# PLANNING AND CONTROL OF AN UNRELIABLE MACHINE IN A MULTI-ITEM PRODUCTION-INVENTORY SYSTEM

by

DAVID BRIAN KLETTER

S.M. Management
Massachusetts Institute of Technology
1994

S.B. Management Science
Massachusetts Institute of Technology
1992

S.B. Mathematics
Massachusetts Institute of Technology
1992

Submitted to the Sloan School of Management
in Partial Fulfillment of
the Requirements of the Degree of
Doctor of Philosophy in Management

at the

Massachusetts Institute of Technology
June 1996





Signature of Author _____
MIT Sloan School of Management
March 18, 1996

Certified by _____
Stephen C. Graves
Professor of Management Science
Thesis Supervisor

Accepted by _____
Birger Wernerfelt
Chair of the Doctoral Program Committee

PLANNING AND CONTROL OF AN UNRELIABLE MACHINE IN A MULTI-ITEM PRODUCTION-INVENTORY SYSTEM

by

DAVID B. KLETTER

Submitted to the Alfred P. Sloan School of Management
on March 18, 1996, in partial fulfillment
of the requirements of the Degree of
Doctor of Philosophy in Management

## ABSTRACT

The management of manufacturing operations is a complex and difficult task due to the dynamic and stochastic nature of most manufacturing systems. Motivated by applications at metal stamping plants, we study a single machine, multiple part make-to-stock production system with setups, where machine reliability is a key source of uncertainty. This thesis is divided into three distinct parts. First, we derive the moments, probability density functions, cumulative distribution functions and Laplace transforms for the number of parts produced over a fixed time interval, and for the amount of time required to produce a fixed number of parts on a machine that experiences random failures and random repair times.

Secondly, we study the operational decision of when to run overtime on an unreliable machine. Given a fixed production schedule and known requirements over a finite horizon, we formulate a dynamic program to determine when (and how much) overtime to use. We show how to compute the sensitivity of the optimal policy to the input parameters, and how this information can be used for rescheduling. In the special case where multiple parts share a single demand point, we present a model that determines the cost minimizing overtime quantity in the presence of stochastic demand.

Lastly, we compare the performance of several different replenishment policies for controlling a single unreliable machine with setups, in an effort to obtain a better understanding of the strengths and weaknesses of different policies in different environments. Our model of the production process includes three types of variability: demand, production, and waiting for setup crews. We use real data from two production lines at a General Motors metal stamping plant in simulations of the policies. For the lines studied, we find the classic lot size/reorder point policies to have relatively superior performance to non-reorder point-based methods of control. A special variant of this policy that considers the availability of setup crews is also shown to be effective.

Thesis Supervisor: Stephen C. Graves
Title: Professor of Management Science

## ACKNOWLEDGMENTS

My deepest appreciation is extended to Professor Steve Graves of MIT, my advisor, my teacher and my friend... for his endless patience – for teaching me so many things – for always believing in me – and for all the wonderful years.

My heartfelt thanks to Dr. Dave Vander Veen of General Motors' Metal Fab Division for suggesting this research topic, and for his support of this research from the beginning to the end. I am indebted for the countless hours spent discussing and refining these ideas, and especially for exposing me to the real-world problems that formed the basis for this work.

I am equally indebted to Dr. Bill Jordan of the GM NAO R&D Center, who suggested and helped formulate the study described in Chapter 4. My exposure to and appreciation for these problems would not have been possible without the time that I spent working under Bill at GM. I will always be grateful for that opportunity.

I owe much to Drs. Jeff Alden and Dave Kim of GM NAO R&D Center for their work on models of an unreliable machine that were the inspiration and basis for the work of Chapter 2. My special thanks to Jeff for his many suggestions and insights, and also for helpful comments on an earlier draft of Chapter 2.

Dr. Don Rosenfield and Professor Vien Nguyen of MIT endured the burden of reading earlier drafts of this thesis. Their patience, care and insight have not only contributed greatly to this thesis, but have taught me many things. I thank them both for all their hard work.

Last, but certainly not least, to Dad, Mom, and Laura, for their endless support, love and encouragement. Without you, this certainly would not have been possible.

The author gratefully acknowledges the support and resources made available to him through the Leaders for Manufacturing Program, a partnership between MIT and major U.S. manufacturing companies.

Dedicated to MIT

With all my respect, admiration and gratitude

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

# 1. Introduction

Most manufacturing operations are extremely complex. Production typically involves multiple products which are composed of multiple different raw materials, often processed by multiple production stages by a variety of different workers with different skills, inventoried at multiple points along the way, and shipped to multiple customers, sometimes from several locations using several different means of transportation. Managing such systems is made more difficult by the dynamic and stochastic nature of most production environments, as well as the many interdependencies that arise among the various decisions that must be made. Effective management, however, is critical, since manufacturing decisions can have a substantial impact on cost, and therefore, on competitiveness.

