GE_2 which are denoted by (α_2, p_2, β_2) and these parameters in turn are used to obtain the generator matrix for the steady state probabilities of Subsystem 2. The form of the generator matrix is given on page 296. It should not be difficult to form the generator matrix for any value of $w_{\max}$ once it is observed that the generator is made up of 2×2 submatrices. The first two rows and columns have a slightly different form, but a pattern can be observed. In addition, the final (single) row and column are also different. The steady-state probabilities are obtained from Property 9.3. Let these probabilities be denoted by v_i where i is an ordered pair representing a state in the state space of the form given on page 296.

The probability that a departure from Subsystem 2 (after the first pass) leaves the system empty while its arrival-machine is in the first phase is

$$p_{d,2}^{(1,0)} = \frac{v_{(1,1)}}{1 - \left(v_{(1,0)} + v_{(2,0)}\right)} , \tag{9.7}$$

and the probability that a departure from Subsystem 2 leaves the system empty while its arrival-machine is in the second phase is

$$p_{d,2}^{(2,0)} = \frac{v_{(2,1)}}{1 - \left(v_{(1,0)} + v_{(2,0)}\right)} . \tag{9.8}$$

Finally, the mean throughput rate

$$th(2) = \frac{1 - v_b}{E[T_a(2)]} , \tag{9.9}$$

where $E[T_a(2)]$ is from Eq. (9.6).

9.3.1.3 First Forward Pass for Subsystem Three

The third subsystem has Machine 2 as the arrival-machine and Machine 3 as the service-machine. The service-machine has an exponential processing time with mean time 6/5 hr or, equivalently, with mean rate 0.8333/hr. If an arrival occurs to Subsystem 3 leaving the previous subsystem not empty, the next inter-arrival time will have a mean of 1.25 hr (service time for Machine 2 with rate $1/\bar{t}_2 = 0.8$); otherwise, there will be an additional delay in the inter-arrival time based on the phase of the arrival-machine. Recall that the parameters for the GE_2 distribution used for the inter-arrivals to Subsystem 2 were $(\alpha_2, p_2, \beta_2) = (0.7278, 0.0878, 1.3917)$; therefore the inter-arrival distribution for Subsystem 3 has an MGE_3 distribution with generator

$$G = \begin{bmatrix} -\alpha_2 & p_2\,\alpha_2 & (1-p_2)\,\alpha_2 \\ 0 & -\beta_2 & \beta_2 \\ 0 & 0 & -1/\bar{t}_2 \end{bmatrix} = \begin{bmatrix} -0.7278 & 0.0639 & 0.6639 \\ 0 & -1.3917 & 1.3917 \\ 0 & 0 & -0.8 \end{bmatrix}$$

and with initial probabilities $\boldsymbol{\alpha} = (0.2951, 0.0122, 0.6927)$. Notice that the first two initial probabilities are given $p_{d,2}^{(1,0)}$ and $p_{d,2}^{(2,0)}$.

From Property 9.2, we have that the MGE_3 process has a mean of mean time of $E[T_a(3)] = 1.6829$ hr and a SCV of $C_a^2(3) = 0.9110$. As always, we simplify the arrival process by approximating it with a GE_2 process and from Eq. (9.2), we have the parameters as

$$(\alpha_3, p_3, \beta_3) = (0.6523, 0.1781, 1.1884) .$$

Because Subsystem 3 is the final subsystem within the factory, there is no need to calculate the probability that a departure will leave the system empty since the service-machine to the final subsystem does not act as an arrival-machine. Therefore, the forward pass for the first iteration is finished since all arrival-machines have been updated.

Summary: For a systems with more that two workstations, the determination update of the arrival-machine follow the same procedure, namely first an MGE_3 distribution is determined and than a GE_2 approximating distribution is calculated. In order to give a general form, assume for this summary that we are analyzing Subsystem k. The MGE_3 process used for the inter-arrival times is described using a generator matrix given by

$$G = \begin{bmatrix} -\alpha_{k-1} & p_{k-1}\,\alpha_{k-1} & (1-p_{k-1})\,\alpha_{k-1} \\ 0 & -\beta_{k-1} & \beta_{k-1} \\ 0 & 0 & -1/\bar{t}_{k-1} \end{bmatrix}$$

where the vector of initial probabilities is given by

$$\boldsymbol{\alpha} = \left(p_{d,k-1}^{(1,0)},\, p_{d,k-1}^{(2,0)},\, 1 - p_{d,k-1}^{(1,0)} - p_{d,k-1}^{(2,0)} \right) .$$

Notice that the parameters of the generator depend on the GE_2 parameters determined for the previous subsystem as well as the mean service rate of the service-machine of the previous subsystem. The initial probabilities depend on the probabilities that a departing job from the previous subsystem leaves the arrival-machine in either Phase 1 or 2. Again, using some matrix algebra, it is possible to find the mean and variance of this in closed form; therefore, the iteration does not need to express the matrix explicitly. Thus, the mean and variance of the the inter-arrival times to the second subsystem (first pass) are given by

$$E[T_a(k)] = \bar{t}_{k-1} + \frac{p_{d,k-1}^{(1,0)}}{\alpha_{k-1}} + \frac{\pi}{\beta_{k-1}} \tag{9.10}$$

$$\begin{aligned} \mathrm{Var}[T_a(k)] &= \bar{t}_{k-1}^2 + \frac{p_{d,k-1}^{(1,0)}\,(2 - p_{d,k-1}^{(1,0)})}{\alpha_{k-1}^2} \\ &\quad + \frac{\pi\,(2-\pi)}{\beta_{k-1}^2} + \frac{2p_{d,k-1}^{(1,0)}\,(p_{k-1} - \pi)}{\alpha_{k-1}\,\beta_{k-1}} , \end{aligned}$$

where $\pi = p_{d,k-1}^{(1,0)} \times p_{k-1} + p_{d,k-1}^{(2,0)}$. With the mean and SCV for the arrival process determined, Eq. (9.1) or (9.2) is used to obtain the parameters for the approximating GE_2 which are denoted by (α_k, p_k, β_k).

9.3.2 The Backward Pass

Each iteration of the algorithm involves a forward pass and then a backward pass. The forward pass updates the arrival-machine distribution parameters and the backward pass updates the service-machine distribution parameters. The difficulty with analyzing the service-machine is that after it is finished processing, the next subsystem may have no space for it so that the service-machine becomes blocked. This effectively increases the job delay time of the job controlling the machine. The way this is handled in the decomposition procedure, where the connection between adjacent workstations is not available, is to increase the job processing times. Thus, in the backwards pass, the probability that an arriving job finds a full subsystem is needed because the time that it takes to unblock service-machine is dependent on the phase of the downstream machine. In addition, the probability that the arriving job finds the service-machine in a specific phase is also needed. Therefore, in the following discussion, we will let $p_{a,k}^{(i,F)}$ denote the probability that an arrival to Subsystem k ($k < n$) finds the subsystem full and its service-machine in Phase i. As we begin the backwards pass, we start with the final subsystem (i.e., Subsystem n) and its service-machine is always exponential so it has no phases; hence, $p_{a,n}^{F}$ will be used to denote that an arrival to the final subsystem finds the subsystem full.

9.3.2.1 Backward Pass for Subsystem 3

The service-machine for the final subsystem needs no updating since it is never blocked; however, the probability that an arrival to the final subsystem finds the subsystem full must be calculated so that the service-machine for the penultimate subsystem can be updated. To obtain this probability, the steady-state probabilities for the subsystem must be determined. The data that are needed for determining the generator matrix for the steady-state probabilities are the arrival-machine parameters (namely, α_3, p_3, and β_3 from p. 298) and the mean processing rate for the service-machine (namely, $1/\bar{t}_3$). The state space for Subsystem 3 is very similar to Subsystem 2 (see p. 296) except there are two additional states since the kanban limit for Subsystem 3 is 4 jobs whereas the capacity for Subsystem 2 was 3 jobs. The generator matrix is also very similar (see p. 296) except it will have two more rows and columns. Once the generator matrix is constructed, the probabilities can be obtained from Property 9.3 to yield the results in Table 9.3.

Table 9.3 Probabilities for Subsystem 3 — first backward pass

Phase of	Number of Jobs in System				
Arrival-Machine	0	1	2	3	4
1	0.2869	0.2246	0.1578	0.1110	0.0787
2	0.0407	0.0180	0.0125	0.0082	0.0045
b					0.0571

The probability that an arrival finds the system full (i.e., the probability that an arrival gets blocked) is a conditional probability based on the state of the arrival-machine when the arrival occurs. (Notice that because the service-machine process is not exponential, the time-averaged probability of finding the system full is not the same as the probability of finding the system full at a departure time.) The probability that an arrival will occur to Subsystem 3 while its arrival-machine is in Phase 1 is $1 - p_3 = 0.8219$ (recall that p_3 is a parameter of the arrival-machine GE_2 distribution as determined by the forward pass) and it will occur while the arrival-machine is in Phase 2 with probability $p_3 = 0.1781$. The conditional probability that the arriving job will find a full system given that the arrival-machine is in Phase 1 at the arrival time is

$$\begin{aligned}\Pr\{\text{full}|\text{Arrival Phase 1}\} &= \frac{0.0787}{0.2869 + 0.2246 + 0.1578 + 0.1110 + 0.0787}\\ &= 0.0916\,,\end{aligned}$$

and the conditional probability that the arriving job will find a full system given that the arrival-machine is in Phase 2 is

$$\begin{aligned}\Pr\{\text{full}|\text{Arrival Phase 2}) &= \frac{0.0045}{0.0407 + 0.0180 + 0.0125 + 0.0082 + 0.0045}\\ &= 0.0534\,;\end{aligned}$$

therefore, the probability of blocking occuring upon an arrival from the second subsystem is

$$p^F_{a,3} = 0.8219 \times 0.0916 + 0.1781 \times 0.0534 = 0.0849\,.$$

The throughput of Subsystem 3 is the arrival rate (the reciprocal of 1.6829 hr) times the probability that Subsystem 3 is not blocked, which is

$$th(3) = 0.5942 \times (1 - 0.0571) = 0.5603/\text{hr}\,.$$

Summary: The backward pass starts with the final subsystem and begins with determining its steady-state probabilities. The state space will always be of the form given on p. 296 and the generator will be similar to that on p. 296. Once the generator matrix is constructed, Property 9.3 is used to yield the probabilities denoted as v_i for i an ordered pair representing a state. The blocking probability is given by

$$p^F_{a,n} = (1 - p_n)\,\frac{v_{(1,\max)}}{\sum_{i=0}^{\max} v_{(1,i)}} + p_n\,\frac{v_{(2,\max)}}{\sum_{i=0}^{\max} v_{(2,i)}}\,, \tag{9.11}$$

where p_n is the parameter from the approximating GE_2 distribution for the arrival-machine of the final subsystem. Finally, the mean throughput rate is

$$th(n) = \frac{1 - v_b}{E[T_a(n)]}\,, \tag{9.12}$$

where $E[T_a(n)]$ is from Eq. (9.10).