The field of production management largely concerns itself with the question of how to configure and operate manufacturing facilities. It has long been recognized that the answer to this question depends on the time horizon under consideration (Anthony, 1965). At one extreme, the longest horizon activity is *strategic planning* (one year or more), in which most of the above mentioned areas can be affected, at least to some degree. These include decisions regarding products, facilities, capital, resources and policies. Medium term planning is sometimes called *tactical planning* (one month or more), in which only a very limited subset of decisions can be affected. Usually, decisions about resource levels and to some degree, policies and facilities, can be made. *Operational control* (one week or less) is the shortest horizon activity, in which only small changes in production resources are possible. At this level we are typically concerned with how to most efficiently utilize the available resources to meet certain goals, such as filling customer orders or minimizing cost. Clearly, the time horizons which are appropriate for each of these three categories

may vary by industry, but the general ideas should remain basically the same. For example, in aircraft assembly, a month may be a fairly short period of time compared to the manufacturing leadtime, while in certain electronic industries, a month may be a long period of time relative to the life-cycle of the product.

If we accept the above framework as valid for thinking about production management, then we can conclude that different models may be appropriate for different time horizons, e.g., a tactical planning model is not likely to be appropriate for operational control. Indeed, this view is also consistent with the concept of hierarchical planning (Hax and Meal, 1975), in which we acknowledge that we can not globally optimize the entire system, and instead attempt to achieve a good solution by solving the problem in a hierarchical fashion.

Figure 1.1 presents a proposed hierarchy in which important production decisions must be made at each level. In the next section, we will present a brief overview of our work and describe how it fits within this hierarchy.

**Overview**

This thesis is divided into three distinct parts: (i) analysis of a machine that is subject to random failures and random repair times, (ii) study of the operational decision of when to run overtime on an unreliable machine, and (iii) comparison of the performance of several different policies for controlling a production/inventory system when the machine is unreliable. The common thread that connects them is the focus on the planning and control of a single unreliable machine with setups that produces multiple products to stock. We now discuss each in turn.

The first part (Chapter 2) analyzes a machine where the times between failures are i.i.d. exponential and times to repair are i.i.d. exponential. Two random variables are defined for study: the amount of machine time required to produce a fixed number of parts, and the number of parts produced over a time interval of fixed length. A relationship between these two random variables is identified and exploited. In total, we derive the moments, probability density functions, cumulative distribution functions and Laplace transforms for the two random variables. We believe several of these results to be new. These results are used extensively in subsequent chapters.