9.3.2.2 Backward Pass Subsystem 2

The probability that service-machine for Subsystem 2 is blocked by Subsystem 3 being full at a departure time is $p^F_{a,3} = 0.0851$. Thus, the service distribution is made up of an exponential service time with mean time 5/4 and an 8.51% chance of an addition exponential wait with mean time 6/5. This phase-type service time distribution is represented as an MGE_2 process with generator matrix G

$$G = \begin{bmatrix} -0.8333 & 0.8333 \\ 0 & -0.8000 \end{bmatrix}$$

with the starting state distribution $\boldsymbol{\alpha} = (0.0849, 0.9151)$. (Notice that the entries of the generator are rates and thus are the reciprocals of the mean times.) This process has a mean time of $E[T_s(2)] = 1.3521$ hr and an SCV of $C_s^2(2) = 0.9830$ (Property 9.2), and the following parameter set

$$(\mu_2, q_2, \gamma_2) = (0.7525, 0.0338, 1.4795)$$

obtained from Property 9.3 will be used for the GE_2 distribution that approximates the service-machine of Subsystem 2. Recall that the forward pass from p. 296 established that the arrival-machine for the subsystem could be approximated by a GE_2 distribution with parameters $(0.7278, 0.0878, 1.3917)$. Since both the arrival-machine and the service-machine are modeled with the GE_2 process, the state space will be composed of three-tuples and the generator matrix for the two-node subsystem will be similar to Fig. 9.7, except that there will be 16 rows and columns. To structure the generator matrix, the pattern for the matrix should be obvious from Fig. 9.7 if you look for the 4×4 submatrices. The first and the last two rows and columns have a slightly different form, but the other rows and columns will have a repeating submatrices along the tri-diagonal block submatrices. Once the generator matrix is formed, the resulting steady-state probabilities for the subsystem are found from Property 9.3 as shown in Table 9.4.

Table 9.4 Probabilities for Subsystem 2 — first backward pass

Phase of	Phase of	Number of Jobs in System				
Arrival-Machine	Service-Machine	0	1	2	3	3-blocked
1	0 or 1	0.2079	0.2034	0.1909	0.1800	—
1	2	—	0.0023	0.0030	0.0031	—
2 or b	0 or 1	0.0142	0.0087	0.0076	0.0054	0.1687
2 or b	2	—	0.0001	0.0001	0.0001	0.0044

The probability that an arrival to Subsystem 2 will find it full is based on the conditional probability that the arrival occurs while the arrival-machine is in either Phase 1 or 2 but not blocked. However, we also need the joint probability that the arrival will find the subsystem full and will find the service-machine in a specific phase. Before writing these probabilities, note that the probability that the arrival-machine to Subsystem 2 is in Phase 1 (and not blocked) equals 0.7906 and the probability that the arrival-machine is in Phase 2 (and not blocked) is 0.0362. In addition, the probability that an arrival enters the subsystem from Phase 1 of the arrival-machine is 91.22% (namely, $1-p_2$), and enters from Phase 2 is 8.78% (namely, p_2); thus, the probability that an arrival to Subsystem 2 will find the subsystem full with the service-machine in Phase 1 is given by

$$p_{a,2}^{(1,F)} = 0.9122 \times \frac{0.1800}{0.7906} + 0.0878 \times \frac{0.0054}{0.0362} = 0.2208 \,,$$

and the probability that an arrival to Subsystem 2 will find the subsystem full with the service-machine in Phase 2 is given by

$$p_{a,2}^{(2,F)} = 0.9122 \times \frac{0.0031}{0.7906} + 0.0878 \times \frac{0.0001}{0.0362} = 0.0038 \,.$$

Finally, the throughput rate is the arrival rate times the probability that the system is not blocked, or

$$th(2) = 0.6959 \times (1 - 0.1687 - 0.0044) = 0.5754/\text{hr} \,.$$

Summary: The backwards pass for penultimate subsystem (namely, Subsystem $(n-1)$) always uses the MGE_2 distribution for the service-machine process; thus, the mean and variance of service times can be given as

$$\begin{aligned} E[T_s(n-1)] &= \bar{t}_{n-1} + p_{a,n}^F \bar{t}_n \quad \text{and} \\ \text{Var}[T_s(n-1)] &= \bar{t}_{n-1}^2 + p_{a,n}^F \bar{t}_{\max}^2 \left(2 - p_{a,n}^F\right) \,, \end{aligned} \tag{9.13}$$

where $\bar{t}_n$ denotes the mean service time of the final workstation and $\bar{t}_{n-1}$ denotes the mean service time of the penultimate workstation.

With the mean and SCV for the service-machine process determined, Eq. (9.1) or (9.2) is used to obtain the parameters for the approximating GE_2 which are denoted by $(\mu_{n-1}, q_{n-1}, \gamma_{n-1})$. This distribution is combined with the GE_2 distribution determined in the forward pass that is used for the arrival process. The parameters for the arrival process were denoted by $(\alpha_{n-1}, p_{n-1}, \beta_{n-1})$. Thus, we have the data needed to establish the steady-state probabilities of Subsystem $(n-1)$. The form of the generator matrix is similar to Fig. 9.7 with the steady-state probabilities coming from Property 9.3. These probabilities are denoted by v_i where i is a three-tuple. The probability that an entry will find the system blocked while the service-machine is in the first phase is

$$\begin{aligned} p_{a,n-1}^{(1,F)} = (1-p_{n-1}) & \frac{v_{(1,1,\max)}}{v_{(1,0,0)} + \sum_{i=1}^{\max}[v_{(1,1,i)} + v_{(1,2,i)}]} \\ & + p_{n-1} \frac{v_{(2,1,\max)}}{v_{(2,0,0)} + \sum_{i=1}^{\max}[v_{(2,1,i)} + v_{(2,2,i)}]} \,, \end{aligned} \tag{9.14}$$

where p_{n-1} is the parameter from the GE_2 distribution for the arrival-machine of the penultimate subsystem. Similarly, the probability that an entry will find the system blocked while the service-machine is in the second phase is

$$p_{a,n-1}^{(2,F)} = (1-p_{n-1})\,\frac{v_{(1,2,\max)}}{v_{(1,0,0)}+\sum_{i=1}^{\max}[v_{(1,1,i)}+v_{(1,2,i)}]} + p_{n-1}\,\frac{v_{(2,2,\max)}}{v_{(2,0,0)}+\sum_{i=1}^{\max}[v_{(2,1,i)}+v_{(2,2,i)}]}\,. \tag{9.15}$$

(Notice that the phase index in the superscript of the blocking probabilities (Eqs. 9.14 and 9.15) refers to the phase of the service-machine; whereas, the index in the superscript of the starving probabilities (Eqs. 9.7 and 9.8) refers to the phase of the arrival-machine.)

From this point on, any subsystem except for the first and last subsystems will have the full state space involving three tuples. Thus, the mean throughput rate, for $k = 2,\cdots,n-1$ is given by

$$th(k) = \frac{1-v_{(b,1,\max)}-v_{(b,2,\max)}}{E[T_a(k)]}\,. \tag{9.16}$$

9.3.2.3 Backward Pass for Subsystem 1

Note that there are three possibilities when a job has finished on the service-machine: (1) the service-machine is not blocked (this has a probability of 77.54%), in which case the service time will be exponential, (2) the service-machine is blocked and the service-machine of Subsystem 2 is in Phase 1 (this has a probability 22.08%), in which case the service time for the next job will experience a delay of according to a GE_2 distribution, and (3) the service-machine is blocked and the service-machine of Subsystem 2 is in Phase 2 (this has a probability 0.38%), in which case the service time for the next job will experience a delay of an additional exponential time associated with the second phase of the GE_2 distribution. Thus, the processing time distribution for the service-machine for Subsystem 1 is an MGE_3 process with generator matrix G

$$G = \begin{bmatrix} -\mu_2 & q_2\,\mu_2 & (1-q_2)\,\mu_2 \\ 0 & -\gamma_2 & \gamma_2 \\ 0 & 0 & -1/\bar{t}_1 \end{bmatrix} = \begin{bmatrix} -0.7524 & 0.0256 & 0.7268 \\ 0 & -1.4792 & 1.4792 \\ 0 & 0 & -0.75 \end{bmatrix}$$

and with initial probabilities $\boldsymbol{\alpha} = (0.2207, 0.0038, 0.7755)$. Notice that the first two initial probabilities are $p_{a,2}^{(1,F)}$ and $p_{a,2}^{(2,F)}$. This MGE_3 process has a mean time of 1.6344 hr and an SCV = 0.9325 (Property 9.2). Thus, the following parameter set

$$(\mu_1, q_1, \gamma_1) = (0.6562, 0.1350, 1.2238)$$

will be used for the GE_2 distribution that approximates the service-machine of Subsystem 1. Recall that for the first subsystem, the arrival-machine is never starved so that its processing time distribution is exponential with mean 1 hr. We do not need to determine the blocking probabilities, but an estimate for the throughput is needed; therefore, we will calculate the steady-state probabilities.

The state space for this system will be slightly different from those we have had so far an is given as

$$\{(00),(11),(21),(12),(22),(13),(23),(14),(24),(1b),(2b)\}\,.$$

The difference between this state space and the state space on Page 296 (other than the larger buffer capacity) is that the first element of the ordered pair refers to the phase of the service-machine instead of the arrival-machine; thus, there is only one empty state and two blocked states instead of two empty states and one blocked state.

To construct the generator matrix for the subsystem, it is best to look for a pattern among 2 × 2 submatrices, except that the first row and column will be different. The first few elements of the matrix are as follows and we leave it to the reader to finish its construction:

$$Q = \begin{bmatrix} -1/\bar{t}_0 & 1/\bar{t}_0 & 0 & 0 & 0 & \cdots \\ (1-q_1)\mu & -(\mu_1+1/\bar{t}_0) & q_1\mu & 1/\bar{t}_0 & 0 & \\ \gamma_1 & 0 & -(\gamma_1+1/\bar{t}_0) & 0 & 1/\bar{t}_0 & \\ 0 & (1-q_1)\mu & 0 & -(\mu_1+1/\bar{t}_0) & q_1\mu & \\ 0 & \gamma_1 & 0 & 0 & -(\gamma_1+1/\bar{t}_0) & \cdots \\ \vdots & & & & \vdots & \ddots \end{bmatrix}.$$

From Property 9.3, the steady-state probabilities for the subsystem can be found as given in Table 9.5.

Table 9.5 Probabilities for Subsystem 1 — first backwards pass

Phase of		Number of Jobs in System				
Service-Machine	0	1	2	3	4	b
0	0.0329					
1		0.0534	0.0882	0.1464	0.2432	0.3706
2		0.0021	0.0045	0.0078	0.0132	0.0376

These probabilities yield a mean throughput of

$$th(1) = 1 \times (1 - 0.3706 - 0.0376) = 0.5918/\text{hr}\,.$$

The throughput estimates for the three subsystems after this first iteration are have been calculated to be 0.5918/hr, 0.5754/hr, and 0.5603/hr. (These are all from the backward pass since they are the more recent estimates than the forward pass.) The algorithm is finished when all three throughputs are the same and do not change with the iteration.

Summary: The update to the service-machine for Subsystem k involves forming the MGE_3 distribution from the parameters determined during the backward pass of Subsystem $k+1$. The generator will have the form

$$G = \begin{bmatrix} -\mu_{k+1} & q_{k+1}\,\mu_{k+1} & (1-q_{k+1})\,\mu_{k+1} \\ 0 & -\gamma_{k+1} & \gamma_{k+1} \\ 0 & 0 & -1/\bar{t}_k \end{bmatrix} ,$$

and with the initial probability vector given by

$$\boldsymbol{\alpha} = \left(p_{a,k+1}^{(1,F)}, p_{a,k+1}^{(2,F)}, 1 - p_{a,k+1}^{(1,F)} - p_{a,k+1}^{(2,F)} \right) .$$

Again, it is possible to obtain closed form expressions of the application of Property 9.2 to this generator. Thus, the mean and variance for the service-machine processing time for Subsystem k are

$$E[T_s(k)] = \bar{t}_k + \frac{p_{a,k+1}^{(1,F)}}{\mu_{k+1}} + \frac{\pi}{\gamma_{k+1}} \tag{9.17}$$

$$\mathrm{Var}[T_a(k)] = \bar{t}_k^2 + \frac{p_{a,k+1}^{(1,F)}\,(2 - p_{a,k+1}^{(1,F)}\,)}{\mu_{k+1}^2} + \frac{\pi\,(2-\pi)}{\gamma_{k+1}^2} + \frac{2p_{a,k+1}^{(1,F)}\,(q_{k+1}-\pi)}{\mu_{k+1}\,\gamma_{k+1}} ,$$

where $\pi = p_{a,k+1}^{(1,F)} \times q_{k+1} + p_{a,k+1}^{(2,F)}$. With the mean and SCV for the arrival process determined, Eq. (9.1) or (9.2) is used to obtain the parameters for the approximating GE_2 which are denoted by (μ_k, q_k, γ_k). Using this distribution together with the exponential-arrival process with mean time $\bar{t}_0$, the steady-state probabilities can be obtained using a generator similar to the generator on Page 305.

The mean throughput rate is given by

$$th(1) = \frac{1 - v_{(1,b)} - v_{(2,b)}}{\bar{t}_0} . \tag{9.18}$$

This completes one iteration of the algorithm. The iterations should be continued until the throughput estimates do not change from one iteration to the next.

9.3.3 The Remaining Iterations

Some of the procedures used in the first forward passes were different simply because the subsystems had not been analyzed before. In the following subsections we indicate the adjustments that will have to be made for the remaining iterations.

9.3.3.1 The Remaining Forward Passes for Subsystem 1

The steady-state probabilities for Subsystem 1 were obtained on the previous backward pass and the arrival-machine needs no updating since it can never be starved. Because the service-machine is no longer exponential, the probability of a departing job leaving the system empty must be conditioned on the phase of the service-machine from which the job departs; otherwise the probability would be the time-averaged probability instead of a departure point probability. This is similar to the logic used for Eq. (9.11) except the conditioning is on the service-machine instead of the arrival-machine. Thus, the probability that a departure will leave the subsystem empty is given as

$$p_{d,1}^{0} = (1-q_1)\,\frac{v_{(1,1)}}{v_{(b,1)}+\sum_{i=1}^{\max} v_{(1,i)}} + q_1\,\frac{v_{(2,1)}}{v_{(b,2)}+\sum_{i=1}^{\max} v_{(2,i)}}\,, \tag{9.19}$$

where q_1 is the parameter from the GE_2 distribution for the service-machine of the Subsystem 1 and $v_{(b,i)}$ is the steady-state probability that the arrival is blocked and the service-machine is in Phase i. These quantities were determined during the backward pass for the first subsystem.

9.3.3.2 The Remaining Forward Passes for Subsystem 2

The determination of the mean and variance for the inter-arrival times is the same as the first forward pass; namely, use Eq. (9.6) and then determine new values for (α_2, p_2, β_2) based on the mean and SCV of the inter-arrival times. The generator matrix is formed according to Fig. 9.8 (i.e., the state space is made up of three-tuples) using the service-machine parameters saved from the previous backward pass, and then Property 9.3 is used to obtain the steady-state probabilities.

Because the service-machine is no longer exponential, the probability of a departing job leaving the system empty must be conditioned on the phase of the service-machine from which the job departs as was done for the first subsystem. Here a joint probability is needed for the phase of the arrival-machine. Thus, the probability that a departure will leave Subsystem k empty while the arrival-machine in Phase 1 is

$$p_{d,k}^{(1,0)} = (1-q_k)\,\frac{v_{(1,1,1)}}{v_{(b,1,\max)} + \sum_{i=1}^{\max}[v_{(1,1,i)} + v_{(2,1,i)}]} + q_k\,\frac{v_{(1,2,1)}}{v_{(b,2,\max)} + \sum_{i=1}^{\max}[v_{(1,2,i)} + v_{(2,2,i)}]}\,, \tag{9.20}$$

where q_k is the parameter from the GE_2 distribution for the service-machine of Subsystem k with $k = 2,\cdots,n-1$. The probability that a departure will leave Subsystem k empty with the arrival-machine in Phase 2 is

$$p_{d,k}^{(2,0)} = (1-q_k)\,\frac{v_{(2,1,1)}}{v_{(b,1,\max)} + \sum_{i=1}^{\max}[v_{(1,1,i)} + v_{(2,1,i)}]} + q_k\,\frac{v_{(2,2,1)}}{v_{(b,2,\max)} + \sum_{i=1}^{\max}[v_{(1,2,i)} + v_{(2,2,i)}]}\,. \tag{9.21}$$

9.3.3.3 The Remaining Forward and Backward Passes

The forward pass for the final subsystem and all the backward passes remain the same as during the first iteration. Recall that one iteration includes both the forward and backward passes. Once the throughputs converge, the decomposition algorithm is finished.

9.3.4 Convergence and Factory Performance Measures

The changing values of throughput by iteration are shown in Table 9.6 and it is seen that five iterations are sufficient for convergence. The throughput values in the table are from the backwards pass. The values calculated from the forward pass are ignored.

Table 9.6 Throughput results from the first five iterations

	Subsystem 1	Subsystem 2	Subsystem 3
Iteration 1	0.5916/hr	0.5754/hr	0.5602/hr
Iteration 2	0.5831/hr	0.5820/hr	0.5798/hr
Iteration 3	0.5824/hr	0.5823/hr	0.5821/hr
Iteration 4	0.5823/hr	0.5823/hr	0.5823/hr
Iteration 5	0.5823/hr	0.5823/hr	0.5823/hr

The performance measures for each workstation and the system as a whole are computed from the throughput rate and the steady-state probabilities for each subsystem as representative of the associated workstation. Notice that the throughputs are the same for each subsystem (0.5823/hr for our example) if convergence has taken place, but of course, the steady-state probabilities are different. To illustrate, consider Table 9.7 that shows the steady-state probabilities for Subsystem 1 as determined by the final backwards pass. The last row of the table gives the probabilities for the number of jobs in the subsystem. Notice that the probability of 4 jobs in the subsystem is the sum for the last two columns since the system contains 4 jobs when it is blocked. Thus the average number of jobs in the system is

Table 9.7 Probabilities for Subsystem 1 — fifth backwards pass

Phase of	Number of Jobs in System					
Service-Machine	0	1	2	3	4	b
0	0.0303					
1		0.0501	0.0846	0.1434	0.2434	0.3745
2		0.0023	0.0048	0.0086	0.0148	0.0432
Sum	0.0303	0.0524	0.0894	0.1520	0.6759	—

$$WIP(1) = 1 \times 0.0524 + 2 \times 0.0894 + 3 \times 0.1520 + 4 \times 0.6759 = 3.391\ ,$$

and the cycle time (from Little's Law) is

$$CT(1) = \frac{WIP(1)}{th(1)} = \frac{3.391}{0.5823} = 5.823 \text{ hr}\ .$$

The system WIP_s is the sum of individual the WIP's for each workstation (subsystem) and equals 7.215 jobs, and the cycle-time estimate is $7.215/0.5823 = 12.391$ hr. The system and individual workstation results from the analytical procedure are compared with those from a simulation model. The simulation run was long enough so that the half-width of the confidence limits for each estimate was approximately 1% of the estimate or smaller. The simulation and analytical comparisons are given in Table 9.8. Both the mean throughput and cycle time errors are less that 1% and the error in the WIP estimates is less than 2%; thus, the results of the algorithm yield very acceptable results.

Table 9.8 Comparison of the analytical and simulation results

	Analytical			Simulation		
	th	WIP	CT	th	WIP	CT
System	0.5823/hr	6.798	11.674 hr	0.588/hr	6.892	11.717 hr
Workstation 1	0.5823/hr	3.391	5.823 hr	0.588/hr	3.447	5.861 hr
Workstation 2	0.5823/hr	1.784	3.063 hr	0.588/hr	1.807	3.072 hr
Workstation 3	0.5823/hr	1.623	2.787 hr	0.588/hr	1.638	2.785 hr

The results of our analysis indicate that the WIP in each workstation is significantly below the kanban limits set for system control. One reason for this is that the job preparation time to initiate each job to the factory is on the same order as the process times. This is established so that the computations would result in numbers that could easily be checked. It is often the case that the rate at which Machine 0 operates would be significantly greater than the workstation processing rates.

9.3.5 Generalizations

Serial flow networks only were considered in this presentation. General feed-forward flow networks, that is acyclic flow only with no feedback branching, were studied by Lee and Pollock [10], and general networks that also allow backward branching were studied by Jun and Perros [9]. This latter problem class encounters the phenomenon called dead-locking and these systems are difficult even to simulate (see Deuermeyer et al. [5] and Venkatesh et al. [16]).

- *Suggestion: Do Problems 9.6–9.8.*

9.4 Setting Kanban Limits

A significant problem associated with the implementation of a WIP limiting control strategy for factory operations is the setting of the kanban or WIP limits. This problem has been studied in the literature for special cases [2, 7, 8, 14] and Chap. 7 of the book by Papadopoulos, Heavey and Browne [12] discusses the results and characterizations of the structural properties found in the literature to that date. A recent analysis by Spinellis, Papadopoulos and Smith [15] uses simulated annealing as the optimization tool to find the buffer settings for long production lines. Heuristic methods (such as simulated annealing, tabu search and genetic algorithms (see [3]) are particularly suited to the optimization of this type of problem due to the combinatorial and stochastic nature of the problem. These methods are called meta-heuristics. According to Glover and Laguna [6]: “A meta-heuristic refers to a master strategy that guides and modifies other heuristics to produce solutions beyond that normally generated in a quest for local optimality.”