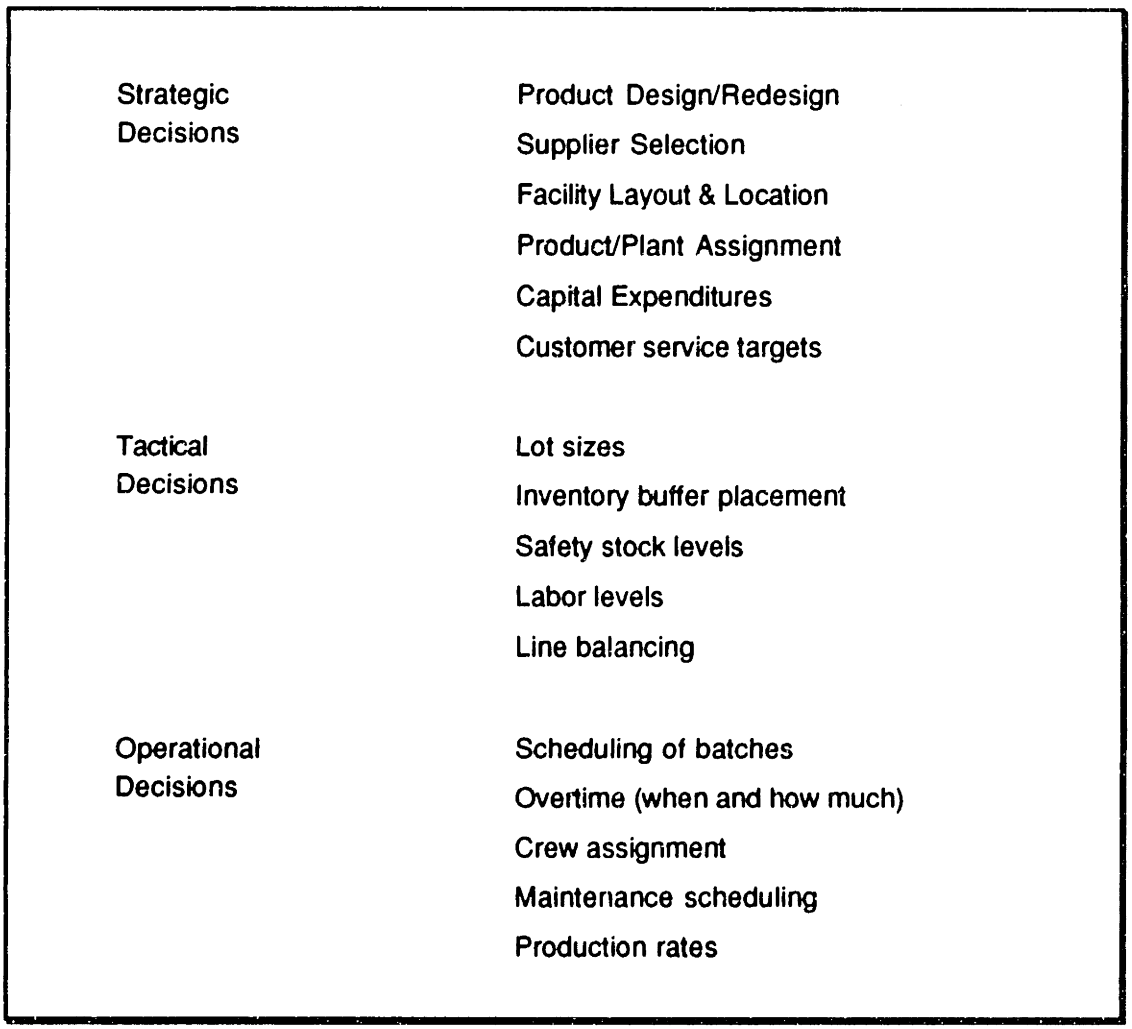


**Figure 1.1** Partial decision hierarchy

The second part of this thesis (Chapter 3) uses the probabilistic results from the first part to study the operational decision of when to run overtime on an unreliable machine. We characterize this model as operational for two reasons. First, we will assume that *tactical* decisions (such as the setting of lot sizes and reorder points) have already been made. Second, we will consider only a finite and short time horizon, such as a day or a week. This will have two important implications. First, we assume that this horizon is short enough so that perfect demand information is available. We will view demand as occurrences at particular known points in time, in known quantities. Secondly, we assume that within this time horizon, random machine failures and random repair times can have significant impact on the output of the production stage, and are the greatest source of variability over this short horizon.

The development of this model is motivated by applications at GM metal stamping plants. These models could be used as part of a manufacturing control system in such a manufacturing operation. One can (and should) envision these models embedded in a software tool that would receive data in real-time from the shop floor and assist plant management in decision making.

We begin by showing how to evaluate a production schedule for an unreliable machine when the requirements over a finite horizon are known. Considering overtime opportunities at fixed points in time and of fixed size, a dynamic program is formulated to determine when overtime should be used. This basic model is extended to include variable sized overtime opportunities, selection among a set of overtime opportunities, and constraints on the amount of overtime that can be used over a given time interval. We show how to compute the sensitivity of the

optimal policy to the input parameters, and how this information can be used for rescheduling. In the special case of a single demand point that is shared by all of the parts, we present a model that determines the cost minimizing overtime quantity in the presence of stochastic demand.

The third part of this thesis (Chapter 4) looks at the issue of selecting a policy for operating a single stage production/inventory system with setups. These are long-term planning decisions, since changing from one operating policy to the next may be quite difficult. By comparing the performance of several different policies, our goal is to obtain a better understanding of the strengths and weaknesses of different policies in different environments. Indeed, we will see that the selection of the operating policy can have a significant impact on the performance of the system.

This study considers only replenishment policies that base production decisions on the quantity of inventory that has been depleted, rather than policies that base production decisions on forecasts of future demand (e.g., MRP). We provide a framework for classifying replenishment policies, and enumerate the possible policies suggested by this framework. Of the 14 policies suggested, we select seven for detailed study. In terms of the production process, three types of variability are included: demand, production, and waiting for setup crews. We propose a set of metrics for comparison of the policies, and use basic analytical reasoning to compare and contrast the policies. For further inferences of phenomena that are difficult to estimate, we turn to simulation of the policies. For these simulations, real data from two production lines at a General Motors metal stamping plant are used.

**Context and literature review**

In the 1980's, many authors recognized the need for new models of manufacturing planning and scheduling. Wagner (1980) outlined many areas of production and inventory theory that were lacking in applicability to practical problems. For example, Wagner noted the lack of planning models that account for uncertainties. Graves (1981) and Abraham et al. (1985) expressed disappointment with the scheduling literature for its focus on static and deterministic problems, as most every real-world problem is both dynamic and stochastic. McKay et al. (1988) echoed this sentiment, and validated it by means of a survey of practitioners, through a case study, and in seminars with real-world schedulers. Abraham et al. (1985) also identified a need for "fresh modeling approaches" to model production systems with disruptions, and to integrate production scheduling and planning activities.

The work presented here has been developed partly in response to these calls for new models. The control models presented in Chapter 3 are inherently dynamic and stochastic in nature, addressing the important real-world issue of how much overtime should be used, and when. We also describe how our models can support rescheduling. In Chapter 4, we study operational control policies for production environments in which there is considerable uncertainty, including machine reliability.

Fortunately, our efforts are not the first. Since the time that the above papers were written, numerous other authors have studied problems in stochastic scheduling, real-time dynamic control, and overtime decisions. We will not review all of this literature here; each chapter will present its own literature review. At this point, we single out two papers in particular that are closely related to our work. As

described earlier, the models in Chapter 3 require as input a production schedule and information about the current state of the system, and are designed to be embedded as part of a real-time decision support tool that could be used on the shop floor. Bean et al. (1991) describe a model that fits this description for the case of make-to-order (MTO) systems with disruptions, with the objective of minimizing total tardiness. Similarly, Gallego (1990) presents a model of make-to-stock (MTS) systems that operate according to a cyclic schedule, and finds an approximate cost minimizing strategy to recover the cyclic schedule after a disruption. While these authors consider a broad class of discrete disruptions (such as the unexpected arrival of additional demand), our models will assume that the major source of variability in the system is machine unreliability and thus "disruptions" are often almost continuously occurring. Further, our treatment is unique in that we consider the option to run overtime to help recover from disruption.

In summary, the portfolio of models presented here attempt to contribute to an area of the literature in which there is an undesirable gap between theory and practice. Further, we hope that the models and framework presented will serve as a good starting point for others to continue research in this area.

**Structure of this thesis**

As described above, the next three chapters will present the three major parts of this thesis. Although these chapters are intended to be readable independent of one another, Chapters 3 and 4 make extensive use of the results in Chapter 2. We will refer the reader to the relevant sections of Chapter 2 when necessary. Each chapter will present its own literature review and contain its own references. In Chapter 5 we identify some opportunities for further research.

## References for Chapter 1


Abraham, Chacko, Brenda Dietrich, Stephen Graves, William Maxwell and Candace Yano. "Factory of the Future: A Research Agenda for Models to Plan and Schedule Manufacturing Systems". Report of Ad Hoc Committee, 1985.

Anthony, Robert N. Planning and Control Systems: A Framework for Analysis. Cambridge, MA: Harvard University, Graduate School of Business Administration, 1965.

Bean, James C., John R. Birge, John Mittenthal and Charles E. Noon. "Matchup Scheduling with Multiple Resources, Release Dates and Disruptions". Operations Research, 39(3), pp. 470-483, 1991.

Gallego, Guillermo. "Scheduling the Production of Several Items with Random Demands in a Single Facility". Management Science, 36(12), pp. 1579-1592, 1990.

Graves, Stephen C. "A Review of Production Scheduling". Operations Research, 29(4), pp. 646-675, 1981.

Hax, Arnoldo C. and Harlan C. Meal. "Hierarchical integration of production planning and scheduling," in M. A. Geisler, ed., Studies in Management Sciences, Vol. 1: Logistics, New York: Elsevier, 1975.

McKay, Kenneth N., Frank R. Safayeni and John A. Buzacott. "Job-Shop Scheduling Theory: What Is Relevant?" Interfaces, 18(4), pp. 84-90, 1988.

Wagner, Harvey M. "Research Portfolio for Inventory Management and Production Planning Systems". Operations Research, 28(3), pp. 445-475, 1980.

# 2. A model of an unreliable machine

## Introduction

In this chapter we analyze the following stochastic system: a single machine produces parts at a deterministic rate but is subject to random failures. When the machine fails, it is completely inoperable until it is repaired. Hence, at any time the machine is in one of two states: working or failed. We assume that the times between failures are i.i.d. exponential with mean time between failures (MTBF) equal to $1/\lambda$ and that times to repair are i.i.d. exponential with mean time to repair (MTTR) equal to $1/\mu$. We assume operation-dependent failures (Gershwin, 1994); that is, the machine can not fail while it is under repair or idle.

## Literature review

Reliability has been a topic of active research with origins dating back to the turn of the century (according to Nahmias, 1989). See Shaked and Shanthikumar (1990) for a recent survey of the field. The analysis of a single unit with two operating states is perhaps the simplest problem in the study of the reliability. Within this problem subclass, the case of i.i.d. exponential failure and repair times is the most tractable, and has received virtually independent attention in a variety of fields. The problem has also been studied in the telecommunications literature as the *asymmetric random telegraph signal*, in radioactive physics as a type II counter problem (Bharucha-Reid, 1960), in the engineering literature in the analysis of the output of a resistance-capacitance (RC) filter driven by a random binary process (Munford, 1986), and in the biology literature in the analysis of channels in the nerve membrane (see FitzHugh, 1983), among others. The problem can also be analyzed using many different methodologies; for example, the problem can be viewed as a simple case of a Markov Process, or as an alternating renewal process.

Barlow and Hunter (1961) provide an excellent summary of the known results at that time, based largely on Laplace transform and renewal theory. They derive the transform of the expected number of failures and repairs in [0, T), the asymptotic number of failures and repairs, and the transform of the distribution of the number of failures in [0, T). If one knows the expected number of failures and repairs in [0, T) then the transient or asymptotic *availability coefficient* (the probability that the unit is functioning at a given time) is given by their difference. For the case of exponential repairs and failures, they give closed form expressions for the availability coefficient and the distribution of the number of failures in [0, T). Classic texts such as Barlow and Proschan (1965) and Gnedenko et al. (1969) on reliability, and Cox (1962) on renewal theory derive many of these results.

Barlow and Hunter (1961) also give an expression for the distribution of downtime over [0, T) as an infinite series of the n-fold convolutions of the failure and repair distributions with themselves. They also express the result as an integral in the case of exponential failure and repair times, and for general failure and repair times give the asymptotic distribution as T approaches infinity. These results are all due to Takács (1957a, 1957b, 1959). For exponential failures and exponential repairs, Gnedenko et al. (1969) express the distribution of total operating time over a fixed period [0, T) as a double infinite series.