The problem is to find the individual buffer capacities (workstation WIP limits) that maximize the throughput for a given total allocation of buffer units for a serial system of workstations. The maximum throughput for a system without a total buffer-units limit is obtained by infinite queues allowed at each machine in the serial configuration. To make the problem realistic, the objective has been taken to find the optimal allocation of a fixed number of buffer units. By allocation is meant the number of buffer units to assign to each of the machines (workstations). So by fixing the total number of units available, the allocation of these units to the various machines so as to maximize the system throughput is a well-defined problem. Then the question of how many total units to allow can be answered based on a secondary criterion such as an upper limit on the mean cycle time or reaching a minimum throughput level.

In this section, a scheme is developed for obtaining very good, if not optimal, buffer level configurations. Researchers such as Altiok and Stidham [2] conjecture that the response function (throughput) is smooth and convex in nature. The example problem discussed below demonstrates that this function is not actually convex for all cases. Thus, the solution methodology must deal with local maxima that are not the global maximum. The general search strategy is to use a neighborhood search procedure for finding local maxima in conjunction with a restart procedure designed to explore the solution space beyond these local maxima. The underlying throughput evaluation methodology is the decomposition approach (mean-value response generator) discussed in this chapter. The approach developed herein can be viewed as a particular application of tabu search, but apparently the complex meta-heuristic structure commonly used for non-convex combinatorial problems is not needed. For example, the approach in [15] of using simulated annealing is much more complex than appears necessary for the buffer allocation problem. Their heuristic methodology, however, allows for the simultaneous optimization of buffer allocations and machine processing rates. This combined optimization is a much more difficult problem that requires this more powerful approach.

9.4.1 Allocating a Fixed Number of Buffer Units

For a given number of buffer space units, the problem is to find the best allocation of these units across the workstations so as to maximize the system throughput. Since this total must remain constant, it seems reasonable to use an exchange algorithm where a single unit is taken from one workstation and assigned to another. Then the throughput for this new configuration is evaluated. The basic step of the algorithm is to evaluate all single units exchanges (both positive and negative) for each pair of workstations (this is called a cycle). A cycle results in $n(n-1)$ evaluations for a n workstation problem. The best configuration for all these exchanges is stored as the current best (incumbent solution) and the process repeated. If the best exchange value is not better than the incumbent solution, then the process has reached a local maximum. In this way a local search is performed with the best configuration being used as the base point for further explorations (cycles). For concave functions this local search procedure converges to the global maximum.

Once a local maximum has been obtained, the pair-wise exchange of a single unit of buffer space between two workstations cannot find a better point and each additional cycle will terminate again with this same solution. To allow the exploration to continue, a restart procedure is initiated once a local maximum has been identified. The restart procedure implemented herein is to start the unit-exchange process from the local maximum with this configuration's throughput value set to zero (an incumbent value of zero). This allows the unit-exchange process to find the second best point in the neighborhood as the solution obtained during the cycle since the starting point (configuration) cannot be generated by this exchange procedure. The neighborhood search (cycle procedure) continues from this solution. This one-unit offset from the local maximum allows the neighborhood search procedure to explore a slightly different region than it could reach from the local maximum point. If the local concave nature of this local maximum is not too broad, the next cycle has a chance of finding a better solution and continuing the search at a higher level than was obtained via the local maximum. If the solution obtained by the restart procedure is the same local maximum, then the search process is terminated with that point as its maximal configuration. The algorithm in pseudo-code is given below. In this pseudo-code, *BP* is the base policy and BP' is the one-unit offset permutation policy obtained by routine $Permutation(\pm 1, BP)$, *maxthru* is the current best throughput value, and *maxpolicy* is the associated policy. The routine "evaluate BP'" solves the decomposition and obtains the system performance throughput value $thru(BP')$. The algorithm is started with a buffer allocation *BP* whose sum determines the total number of units to be allocated.

```
Algorithm
start: $BP \leftarrow \{b_1, b_2, \cdots, b_n\}$
    $maxthru \leftarrow 0$
    $maxpolicy \leftarrow BP$
    $found \leftarrow 0$
    $holdthru \leftarrow 0$
cycle: $BP \leftarrow maxpolicy$
    Repeat
        $BP' \leftarrow Permutation(\pm 1, BP)$
        evaluate $BP'$
        If $thru(BP') > maxthru$ Then
            $maxthru \leftarrow thru(BP')$
            $maxpolicy \leftarrow BP'$
        EndIf
    Until $BP' = \emptyset$
    If $maxthru > holdthru$ Then
        $holdthru \leftarrow maxthru$
        GoTo cycle
    EndIf
local: $found \leftarrow found + 1$
    If $found = 1$ Then
        $maxthru \leftarrow 0$
        GoTo cycle
    EndIf
Stop: Print $maxthru$, $maxpolicy$
```

Since there is no closed-form relationship that describes the throughput rate as a function of the buffer configuration, it cannot be determined analytically whether or not this throughput function is concave. Most of the optimization approaches used in the literature for the buffer allocation problem will only isolate local maxima. The simulated annealing approach of Spinellis, Papadopoulos and Smith [15] being an exception. Experience with problems of 3, 5, and 7 workstations in series leads us to believe that this relationship is "almost" concave. Only one problem was found that exhibited a local solution which was not the global solution. This particular problem is used in the illustration below. This local maximum was only about 0.04% larger than the best neighbor point and, thus, this throughput function is very close to being a concave function. It is interesting to note that to solve this particular problem by exhaustive search of the whole solution space is beyond reason since there are 736,281 configurations that would have to be evaluated.

The overall search procedure presented above is a simple implementation of a tabu search method (see [6]). To summarize, a local optimal point is considered "tabu" for one (or more) exchange-evaluation cycle(s) and the next best point in the neighborhood is found. From this point, the one-unit exchange process might find a better point than this local maximum and, thus, continue to improve without hanging up on the local maximum. Another approach for moving away from a local maximum would be to perform an exhaustive search around the local maximum with some specified radius. This can be accomplished for all exchanges that are within a specified number of units away from the base point. Then, the process would restart from the new best point and hopefully be free to find a better maximum.

The number of possible configurations c for a problem with n workstations (machines) and b total buffer units to be allocated among these workstations is a combinatorial problem and is computed as

$$c = \binom{b+(n-1)}{n-1} .$$

For example, a seven workstation series system with a total of 25 buffer units to be allocated across these workstations results in 736,281 possible configurations. Using exhaustive search to solve problems of this size is computationally prohibitive. In addition, the state space to model this system can be as large as 40,000 states (allocations of the form $\{3,4,4,4,4,3,3\} \rightarrow 5^4 4^3$ states). So it is also unreasonable to model this system without using a decomposition approach such as the one discussed in this chapter.

Example 9.2. Consider a seven workstation series system with a total of 25 buffer units to be allocated. The processing times used for this problem are increasing from the first to the last workstation. The mean processing time vector for the seven workstations is $\{0.9,1.0,1.1,1.2,1.3,1.4,1.5\}$ and the order generating machine has a rate of 1 job per hour. The starting buffer units allocation configuration is $\{4,4,5,3,3,3,3\}$. From this starting point, all the one-unit pair-wise exchanges (42 of them) are evaluated. To illustrate the 12 exchanges with the first workstation are

$$\begin{gathered}
\{3,5,5,3,3,3,3\},\{3,4,6,3,3,3,3\},\{3,4,5,4,3,3,3\},\\
\{3,4,5,3,4,3,3\},\{3,4,5,3,3,4,3\},\{3,4,5,3,3,3,4\},\\
\{5,3,5,3,3,3,3\},\{5,4,4,3,3,3,3\},\{5,4,5,2,3,3,3\},\\
\{5,4,5,3,2,3,3\},\{5,4,5,3,3,2,3\},\{5,4,5,3,3,3,2\} .
\end{gathered}$$

The 10 exchanges with the second workstation (12 minus two duplicates exchanges with workstation one) are

$$\{4,3,6,3,3,3,3\},\{4,3,5,4,3,3,3\},\{4,3,5,3,4,3,3\},$$
$$\{4,3,5,3,3,4,3\},\{4,3,5,3,3,3,4\},$$
$$\{4,5,4,3,3,3,3\},\{4,5,5,2,3,3,3\},\{4,5,5,3,2,3,3\},$$
$$\{4,5,5,3,3,2,3\},\{4,5,5,3,3,3,2\}\ .$$

This process is continued until all exchanges have been made and evaluated. There are a total of 42 pair-wise exchanges and throughput-rate evaluations in this cycle. The highest throughput rate was 0.51529/hr for configuration $\{3,4,5,3,3,3,4\}$. Using this configuration now as the base point, the process (cycle) is repeated. The cycle results are shown in Table 9.9. Note that the cycle process is continued until the same solution is repeated (for Cycles 6 and 7). The local maximum throughput of 0.55546/hr is obtained on the sixth cycle and the seventh cycle is needed to determine that this point is a local maximum.

Table 9.9 Cycle results from first pass for Example 9.2

Cycle	Throughput	*CT*	Configuration
0			$\{4,4,5,3,3,3,3\}$
1	0.51529	38.313	$\{3,4,5,3,3,3,4\}$
2	0.52834	36.255	$\{2,4,5,3,3,4,4\}$
3	0.53703	34.712	$\{2,3,5,3,3,5,4\}$
4	0.54499	32.104	$\{2,2,5,3,3,5,5\}$
5	0.55235	31.395	$\{2,2,4,3,4,5,5\}$
6	0.55546	30.058	$\{2,2,3,3,4,6,5\}$
7	0.55546	30.058	$\{2,2,3,3,4,6,5\}$

The resulting configuration $\{2,2,3,3,4,6,5\}$ is a local maximum with a throughput rate of 0.55546/hr. The restart process initiates from this configuration with the incumbent throughput rate set to zero. The cycle results for the second pass of the search process are shown in Table 9.10.