Lie et al. (1977) give a comprehensive although now somewhat outdated survey and classification of availability models. More recently, Baxter (1985) presents a critical review of the literature on the availability of two-state unit modeled as an alternating renewal process. This paper, in conjunction with Baxter (1981), seem to be the most complete and recent summary of important results, and also "fill some gaps in the

theory". Unique to this paper is a review and extension of the results on waiting times (i.e., the distribution of time until a repair greater than a certain length occurs) and on the alternating renewal process where each repair time is correlated with the previous failure time. Baxter also reviews and extends the theory on point availability and average availability, and criticizes Barlow and Proschan (1965) for their "uncritical" application of the asymptotic approximation for the distribution of availability, citing simulation studies which show that passage to the limit can be extremely slow.

Other important contributions are numerous and are scattered over a variety of works. Baxter (1985) finds expressions for the average availability of an alternating renewal process, and gives the simple result in the case of exponential failure and repair times. The average availability over (0, T] can be used to find the average uptime over (0, T] simply by multiplying by T, and the average repair time by subtracting the average uptime from T. Martz (1971) develops a method which can be used to find the distribution of the average availability over n failure and repair cycles for any failure and repair distributions for which the n-fold convolutions are known. For exponential failures and exponential repairs, FitzHugh (1983) finds both the density and the Laplace transform of the number of failure/repair cycles over a fixed period [0, T). He also cites expressions for the autocorrelation and spectral density of the process, which have been derived in the biology and physics literature. For the general case of alternating renewal processes, Mortensen (1990) finds the Laplace transform for the density of the availability coefficient at time T and the asymptotic autocorrelation of the availability coefficient.

Feller (1971), Brouwers (1986) and Kim and Alden (1992) independently derived the density of time to produce a fixed lot size on a machine operating at a constant speed with exponential failure and repair times. This is equivalent to the density of time until

the total uptime reaches some constant T. These authors express the result as a modified Bessel function, which has important theoretical and practical implications, and are a considerable improvement over the previous result of Gnedenko et al. The latter two works also give a simple expression for the variance as a function of T, which is a trivial result if one recognizes that the process can be viewed as a Compound Poisson process (Ross, 1983).

There has also been considerable work, both exact and approximate, on other failure and repair distributions. For example, Kabak (1969) analyzes the exponential failure and constant repair time problem, finds the average availability over (0, T], and develops an approximation for the variance. Takács (1951) uses his general methods to find the distribution of repair time over [0, T), with the machine either starting in a known state or starting in steady-state. For the case of constant repair times and the family of Weibull failure distributions (which include the exponential), Dickey (1991) derives a double series for the availability coefficient at time T and renewal function (expected number of failures in [0, T)).

In addition, numerical results for general failure and repair distributions can be obtained by the method of Cléroux and McConalogue (1976) and McConalogue (1978, 1981). This method numerically evaluates convolution integrals, so many of the above results can (at least in principle) be obtained by algorithm for the general case. When the Laplace transforms of the failure and repair distributions are explicitly known, they could instead be numerically inverted. See Baxter (1981a, 1981b) for a further discussion. Laplace transform inversion has received considerable attention from many authors; see Krylov and Skoblya (1969), for a survey of the classic methods. The last two decades have seen a variety of more powerful and sophisticated methods using both new and old techniques, such as Fourier series approximation (Crump, 1976, De

Hoog, et al. 1982, Piessens and Huysmans, 1984, Abate and Whitt, 1992), continued fraction expansion (Grundy, 1977), contour integration (Murli and Rizzardi, 1990), and expansion in Laguerre polynomials (Garbow et al. 1988). Many of these codes are available via *netlib* (Dongarra and Grosse, 1987).

Lastly, we note that some work on more complex systems could be used to analyze our (relatively) more simple system. For example, Sericola (1990) develops a closed-form solution for the transient distribution of total time spent in a subset of states of a homogenous Markov process over a fixed period of time [0, T). Our two state Markov process is the most trivial problem to which this method could be applied.

In the sections that follow we explore many of these same question as those described above, sometimes from new perspectives and obtaining some new results. When our results duplicate those of previous works, references will be given in context. In whole, this chapter will present a unified treatment of the results that we will need in other parts of our work.

**Notation, summary of key results, and overview of this chapter**

Density functions will be denoted by a lowercase letter (r), cumulative distribution functions by an uppercase letter (R) and random variables by a bold capital letter (**R**). The Laplace transform of a function g(t) will be denoted by g*(s). We will also use $\mathcal{L}\{\ \}$ and $\mathcal{L}^{-1}\{\ \}$ to denote the Laplace transform and inverse Laplace transform of the expression in brackets. We will use the symbol ☆ to denote the convolution operator, and the symbol ⇔ to represent that the expression on the left is the Laplace transform of the expression on the right. Pr{ } will denote the probability of the event in brackets. When we wish to write the probability that a continuous random variable $\mathbf{X} \in (a, a+dx)$, we will write dens{ $\mathbf{X} = a$ }, since Pr{ $\mathbf{X} = a$ } = 0.

$\lambda$ will denote the failure rate when the machine is working, and $\mu$ will denote the repair rate when the machine is failed. Let $\alpha(\cdot)$ be an indicator function, where $\alpha(\tau) = 0$ if the machine is failed at time $\tau$, and $\alpha(\tau) = 1$ if it is working at time $\tau$.

We will now give an overview of the remainder of this chapter. This overview will also serve to introduce much of the important notation that we will use. As we proceed, we will list some of the key results of this chapter. Many of the equation numbers for these key results are also given, in parentheses.

The purpose of the next seven sections will be to characterize the number of parts produced over a fixed time interval, and the quantity of time required to produce a fixed number of parts. Although it is *numbers of parts* that we are concerned with, we will often derive expressions in terms of machine time. It is important to keep in mind that machine time can be converted to parts by simply multiplying by the production rate, which is assumed to be constant when the machine is working.

The purpose of Sections 2.1 - 2.5 is to derive the PDF, CDF and Laplace transform of the number of parts produced over a fixed time interval, and of the quantity of time required to produce a fixed number of parts. The models in Chapters 3 will require a probabilistic description of the number of parts produced over a fixed time interval. The simulations in Chapter 4 will require the quantity of time required to produce a fixed number of parts. We will see throughout the development that the number of parts produced over a fixed time interval and the quantity of time required to produce a fixed number of parts are very closely related.

We now give a summary of the key results of each section of this chapter. The focus of Section 2.1 will be to derive the probability density function for the uptime of an unreliable machine over an interval of length T. If the machine is working at time 0, we will denote this density as this as this as $f(t; T \mid \alpha(0) = 1)$ and abbreviate it as $f(t; T \mid 1)$. We will show that

$$(3) \quad f(t; T \mid 1) = \left[\lambda\mu t \, \frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}} + \lambda \, I_0\left(2\sqrt{x}\right)\right] e^{-\lambda t - \mu(T-t)} + u_0(T-t)\, e^{-\lambda T}, \qquad 0 \le t \le T$$

where $I_0$ and $I_1$ are modified Bessel functions of orders zero and one, $x = \lambda \mu t (T-t)$, and $u_0(z)$ is the unit impulse (Dirac delta) function which is zero everywhere except for an impulse of mass one at z. We derive similar expressions for the cases where the machine is initially failed (4) or in steady state (5). We then derive the probability density function of uptime over an interval of length T conditional on the initial machine state and the machine state T time units later. In our notation this is $f(t; T \mid \alpha(0) = a, \alpha(T) = b)$, and we will abbreviate this as $f(t; T \mid ab)$. The results for the four possible combinations of beginning and ending machine states are given in (6) - (9). We believe that these results have not appeared in the literature. Unfortunately, these PDFs are not easily integrated, so we will need to obtain the CDF and the Laplace transform by other methods.

In Section 2.2 we derive the Laplace transform of the density function (3). The result is

$$(17) \quad f^*(s, T \mid 1) = \frac{(\lambda+\mu-s)T}{2} \, \frac{\sinh y}{y\, e^h} + \frac{\cosh y}{e^h},$$

where $y = \sqrt{\lambda^2 + 2\lambda\mu + \mu^2 + 2\lambda s - 2\mu s + s^2}\; T / 2$ and $h = (\lambda + \mu + s)\, T / 2$.

We believe that this result is new. We then use this expression to find the transient mean (18) and variance (25). The asymptotic mean and variance follow easily. We derive similar results for the cases where the machine is initially failed or in steady state. Lastly, we derive the transform, mean and variance for the cases where the machine where the initial and terminal machine states are given.

In Section 2.3 we turn our attention to characterizing the time to produce a fixed lot of parts. Since the quantity of machine uptime required to produce a fixed size lot is deterministic, we focus on the probability density function for downtime incurred while producing a fixed size lot. We will denote this PDF by $r(t; b \mid 1)$ where $b = \lambda q/p$, q is size of the lot to be produced, and p is the production rate of the machine when it is not failed. We will show that the Laplace transform of this PDF is

$$(32)\quad r^*(s;\ b \mid 1) \;=\; \exp\left(-b + b\frac{\mu}{s+\mu}\right)$$

which, when inverted, is

$$(33)\quad r(t;b \mid 1) \;=\; u_0(t)\exp(-b) \;+\; \mu\ b\ \exp(-\mu t - b)\ I_1\left(2\sqrt{\mu b t}\right)\ (\mu b t)^{-\frac{1}{2}},\ t \geq 0.$$

Although these results have been derived by Kim and Alden (1992) and others, the approach that we present is different, and although not new, provides insight into a more general problem. We also derive the transform and density for the case where the machine is initially failed.

Sections 2.4 and 2.5 derive expressions for the cumulative distributions of Sections 2.1 and 2.3. In Section 2.4 we show that

$$(36)\quad R(t;b \mid 1) = \exp(-\mu t - b) \sum_{v=0}^{\infty} \left(\frac{\mu t}{b}\right)^{\frac{v}{2}} I_v\left(2\sqrt{\mu b t}\right),\ t \geq 0,$$

and a similar expression for $R(t; b \mid 0)$, where $I_v(z)$ is the modified Bessel function of order v. We believe these results to be new, but have been independently derived by Kim (1994, unpublished).

In Section 2.5 we describe an equivalence between $R(t; b \mid 1)$ from Section 2.4 and $F(t; T \mid 1)$, the CDF corresponding to the density of Section 2.1. Using this equivalence we easily conclude that

$$(39)\quad F(t;T \mid 1) = \begin{cases} 0 & t = 0 \\ 1 - e^{-\mu(T-t)-\lambda t} \sum_{v=0}^{\infty} \left(\frac{\mu(T-t)}{\lambda t}\right)^{\frac{v}{2}} I_v\left(2\sqrt{\mu\ \lambda\ t\ (T-t)}\right) & 0 < t < T \\ 1 & t \geq T. \end{cases}$$

We obtain a similar result for the case where the machine is initially failed.