Table 9.10 Cycle results from second pass for Example 9.2

Cycle	Throughput	*CT*	Configuration
0			$\{2,2,3,3,4,6,5\}$
1	0.55536	27.935	$\{1,2,3,3,5,6,5\}$
2	0.55575	29.162	$\{1,2,3,4,5,5,5\}$
3	0.55585	27.790	$\{1,2,3,4,4,5,6\}$
4	0.55585	27.790	$\{1,2,3,4,4,5,6\}$

Repeating the restart process a third time results in no improvement. Thus, the search procedure is terminated with a (hopefully global) maximum throughput of 0.55585/hr. The optimal 25 buffer space allocation for this seven-workstation problem is the configuration $\{1,2,3,4,4,5,6\}$. The whole search process started from the initial configuration $\{4,4,5,3,3,3,3\}$. If the optimization procedure is started from $\{3,3,4,5,4,3,3\}$ instead of $\{4,4,5,3,3,3,3\}$, the process does not hang up at a local maximum and proceeds directly to the 0.55585/hr throughput solution. □

9.4.2 Cycle Time Restriction

Sometimes the allocation of buffer units must be accomplished under the restriction that the mean cycle time for the factory is less that some pre-specified quantity. This is a one-dimensional search problem and sophisticated techniques can be used, however, it is easy to merely search over the allocation total using a decrement size. This approach is based on the assumption that the cycle-time relationship is a monotonic function of the total units to be allocated. Under this assumption, once the proper value has been covered (a result above and a result below the desired cycle time) then the increment can be decreased and the process repeated. The complete solution process illustrated above must be used to obtain the maximum throughput value for each specified total units available for allocation to the seven workstations.

Example 9.3. Suppose for the seven-workstation example problem that the goal is to maximize the throughput while maintaining a cycle time that is less than 25 hours. Then from the above example analysis, it is obvious that less than the 25 buffer units should be used, since the 25 units allocation results in a cycle time of 27.790 hr. Since the example problem result is near the desired cycle time (27.79 verses 25) a small step-size increment can be used. Again for illustration purposes, a total-units step-size decrement of two will be used. The results for the complete optimization analyses for each total allocation until the goal cycle time of 25 time units has been met are as shown in Table 9.11.

Table 9.11 Results with cycle time restricted to less than 25 hr

Total	Throughput	*CT*	Configuration
25	0.55585	27.790	$\{1,2,3,4,4,5,6\}$
23	0.54473	26.659	$\{1,2,3,3,4,5,5\}$
21	0.53015	25.122	$\{1,2,2,3,4,5,4\}$
19	0.51522	23.995	$\{1,2,2,3,3,4,4\}$

It should be obvious even without performing the analysis for the total allocation quantity of 19, that a total of 20 units should satisfy the goal. The cycle time result at 21 total units is very near the 25 time units desired, hence, a drop of one more unit should reduce the cycle time below the goal of 25 time units. And this is the observed result; at 20 units to allocate the results are according to Table 9.12. Thus, the cycle time function does appear to be a monotone decreasing function of the total units allocated (for the maximal configuration). Therefore, the proposed one-dimensional search procedure should result in the maximal throughput configuration subject to the cycle time being less than or equal to some specified level. □

Table 9.12 Results for an allocation of 20 units

Total	Throughput	*CT*	Configuration
20	0.52339	24.819	$\{1,2,2,3,4,4,4\}$

There are situations where cycle time is not a monotone function of the total units allocated and special care needs to be taken when solving these problems. This non-monotone phenomenon occurs when there are equal service times for all the machines and for the three workstations in series systems illustrated in the section.

9.4.3 Serial Factory Results

Several serial factory configurations are studied. First a seven-workstation system with equal processing times is addressed. This is followed by studying the optimal buffer allocation configurations for all permutations of three processing rates assignments to machines. These results indicate the optimal buffer allocations are reasonably stable regardless of the position in the series of the bottleneck machines for a three workstation structure.

Example 9.4. Consider first a seven-workstation serial system with all service times equal (1 time unit each). The results for this system for 23-36 buffer units available for allocation are displayed in Table 9.13. There are two interesting aspects for this system. First, there are ties in the maximal throughput configurations for three of the allocation totals (23, 25 and 32 units), but the cycle times are quite different for the allocations. Additionally, the cycle times for these tied throughput values are not consistent with similar total allocations in that they are not monotone increasing with respect to the total units allocated. The second interesting aspect of these results is the deviation from a concave form for the number of units optimally allocated across the serial workstations. This concave structure has been reported in the literature [2, 15]. For this example there frequently are dips of one unit in the middle workstation's allocation quantity below those allocated to adjacent workstations. These dips get filled in when the number of available units is increased by one (28, 30 and 35 units).

□

Example 9.5. The optimal individual buffer units allocations for three workstations in series for all permutations of three mean service times (exponentially distributed) are given in Table 9.14. The mean processing times are (4/3, 5/4, 6/5). The optimal allocations are reasonably stable for all permutations of these times. A permutation of these times means that Workstation 1 will be assigned one of the three values, Workstation 2 is assigned one of the remaining two values, and Workstation 3 is assigned the last unassigned value. A permutation is denoted like (1, 2, 3), where 1 means the first processing time mean (4/3), 2 represents the second mean value (5/4) and 3 represents the third mean value (6/5). This short-hand notation is necessary

Table 9.13 Optimal throughput configurations for a serial system with equal processing times for several total buffer units allocations

Total	Throughput	Cycle Time	Allocation
23	0.65858	20.70	$\{3,3,3,4,4,3,3\}$
23	0.65858	21.74	$\{3,3,4,4,3,3,3\}$
24	0.66905	21.68	$\{3,3,4,4,4,3,3\}$
25	0.67661	21.19	$\{3,3,4,4,4,4,3\}$
25	0.67661	23.24	$\{3,4,4,4,4,3,3\}$
26	0.68645	22.67	$\{3,4,4,4,4,4,3\}$
27	0.69370	23.18	$\{3,4,4,5,4,4,3\}$
28	0.70039	23.70	$\{3,4,5,4,5,4,3\}$
29	0.70694	24.23	$\{3,4,5,5,5,4,3\}$
30	0.71260	24.75	$\{3,5,5,4,5,5,3\}$
31	0.71961	25.23	$\{3,5,5,5,5,5,3\}$
32	0.72495	24.81	$\{4,4,5,5,5,5,4\}$
32	0.72495	26.71	$\{4,5,5,5,5,4,4\}$
33	0.73190	26.23	$\{4,5,5,5,5,5,4\}$
34	0.73758	26.73	$\{4,5,5,6,5,5,4\}$
35	0.74273	27.23	$\{4,5,6,5,6,5,4\}$
36	0.74795	27.73	$\{4,5,6,6,6,5,4\}$

to present this table as one unit. There are six permutations of these mean times, so six different serial systems are analyzed for each total buffer units allocation from 3 to 12 units. For some totals, there are two distinct optimal allocations. That is, the optimal policy is not the same for all six systems, but never more than two different policies for a given number of total units to be allocated. So frequently there will be two entries (rows) in the table for a given total quantity. Only the optimal systems have throughput values displayed in the table.

These results are very consistent for all total allocations and the optimal allocation structure is concave across the workstations. The cycle time verses total units allocated is particularly unusual for these three-workstation systems in that it is not necessarily monotone increasing with increasing total units allocated. To illustrate using the service time permutation (2,1,3) system, the cycle times at the optimal throughput configurations are longer for lower total units for the totals 4&5, 7&8, 10&11, 13&14, and 16&17 (see Table 9.15). The throughputs and average *WIP* levels are monotone increasing functions of the total units allocated while the cycle time function is not.

□

Problems

9.1. Consider a process that takes an exponentially distributed time with a mean of 1.5 hours to run on a machine. However, before each run the machine must be checked for debris and 25% of the time it is found that the machine must be cleaned before the job can be processed. The cleaning time is also exponentially distributed

Table 9.14 Optimal buffer units allocations for three workstations in series with six service time permutations

Allocation	(1,2,3)	(1,3,2)	(2,1,3)	(2,3,1)	(3,1,2)	(3,2,1)
3:{1,1,1}	0.412	0.413	0.411	0.415	0.412	0.414
4:{1,2,1}	0.452	0.453	0.451	0.453	0.451	0.452
5:{1,2,2}	0.480	0.482	0.481	0.486	0.484	0.487
6:{2,2,2}	0.506	0.508	0.505	0.511		
6:{1,3,2}					0.507	0.510
7:{2,3,2}	0.532	0.533	0.530	0.534	0.530	0.532
8:{2,3,3}	0.549	0.552	0.550	0.556	0.552	0.556
9:{2,4,3}	0.566	0.568	0.566	0.672	0.568	0.571
10:{3,4,3}	0.581	0.583	0.579			
10:{2,4,4}				0.586	0.581	0.586
11:{3,5,3}	0.593					
11:{3,4,4}		0.596	0.593	0.600	0.594	0.599
12:{3,5,4}	0.606	0.609	0.605	0.612	0.606	
12:{3,4,5}						0.610

Table 9.15 Optimal configuration throughput, *WIP* and cycle time results for a serial three workstation system with processing times (5/4,4/3,6/5)

Total	Allocation	Throughput	*WIP*	Cycle Time
3	{1,1,1}	0.41159	2.684	6.521
4	{1,2,1}	0.45089	3.328	7.382
5	{1,2,2}	0.48102	3.446	7.163
6	{2,2,2}	0.50538	4.489	8.882
7	{2,3,2}	0.52982	5.143	9.707
8	{2,3,3}	0.54950	5.248	9.550
9	{2,4,3}	0.56639	5.856	10.338
10	{3,4,3}	0.57867	6.949	12.008
11	{3,4,4}	0.59246	7.053	11.905
12	{3,5,4}	0.60493	7.695	12.721
13	{3,6,4}	0.61400	8.340	13.584
14	{3,6,5}	0.62356	8.404	13.478
15	{4,6,5}	0.63149	9.530	15.092
16	{4,7,5}	0.63870	10.204	15.976
17	{4,7,6}	0.64626	10.281	15.908
18	{4,8,6}	0.65221	10.936	16.768

but with a mean time of 30 seconds. Counting the cleaning and processing time as the operation time, what is the mean and SCV of this operation time?

9.2. Consider a sequence of three machines each with exponentially distributed processing times. The mean processing rates for these three machines are 1, 2, and 3, jobs per hour, respectively. Consider the time between completed jobs at Machine 3 (the third machine in the series). The system operates as follows. With probability 5/8, machine three has a next job ready for processing immediately after completing a job. Sometimes there isn't a next job waiting to be processed at Machine 3 but there is a job processing on Machine 2 which when completed will be sent immediately to Machine 3. This situation occurs 1/4 of the time. When there is not a job being processed on Machine 2 when Machine 3 needs a next job for processing, there is always a job processing on Machine 1. Thus, Machine 3 must wait for the completion of the job on Machine 1 and then its processing time on Machine 2 before it is available for processing on Machine 3. What is the mean and SCV of the time between job departures from Machine 3?

9.3. Obtain the parameters (μ, a, γ) of the GE_2 distribution fit to the following data:
(a) $E[T] = 2.5$, and $C^2[T] = 0.75$.
(b) $E[T] = 3.5$, and $C^2[T] = 2.0$.
(c) $E[T] = 3$, and $C^2[T] = 1$.

9.4. Obtain the steady-state probabilities for a $GE_2/GE_2/1/2$ system where an arrival to a full system blocks the arrival process, and with inter-arrival time distribution parameters $(\alpha, p, \beta) = (1, 1/4, 2)$ and service time distribution parameters $(\mu, q, \gamma) = (1, 1/2, 3)$. (Notice that the maximum number of jobs allowed in the system is 2.) Also assume that a blocked arrival stops the arrival process. Note that this system has 12 probability states.
(a) What are the probabilities that there will be 0, 1, and 2 jobs in the system?
(b) What is the probability that the arrivals to this system are blocked?
(c) Counting a blocked arrival as an extra job in this system, what are the probabilities that there will be 0, 1, 2, and 3 jobs in the system?

9.5. Obtain the steady-state probabilities for a $GE_2/GE_2/1/1$ system where an arrival to a full system blocks the arrival process, and with inter-arrival time distribution parameters $(\alpha, p, \beta) = (1.5, 1/3, 3)$ and service time distribution parameters $(\mu, q, \gamma) = (2, 1/6, 4)$. (Notice that the maximum number of jobs allowed in the system is 2.) Also assume that a blocked arrival stops the arrival process. Note that this system has 8 probability states.
(a) What are the probabilities that there will be 0, and 1 jobs in the system?
(b) What is the probability that the arrivals to this system are blocked?
(c) Counting a blocked arrival as an extra job in this system, what are the probabilities that there will be 0, 1, and 2 jobs in the system?

9.6. Consider a two workstation serial flow system with total units limits of 2 at each workstation. Assume that all times are exponentially distributed. Let the mean time to prepare jobs for entry into the factory is one hour, and the let mean service rates be 4/3 and 5/4 jobs/hour for Workstations 1 and 2, respectively. Develop the workstation throughput estimates after the first full cycle of the decomposition procedure (forward and backward passes).

9.7. Develop the workstation throughput estimates after the second and third full cycles of the decomposition procedure (forward and backward passes) of Problem 9.6.

9.8. Consider a three workstation serial flow system with total units limits of (3, 2, 3) at the workstations. Assume that all times are exponentially distributed. Let the mean time to prepare jobs for entry into the first workstation be 60 minutes, and let the mean service rates be 1.2, 1.3, and 1.1 jobs per hour for the three workstations, respectively. Develop the workstation throughput estimates for the first five full cycles of the decomposition procedure (forward and backward passes).

References

1. Altiok, T. (1996). *Performance Analysis of Manufacturing Systems*. Springer-Verlag, N.Y.
2. Altiok, T., and Stidham, Jr., S. (1983). The Allocation of Interstage Buffer Capacities in Production Lines. *IIE Transactions*, **15**:292–299.
3. Arrts, E., and Lenstra, J.K., editors (1997). *Local Search in Combinatorial Optimization*. John Wiley & Sons.
4. Dallery, Y., and Frein, Y. (1993). On Decomposition Methods for Tandem Queueing Networks with Blocking. *Operations Research*, **41**:386–399.
5. Deuermeyer, B. L., Curry, G.L., Duchowski, A.D., and Venkatesh, S. (1997). An Automatic Approach to Deadlock Detection and Resolution in Discrete Simulation Systems. *INFORMS Journal on Computing*, **9**:195–205.
6. Glover, F., and Luguna, M. (1997). *Tabu Search*, Kluwer Academic Publishers.
7. Hillier, F.S., and So, K.C. (1991). The Effect of the Coefficient of Variation of Operation Times on the Allocation of Storage Space in Production Line Systems. *IIE Transactions*, **23**:198–206.
8. Hillier, F.S., and So, K.C. (1995). On the Optimal Design of Tandem Queueing Systems with Finite Buffers. *Queueing Systems*, **21**:245–266.
9. Jun, K.P., and Perros, H.G. (1988). Approximate Analysis of Arbitrary Configurations of Queueing Networks with Blocking and Deadlock. *Queueing Networks with Blocking: Proceedings of the First International Conference Workshop*, Raleigh, NC, pp. 259–279.
10. Lee, H. S., and Pollock, S.M. (1990). Approximation Analysis of Open Acyclic Exponential Queueing Networks With Blocking. *Operations Research*, **38**:1123–1134.
11. Neuts, M.F. (1981). *Matrix-Geometric Solution in Stochastic Models*, Johns Hopkins University Press, Baltimore.
12. Papadopoulos, H.T., Heavey, C., and Browne, J. (1993). *Queueing Theory in Manufacturing Systems Analysis and Design*. Chapman & Hall, London.
13. Perros, H. (1994). *Queueing Networks with Blocking*. Oxford University Press, N. Y.
14. Smith, J.M., and Daskalaki, S. (1988). Buffer Space Allocation in Automated Assembly Lines. *Operations Research*, **36**:343–358.
15. Spinellis, D., Papadopoulos, C. and Smith, J.M. (2000). Large Production Line Optimization using Simulated Annealing. *Int. J. Prod. Res.*, **38**:509–541.
16. Venkatesh, S., Smith, J., Deuermeyer, B.L., and Curry, G.L. (1998). Deadlock Detection and Resolution for Discrete-Event Simulation: Multiple-Unit Seizes. *IIE Transactions*, **30**:201–216.

Appendix A
Simulation Overview

Simulation is an important technique used by an analyst to validate or verify suggested improvements for manufacturing processes and to verify that suggested conditions or configurations satisfy design specifications. In this appendix we give a brief overview of some of the basic concepts used to develop simulations, especially simulations involving time. Simulations not involving time are often referred to as Monte Carlo simulations; however, the majority of this appendix is devoted to a discussion of clock management while simulating processes involving time.

To simulate complex systems, specialized simulations languages are available; however, our purpose here is not to enable the reader to build complex, realistic simulations for which specialized languages are needed. Our intent is to give the reader an idea of what is involved in simulations and to provide the capabilities of building simple examples. Several of the chapter appendices have already presented some simple simulation models. This appendix is to be used by the interested reader if there is further interest in slightly more complex simulations than have already been discussed. The interested reader can find a good summary of simulation in [1, Chaps. 2 and 9] and a comprehensive discussion in [2].

We also remind the reader that simulations are statistical experiments, and thus the results do not yield deterministic values. Whenever results are reported from a simulation study, it is important to also provide some idea of the variability of the estimates. One approach is to always report confidence intervals (see p. 99) together with the statistical estimates obtained from the simulation.

A.1 Random Variates

Random numbers refer to streams of real values between 0 and 1 that give the appearance of being stochastically independent and uniformly distributed between zero and one. Almost all computer languages and most calculators have some function that will generate random numbers. On a calculator, usually a key marked "RND" will generate a different random number every time it is pushed. In Ex-

G.L. Curry, R.M. Feldman, *Manufacturing Systems Modeling and Analysis*, 2nd ed.,
DOI 10.1007/978-3-642-16618-1_10, © Springer-Verlag Berlin Heidelberg 2011

cel, the function RAND() will generate a different random number every time it is used.

A random variate is a generalization of a random number to an arbitrary distribution other than uniform between 0 and 1. The principal mechanism for generating random variates is the recognition that if U is a uniform random variable between 0 and 1 and F is a CDF, then

$$X = F^{-1}(U) \tag{A.1}$$

is a random variable that is distributed according to F, where F^{-1} is the inverse of F if it exists and it is defined by $F^{-1}(y) = \min\{t|F(t) \geq y\}$ for $0 \leq y \leq 1$ if the inverse does not exist. In other words, to generate a random variate according to the distribution function F, a random number is first generated and then the inverse of the CDF is evaluated at the value specified by the random number.

For continuous random variates, Excel has several inverse distributions as built-in functions. Table A.1 (taken from [1]) lists the associated Excel function that is used for generating random variates from the listed distributions. Some of the listed Excel functions have parameters that must be supplied with numerical values. These parameters are listed using the notation from the corresponding equation as shown in the table.

Table A.1 Excel functions for some continuous random variates

Distribution	Equation #	Excel Function
Uniform	(1.14)	a + (b-a)*RAND()
Exponential	(1.15)	$-(1/\lambda)$*LN(RAND())
Gamma	(1.19)	GAMMAINV(RAND(), α, β)
Weibull	(1.20)	β*(-LN(RAND()))^(1/α)
Standard Normal		NORMSINV(RAND())
Normal	(1.21)	NORMINV(RAND(), μ, σ)
Log Normal	(1.23)	LOGINV(RAND(), μ_N, σ_N)

When using any of these functions within a cell, do not forget to type the equal sign before the function. As a reminder, the standard normal distribution is a normal distribution with mean zero and variance one. It might also be noted that in the authors' experience, the random variate for the gamma distribution with shape parameter less than one ($\alpha < 1$) does not appear to be very accurate with respect to goodness of fit tests.

For discrete random variables, again it is important that the cumulative distribution is used and not the mass function. For example, let N have a mass function given by $\Pr\{N=2\} = 0.5$, $\Pr\{N=3\} = 0.3$, and $\Pr\{N=4\} = 0.2$. To generate a random variate according to this mass function, a nested if statement could be used as the following Excel portion shows:

	A	**B**
1	random number	random variate
2	=RAND()	=IF(A2<0.5,2, IF(A2<0.8,3,4))

A.2 Event-Driven Simulations

The most common method for keeping track of time within a simulation model of a process involving time is a next-event time-advance mechanism. Conceptually, a list of known "future" events is maintained and whenever a time advance is necessary, this *future events list* is searched for the future event with the minimum time of occurrence and then the internal simulation clock time is advanced to the time of this "future" event. In order to build this future event's list, *entities* are created representing items that move through the system being simulated. For example, in a simulation study of the Panama Canal, entities may represent ships. If an airport is being simulated to better understand congestion at security points, passengers would be entities. If a drive-in window facility at a bank is being simulated, entities may represent arriving vehicles.

Within an event-driven simulation, there is always one active entity. If there is more than one entity within the simulation (and there are usually many entities), all other entities are called passive entities. Passive entities are always maintained on either the future event's list or on a queue list. There may be several queue lists within a simulation but there is always one and only one future event's list. The future event's list contains a list of those entities whose next future event is known. Most simulations are initialized by one or more arrival streams of entities. The general steps for an event-driven simulation are as follows:

1. Set the clock time to zero.
2. Initialize all variables and the system state.
3. Determine the arrival time for the initial entity of each arrival stream and place that entity on the future event's list. Keep the future event's list sorted so that the top entity has the minimum time associated with its future event.
4. Remove the top entity from the future event's list and increase the simulated clock time to the time of its future event. This entity now becomes the active entity.
5. If the event of the active entity is an arrival, generate the next arriving entity and place it on the future event's list remembering to keep the list sorted by the timing of its future events. If the event is the "stop" event, the simulation would stop.
6. Update the system state according to events generated by the active entity until an event causes the entity to become passive. The events that cause an entity to become passive are to place the entity back onto the future event's list, place the entity on one of the queue lists, or to dispose of the entity. Maintain the various statistics for the desired system descriptions.
7. Return to the Step 4. If the future event list is empty, the simulation is finished.
8. When the simulation is finished, do the final statistical calculations and output the desired systems statistics.