The remainder of Section 2.5 focuses on the more difficult case where we are given both the initial machine state and the machine state at some future point in time. This is useful in the dynamic decision making context described in Chapter 3, where the decision made at some future point in time may depend on the state of the machine.

We first derive the probability that the downtime while producing a batch of size q is at most t, given that the machine starts working and is also working at time t + q/p, where p is the production speed of the machine. In our notation, this probability is $R(t; \lambda\ q/p \mid \alpha(0) = 1, \alpha(t+q/p) = 1)$, which we abbreviate as $R(t; b \mid 11)$. The result is

$$(41)\quad R(t;\, b \mid 11) = \frac{\mu\, R(t;\, b \mid 1) + \lambda\, e^{-b-\mu t} \sum_{v=0}^{\infty} (-1)^v \left(\frac{\lambda t}{\mu q/p}\right)^{v/2} I_v\left(2\sqrt{\mu b t}\right)}{\mu + \lambda\, e^{-(\lambda+\mu)(t+q/p)}}.$$

We derive expressions for the other three cases (00, 01, 10) as well (42)-(44). An equivalence is between R(t; b | 11) and F(t; T | 11) is described and used to conclude that

$$(45)\quad F(t;\, T \mid 11) = 1 - \frac{\mu\left(1 - F(t;\, T \mid 1)\right) + \lambda\, e^{-\lambda t-\mu(T-t)} \sum_{v=0}^{\infty} (-1)^v \left(\frac{\lambda(T-t)}{\mu t}\right)^{v/2} I_v\left(2\sqrt{\mu\lambda t(T-t)}\right)}{\mu + \lambda\, e^{-(\lambda+\mu)T}}.$$

Similar expressions are derived for the other three cases (46)-(48). We then find

$$(49)\quad R^*(s;\, b \mid 10) = \sum_{n=0}^{\infty} \exp\left(-(\lambda+\mu)n\frac{q}{p} - b + b\frac{\mu}{s + \mu + (\lambda+\mu)\, n}\right) \times \left[\frac{1}{s + (\lambda+\mu)\, n} - \frac{1}{s + (\lambda+\mu)\,(n+1)}\right]$$

and similar expressions for the other three cases (50)-(52). We also show that

$$(53)\quad \mathcal{L}\{\, R(t;\, b \mid 11)\, P_{11}(t+q/p)\,\} = \frac{1}{\lambda+\mu} \exp\left(-b + b\frac{\mu}{s + \mu}\right)\left[\frac{\mu}{s} + \frac{\lambda}{s + (\lambda+\mu)}\right],$$

where $P_{11}(T) = \Pr\{\alpha(T) = 1 \mid \alpha(0) = 1\}$. The left-hand side can be interpreted as $\Pr\{\text{downtime} \le t \;\&\; \alpha(t+q/p) = 1 \mid \alpha(0) = 1\}$. The advantage of (53) is that it does not contain an infinite series like the one in (49). We also obtain similar results for the other three cases. Lastly, equation (53) also allows us to write

(57) $F(q/p; t+q/p \mid 11) =$

$$1 - \frac{1}{P_{11}(t+q/p)} \mathcal{L}^{-1}\left\{\frac{1}{\lambda+\mu} \exp\left(-b + b\frac{\mu}{s+\mu}\right)\left[\frac{\mu}{s} + \frac{\lambda}{s+(\lambda+\mu)}\right]\right\}.$$

We believe all of these results to be new.

Section 2.6 explores the transient effects of initial machine conditions on the mean and variance of uptime over a fixed period of time. In Section 2.7 we investigate the accuracy of using a normal distribution to approximate the distribution of parts produced over a fixed period of time. Our results will confirm those cited by Baxter (1985), namely, that under certain conditions the normal distribution can be a poor approximation even after long time intervals. In Section 2.8 we develop exact and approximate methods for obtaining the distribution of time to produce multiple batches on a single machine, and show how that distribution is equivalent to another distribution of interest. Lastly, in an appendix to this chapter we discuss our experience testing two different Laplace transform inversion algorithms that we use in our empirical work in subsequent chapters.

## 2.1 Distribution of parts over a fixed period of time

In this section we analyze the uptime of a machine over the period of time [0, T) when interarrivals of failures and repairs are exponentially distributed with means $\lambda$ and $\mu$, respectively. This is equivalent to analyzing the number of parts produced if the uptime is scaled by the processing speed p.

### Machine initially working

We will denote the state of the machine at any point in time $\alpha(\cdot)$, where $\alpha(\tau) = 0$ if the machine is failed at time $\tau$, and $\alpha(\tau) = 1$ if it is working at time $\tau$. Let $f(t; T \mid \alpha(0) = 1)$ be the PDF of uptime over [0, T) conditional on the machine working at time 0. We will abbreviate this as $f(t; T \mid 1)$. Let $h(t; T \mid 1)$ denote the PDF of downtime over [0, T) conditional on the machine working at time 0. Note that

$$f(t; T \mid 1) = h(T-t; T \mid 1),$$

since downtime = T – uptime. We will use this relation when we derive expressions for the PDF of uptime by characterizing the amount of downtime.