Example A.1. **By-hand example.** Consider an M/M/1 queueing system with a mean arrival rate of four per hour, a mean service rate of five per hour, and with the first arrival to the system occuring at time 0. As shown in the Appendix of Chap. 3, the M/M/1 system can be simulated without the use of a future event's list; however, the simulation in that appendix cannot be generalized to a multi-server system; whereas, if we use a future event's list, it can be generalized to a multi-server system.

Our system is a relatively easy system to represent since it is necessary to only keep track of the number of customers (entities) in the system. For example, if four customers are present, then the service facility is occupied and the queue contains three customers. The simulation will represented by a table, where each row of the table represents the state of the system at the specified clock time. There is one arrival stream so to initialize the system we generate the first arrival which has a value of 0 according to the system description.

We also observe that there are two relevant events in the queueing system; namely, an arrival (to be denoted by `A`) to the system and a departure (to be denoted by `D`) from the server. When an entity is placed on the future event's list, it will be represented by an ordered pair giving its event type and the time at which it will be removed from the list. Thus, the simulation is initialized by setting the clock time to 0, setting the number of customers in the system to 0, and placing one entity in the future event's list which is represented as $\{(\mathtt{A}, 0.0)\}$ where entries to the future event's list are ordered with the first component being the event that will be executed when the entity is removed from the future event's list and the time of that the entity is to be removed.

The first step of the simulation is to remove the entity from the future event's list and "advance" the clock time to 0.0. Since this entity represents an arrival to the system, the next arriving entity is immediately generated. Since the inter-arrival times are exponentially distributed, the arrival time is obtained by generating a random number, taking the natural log of that number, and multiplying it by 15 minutes (see the second row of Table A.1), and adding the generated inter-arrival time to the current clock time. This future arrival is then placed on the future event's list. For this example, our random number was 0.628 which generated a value of 6.978 representing the next arrival to the queueing system. Returning to time zero (our current clock time), we add one to the system and then generate a service time since the arriving entity will enter the server. We generate 0.416 as the random number which yields a service time of 10.525 which completes the simulation at time zero. The table describing these steps is shown in Table A.2. After finishing the description of the system with clock time 0, the next entity to become active is pulled from the future event's list and the clock time is advanced according to the next event's future time. Since the next event is an arrival again, another entity is created and the time of its arrival is immediately generated and placed on the future event's list. Notice that the future event's list is arranged so that the event's are listed in increasing order of their future event's time. Because the entity that arrives at time 6.978 finds the

Table A.2 Results at clock time = 0.0 yielding a future event's list $\{(\mathtt{A}, 6.978), (\mathtt{D}, 10.525)\}$

Clock Time	Event Type	Random Number	Next Arrival	# in System	Random Number	Next Departure
0.000	`A`	0.628	6.978	1	0.416	10.525

server busy, it must be placed on the queue list and no service time is generated. This results in Table A.3. The next time advanced yields a clock time of 10.525 and because this event is a departure from the server, the entity that was placed in the queue is moved to the server and another service time (equal to 4.178 in our example) is generated to create a future event at time 6.978 + 4.178 = 10.525 (with some round-off error). Notice that the time of departure equals the service time plus the clock time, see Table A.4. Continuing in the same manner for the next three events will yield the Table A.5 which you should use to verify your understanding of the process. To continue this example, the next advance of the clock time will move the clock to 48.637 with another arrival.

Table A.3 Results at clock time = 6.978 yielding a future event's list $\{(\mathtt{D}, 10.525), (\mathtt{A}, 16.083)\}$

Clock Time	Event Type	Random Number	Next Arrival	# in System	Random Number	Next Departure
0.000	A	0.628	6.978	1	0.416	10.525
6.978	A	0.545	16.083	2	—	—

Table A.4 Results at clock time = 10.525 yielding a future event's list $\{(\mathtt{D}, 14.703), (\mathtt{A}, 16.083)\}$

Clock Time	Event Type	Random Number	Next Arrival	# in System	Random Number	Next Departure
0.000	A	0.628	6.978	1	0.416	10.525
6.978	A	0.545	16.083	2	—	—
10.525	D	—	—	1	0.706	14.703

Table A.5 Results at clock time = 39.001 yielding a future event's list $\{(\mathtt{A}, 48.637), (\mathtt{D}, 62.442)\}$

Clock Time	Event Type	Random Number	Next Arrival	# in System	Random Number	Next Departure
0.000	A	0.628	6.978	1	0.416	10.525
6.978	A	0.545	16.083	2	—	—
10.525	D	—	—	1	0.706	14.703
14.703	D	—	—	0	—	—
16.083	A	0.217	39.001	1	0.021	62.442
39.001	A	0.526	48.637	2	—	—

Since this is a "by-hand" example that is meant to illustrate the concepts, we shall stop at this point. However, it is important to remember that a simulation is a statistical experiment so that if the goal was to actually simulate this system, it would be necessary to continue the example for a long time and then repeat it for multiple replications. □

Before moving to Excel, the method for determining the average number of entities in the system (WIP) must be given. If a plot of the number of entities in the system versus time is created, the average number of entities in the system is obtained by determining the area under that curve divided by the total time. These calculations are shown in Table A.6, where a column has been added to represent the area under the WIP curve that is added each time the clock jumps ahead. The quantity in the area column is the time difference from the first column multiplied by WIP that was in the system during that time interval, e.g., to obtain the final column of the third row, we have $7.093 = (10.525 - 6.978) \times 2$ and for the final row we have $22.918 = (39.001 - 16.083) \times 1$. Thus, for this small example, the estimate for the number of entities in the system is given as 41.167/39.001 = 1.056, where 41.167 is the sum of the areas contained in the final column. This is, of course, one data point. To develop a confidence interval, the simulation would have to be repeated several times to obtain a random sample representing the WIP in the system and then the techniques described on p. 99 could be used for the confidence interval.

Table A.6 Results at clock time = 39.001 yielding a future event's list $\{(\texttt{A},48.637),(\texttt{D},62.442)\}$

Clock Time	Event Type	Random Number	Next Arrival	# in System	Random Number	Next Departure	Area
0.000	A	0.628	6.978	1	0.416	10.525	0
6.978	A	0.545	16.083	2	—	—	6.978
10.525	D	—	—	1	0.706	14.703	7.093
14.703	D	—	—	0	—	—	4.178
16.083	A	0.217	39.001	1	0.021	62.442	0
39.001	A	0.526	48.637	2	—	—	22.918

Example A.2. **Excel example.** We now consider an Excel example involving two servers, namely, an M/M/2 queueing system with unequal servers. Using a future event's list with Excel is a little awkward; specifically, some nested if statements will be required that may need patience in reading them. Future event's list simulations are best written with a programming language, but it is possible to demonstrate the concept using Excel. There will be three types of events for this simulation: an arrival, a departure from the first server, and a departure from the second server.

Arriving customers are according to a Poisson process with mean rate four per hour (15 minute inter-arrive times). The first service facility can process an average of 3 per hour (average service time of 20 minutes) and the second service facility can process an average of 2 per hour (average service time of 30 minutes), and arriving customers can go to either (but not both) of the service facilities. To setup the spreadsheet, first select the first row and then click the "Wrap Text" icon on the "Home" tab of the ribbon. Type the following in the first two rows.

	A	**B**	**C**	**D**
1	Clock Time	Event Type	Time of Next Arrival	Number in System
2	0	A	=-15*LN(RAND())	1

	E	F	G	H	I
1	Service Time-1	Time of Depart-1	Service Time-2	Time of Depart-2	Area Under Curve
2	`=-20*LN(RAND())`	`=A2+E2`	0	--	0

In order to build the future rows, we use the following formulas in row 3.

```
Column A   =MIN(C2,F2,H2)
Column B   =IF(A3=C2,"A",IF(A3=F2,"D1","D2"))
Column C   =IF(B3="A",A3-15*LN(RAND()),C2)
Column D   =IF(B3="A",D2+1,D2-1)
Column E   =IF(OR(AND(B3="A",F2="--"),AND(B3="D1",D3>1)),
             -20*LN(RAND()),"--")
Column F   =IF(E3<>"--",A3+E3,IF(B3="D1","--", F2))
Column G   =IF(D2=0,"--",IF(OR(AND(B3="A",H2="--"),
             AND(B3="D2",D3>1)),-30*LN(RAND()),"--"))
Column H   =IF(G3<>"--",A3+G3,IF(B3="D2","--",H2))
Column I   =(A3-A2)*D2
```

(Note that the formulas in cells E3 and G3 are long formulas and should be typed on one line; the line feed in the above description is due to the width of the printed page and should not be included in your formula.) The future event's list is always contained in columns C, F, and H. The final step of the simulation is to copy the formulas in Cell A3:I3 down for several thousand rows and the simulation is complete. Type "`Avg.WIP`" in Cell K1 and type

```
=SUM(I:I)/MAX(A:A)
```

in Cell K2 to obtain an estimate for the time-averaged value of WIP for the simulation. □

Example A.3. **Coxian example.** The next queueing example is to incorporate a Coxian distribution (see Fig. 3.4) into the previous example. Specifically, we simulate a M/G/2 system with the processing time for the first server having a mean of 30 minutes and an SCV of 1.0 and the processing time of the second server having a mean of 30 minutes and an SCV of 0.8. Using the formulas given by Eq. (3.15), the second server can be described by a two-phase system with the first phase being exponential having a mean of 24 minutes, the second phase being exponential having a mean of 15 minutes, and a probability of 0.4 of going from the first to the second phase and a probability of 0.6 of finishing after the first phase.

The set-up for Excel is very similar to the previous example except that three extra columns will be inserted in the table immediately before Column G. In other words, Columns A through F are exactly the same as Example A.2, and Columns G through L would be as

	G	H	I	J	K	L
1	Phase-1 Time	Continue to Next Phase?	Phase-2 Time	Service Time-2	Time of Depart-2	Area Under Curve
2				0	--	0

with row 3 having the following formulas that would need to be copied down.

Column G	`=-24*LN(RAND())`
Column H	`=IF(RAND()<0.4,1,0)`
Column I	`=-15*LN(RAND())`
Column J	`=IF(D2=0,"--",IF(OR(AND(B3="A",K2="--"), AND(B3="D2",D3>1)),G3+H3*I3,"--"))`
Column K	`=IF(J3<>"--",A3+J3,IF(B3="D2","--",K2))`
Column L	`=(A3-A2)*D2`

Columns G through I simulate the components of the Coxian distribution, and Column J using the formula `G3+H3*I3` to insert the Coxian distribution for the service time whenever it is needed. □

Example A.4. Our final example is to verify the formulas used to adjust the mean and SCV of services times when equipment is used that is not 100% reliable. Namely, Eqs. (4.3) and (4.4) are used to obtain the effective mean and SCV of the service time for a processor whose availability is less than 100%. Our goal is to simulate failures and repairs on equipment and the resulting service times under the assumption that a failure halts services and then after a repair is complete service resumes where it was interrupted. We will keep the example general so that the simulated service times can be easily obtained for different parameter sets.

The initial model will be to determine the effective service time for a processor whose time to failure is exponentially distributed with a mean time between failures 4 hours. The time to repair has a gamma distribution with a mean of 1 hour and an SCV of 2. If there is no interruption of service, then the service time has a gamma distribution with a mean of 2 hours and an SCV of 0.4. To set the stage, the basic data is given in Cells A2:B7 and Eqs. (4.3) and (4.4) are in Cells A10:B11.

	A	B
1	Given Data	
2	Avg Fail	4
3	Avg Repair	1
4	Avg Service	2
5	SCV Fail	1
6	SCV Repair	2
7	SCV Service	0.4
8	Calculated Data	
9	Availability	`=B2/(B2+B3)`
10	Effective Mean	`=B4/B9`
11	Effective SCV	`=B7+(1+B6)*B9*(1-B9)*B3/B4`

Notice that the SCV for the time until failure (Cell B5) must be one for the formulas of (4.3) and (4.4) to be accurate; however, we leave it general so you can try other approximations and see the results of non-exponential failures. The simulation is generated in Columns C–G, the statistic collection occurs in Columns H–I and O–Q; finally, Columns K–M contain be basic random times for failures, repairs, and nominal service times. Columns J and N are left blank to provide some separation of the numbers.

	C	**D**	**E**	**F**	**G**	**H**	**I**
1	Event Type	Clock Time	Time to Failure	Repair Time	Service Time	Start Service	Actual Service
2	`serve`	0	`= K2`	0	`= M2`	0	

To make the formulas that follow easier to understand (and shorter), the random variates for the failure, repair, and service times are determined in separate columns; namely, in Columns K, L, and M.

	K
1	Time to Failure
2	`= GAMMAINV(RAND(),1/$B$5,$B$5*$B$2)`

	L
1	Repair Time
2	`= GAMMAINV(RAND(),1/$B$6,$B$6*$B$3)`

	M
1	Service Time
2	`= GAMMAINV(RAND(),1/$B$7,$B$7*$B$4)`

As usual, the main work of the simulation is contained in the third row. These formulas are given next with their explanation following.

Column C	`=IF(C2="fail", "repair", IF(E2<G2,"fail","serve"))`
Column D	`=IF(C3="fail",E2,IF(C3="serve",G2,F2))`
Column E	`=IF(C3="fail",0,IF(C3="serve",E2,D3+K3))`
Column F	`=IF(C3="fail",D3+L3,0)`
Column G	`=IF(C3="fail",L3+G2,IF(C3="serve",D3+M3,G2))`
Column H	`=IF(C3="serve",D3, H2)`
Column I	`=IF(C3="serve",H3-H2," ")`

There are three types of events: a failure has just occured, a repair has just been completed, or a service has just been completed. To understand the `If` statement in Cell C3, consider the following logic. If a failure has just occured, then the next event must be a repair; otherwise, both service and another failure are "in process" so the next event depends on which one occurs first. Column D contains the clock time and the clock time depends on the event that caused the clock to advance as shown in Column C. The time to the next failure (Column E) is updated only when the event causing the clock to advance is the completion of a failure since that is the only time that a new failure time begins. The only time that the time to repair (Column F) is relevant is when the event causing the clock advance is a failure. Finally, the only time that the service time (Column G) needs to be updated is when a service is completed. Columns H and I are only for statistical collection purposes. The goal is to determine the effective service times which equals the difference between successive service completion times, and this is what is contained in Column I. To complete the simulation, Cells C3:M3 should be copied down for 25000 (or more) rows.

The final statistical estimates from the simulation are contained in Columns O–Q. Notice that Cells O3:Q3 contain the estimators for effective service times and thus the value in O3 should be compared to the value in B10 while the value in Q3 should be compared to the value in B11.

	O	**P**	**Q**
1	Mean	St.Dev.	SCV
2	Effective Service Time		
3	=AVERAGE(I:I)	=STDEV(I:I)	=(P3/O3)^2
4	Time to Failure		
5	=AVERAGE(K:K)	=STDEV(K:K)	=(P5/O5)^2
6	Repair Time		
7	=AVERAGE(L:L)	=STDEV(L:L)	=(P7/O7)^2
8	Nominal Service Time		
9	=AVERAGE(M:M)	=STDEV(M:M)	=(P9/O9)^2

The values in O5:Q5, O7:Q7, and O9:Q9 are only given as a check on the initial distributions. If the simulation is proper, then these value can be compared to the appropriate values in Column B to verify the initial data.

□

References

1. Feldman, R.M., and Valdez-Flores, C. (2010). *Applied Probability & Stochastic Processes*, Second Edition, Springer-Verlag, Berlin.
2. Law, A.M. (2007). *Simulation Modeling and Analysis*, Fourth Edition, McGraw-Hill Book Company, New York.

Glossary

availability The long-run average fraction of time that the processor is available for processing jobs, denoted by a (p. 113).

cellular manufacturing The concept of organizing the factory into sub-factories with the capability to produce a technology group (p. 177).

closed queueing network A network of queues in which no arrivals are possible from outside the network and no jobs within the network can leave (p. 242).

coefficient of variation (CV) The standard deviation divided by the mean; usually restricted to positive random variables (p. 13).

conditional probability The probability of event A given B is $\Pr(A \cap B)/\Pr(B)$ if $\Pr(B) \neq 0$ (p. 2). Also used for random variables when information of one random variable is known and the distribution of the other random variable is desired (p. 27).

CONWIP A production control strategy in which a constant level of work-in-process is maintained within the facility and thus a form of pull-release control is used for jobs entering the system but not at each workstation (p. 241).

correlation coefficient The covariance of two random variables divided by the product of the two standard deviations (p. 30).

covariance The expected value of the product of the difference of one random variable and its mean multiplied by the difference of the second random variable and its mean (p. 29).

cumulative distribution function (CDF) A function associated with a random variable giving the probability that the random variable is less than or equal to the specified value (p. 5).

cycle time The time that a job spends within a system. The average cycle time is denoted by CT (p. 46).

G.L. Curry, R.M. Feldman, *Manufacturing Systems Modeling and Analysis*, 2nd ed.,
DOI 10.1007/978-3-642-16618-1, © Springer-Verlag Berlin Heidelberg 2011

effective arrival rate The rate at which jobs enter the system, often denoted by λ_e. Notice that λ often represents the rate that jobs come *to* the system and λ_e represents the rate that jobs are allowed *into* the system (p. 73).

effective processing time The time duration from when a job first has control of a processor or machine until the time at which the job releases the processor or machine so that it is available to begin work on another job; thus, it might include actual processing time plus a setup time or repair time in case of processor failure (p. 113).

event A subset from the sample space, or a set of outcomes (p. 1).

expected value The expected value of a discrete random variable is the sum over all possible values of the random variable times the probability that the value will occur; with continuous random variables, the integral replaces the sum (p. 10).

group technology The analysis of processing operations with the goal of determining the similarity of the processing functions and, hence, the grouping of the associated parts for production purposes (p. 177).

independence Random variables are independent if knowledge of the value of one random variable does not provide any information in predicting the value of the other random variables (p. 7).

indicator function The indicator function for integers is a matrix with the value of 1 on the diagonal and 0 off the diagonal. If the matrix is square, it is an identity matrix (p. 170).

job type Jobs with different routes or different processing characteristics are said to be of different job types (p. 48).

joint distribution function The distribution function associated with two or more random variables (p. 24).

kanban A production control strategy in which a maximum limit on work-in-process at each workstation is maintained and thus a form of pull-release control is used at each workstation (p. 281).

marginal distribution function The distribution function associated with one random variable, usually derived from a joint distribution function (p. 25).

mean The mean of a random variable is its expected value (p. 11).

memoryless property The lack of memory property is usually associated with a random variable that denotes the time at which an event occurs and the property implies that the probability of when the event will occur is the same as the conditional probability of when the event occurs given that the event has not yet occurred (p. 17).

mixture of random variables The probabilistic selection of one random variable among a group of independent random variables (p. 35).

outcome An element of the sample space (p. 1).

offered workload See workload.

Poisson process A renewal process formed by the sum of exponential random variables (pp. 16 and 134).

probability density function (pdf) A function associated with a continuous random variable such that a probability that the random variable is between to values equals the integral of the function between those values (p. 6).

probability mass function (pmf) A function associated with a discrete random variable giving the probability that the random variable equals the independent variable (p. 6).

probability space A three-tuple $(\Omega, \mathscr{F}, \Pr)$ where Ω is a sample space, $\mathscr{F}$ is a collection of events from the sample space, and Pr is a probability measure that assigns a number to each event contained in $\mathscr{F}$ (p. 1).

pull A general control strategy applied to a system that has a limit applied to its work-in-process. After the maximum number of jobs are within the system, further jobs are allowed into the system only when they are "pulled" into the system by other jobs departing from the system (pp. 241 and 267).

push The standard operating assumption for open queueing networks in which jobs enter the system whenever they arrive to the system or according to a schedule independent of the system status (pp. 241 and 267).

random variable A function that assigns a real number to each outcome in the sample space (p. 4).

renewal process A process formed by the sum of nonnegative random variables that are independent and identically distributed (p. 134).

routes The sequence of processing steps for a job (p. 48).

routing matrix A matrix of probabilities, $P = (p_{ij})$, where $p_{i,j}$ is the probability that an arbitrary job leaving Workstation i will be routed directly to Workstation j (p. 139).

sample space A set consisting of all possible outcomes (p. 1).

squared coefficient of variation (SCV) The variance divided by the square of the mean value (usually restricted to positive random variables) (p. 13).

standard deviation The square root of the variance (p. 11).

step-wise routing matrix A routing matrix indicating the probability of moving from processing step to processing step instead of from workstation to workstation (p. 169).

switching rule The probabilities that indicate the probabilistic branching for jobs as they depart from one workstation and get routed to another (p. 139).

throughput rate The number of completed jobs leaving the system per unit of time. The throughput rate averaged over many jobs is denoted by th (p. 47).

variance The variance of a random variable is the expected value of the squared difference between the random variable and its mean. Equivalently, it is the second moment minus the square of the mean (p. 11).

work-in-process The number of jobs within a system that are either undergoing processing or waiting in a queue for processing. The average work-in-process is denoted by WIP (p. 46).

workload The total amount of work that is required of a workstation per unit of time and is determined by the sum of the total arrival rate (per time unit) for each product type multiplied by its associated mean processing time (in time units consistent with the arrival rate) (p. 159).

workstation A collection of one or more identical machines or resources (p. 47).

workstation mapping function Gives the workstation assigned to each step of the production plan (p. 168).

Index