The PDF $f( t; T \mid 1 )$ has both a continuous and a discrete component. The discrete component is an impulse at T that corresponds to the probability that the machine does not fail over the entire interval of length T. This is the probability that the time of the first arrival in a Poisson process of rate $\lambda$ is greater than T, so that

$$(1) \quad f(T; T \mid 1) = u_0(T-t)\, e^{-\lambda T}.$$

where $u_0(z)$ is the unit impulse (Dirac delta) function, that is, a function that is zero everywhere except for an impulse of mass one at z.

The continuous component corresponds to the density of uptime for $0 \le t < T$. For $0 \le t < T$ we can write

$$f(t; T \mid 1) = h(T-t; T \mid 1) = \sum_{n=1}^{\infty} \text{dens}\{ \text{n failures comprising T-t units of downtime} \}$$

$$= \sum_{n=1}^{\infty} \text{dens}\{ \text{(n failures) and (T–t units of downtime \& machine working at time T)} \}$$
$$+ \sum_{n=1}^{\infty} \text{dens}\{ \text{(n failures) and (T–t units of downtime \& machine failed at time T)} \}.$$

In order for the machine to be working at time T when there are n failures, the $n^{th}$ repair must occur after T-t units of downtime and there must be n failures in the t units of uptime. Note that these events are independent once we have fixed t, the amount of uptime. Similarly, in order for the machine to be failed at time T when there are n failures, the $n^{th}$ failure must occur after t units of uptime and there must be n-1 repairs in the T-t units of downtime (the $n^{th}$ repair has not yet occurred). These two events are also independent once we have fixed t. Therefore,

$$f(t; T \mid 1) = \sum_{n=1}^{\infty} \Pr\{ \text{n failures in t time units} \} \text{dens}\{ n^{th} \text{ repair occurs at time T–t} \}$$
$$+ \sum_{n=1}^{\infty} \Pr\{ \text{n–1 repairs in (T–t) units} \} \text{dens}\{ n^{th} \text{ failure occurs at time t} \}$$

$$= \sum_{n=1}^{\infty} \Pr\{ \text{n arrivals in Poisson process at rate } \lambda t \} \text{dens}\{ \text{time of the } n^{th} \text{ arrival in}$$
$$\text{Poisson process of rate } \mu \text{ is T–t} \}$$

$$+ \sum_{n=1}^{\infty} \Pr\{\, n-1 \text{ arrivals in Poisson process at rate } \mu\,(T-t)\,\} \; \text{dens}\{\, \text{time of the } n^{th}$$

arrival in Poisson process of rate $\lambda$ is $t$ }

for $0 \le t < T$. Substituting the Poisson PMF and Erlang density (Ross, 1983) we obtain

$$\sum_{n=1}^{\infty} \frac{(\lambda t)^n \; e^{-\lambda t}}{n!} \; \frac{\mu^n \; (T-t)^{n-1} \; e^{-\mu(T-t)}}{(n-1)!} + \frac{(\mu(T-t))^{n-1} \; e^{-\mu(T-t)}}{(n-1)!} \; \frac{\lambda^n \; t^{n-1} \; e^{-\lambda t}}{(n-1)!}$$

$$= e^{-\lambda t - \mu(T-t)} \sum_{n=1}^{\infty} \left[ \lambda\mu t \frac{(\lambda\mu t(T-t))^{n-1}}{(n-1)! \; n!} + \lambda \frac{(\lambda\mu t(T-t))^{n-1}}{(n-1)! \; (n-1)!} \right].$$

Letting $x = \lambda \mu t (T-t)$,

$$f(\, t; T \mid 1\,) = e^{-\lambda t - \mu(T-t)} \sum_{n=1}^{\infty} \left[ \lambda\mu t \frac{x^{n-1}}{(n-1)! \; n!} + \lambda \frac{x^{n-1}}{(n-1)! \; (n-1)!} \right], \qquad 0 \le t < T.$$

Each of the two terms in the brackets can be written as terms of modified Bessel functions (of order 1 and 0, respectively),

$$(2) \quad f(\, t; T \mid 1\,) = \left[ \lambda\mu t \; \frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}} + \lambda \; I_0\left(2\sqrt{x}\right) \right] e^{-\lambda t - \mu(T-t)}, \qquad 0 \le t < T$$

where

$$I_0(z) = \sum_{k=0}^{\infty} \frac{\left(\frac{z}{2}\right)^{2k}}{k! \; k!}, \quad I_1(z) = \frac{z}{2} \sum_{k=0}^{\infty} \frac{\left(\frac{z}{2}\right)^{2k}}{k! \; (k+1)!}.$$

The modified Bessel functions $I_0$ and $I_1$ can be computed numerically using a variety of methods. For example, Press et al. (1989) present a polynomial approximation based on Abramowitz and Stegun (1964). More sophisticated methods have been developed by

many authors, including Sookne (1973), Cody (1983), and Boisvert and Saunders (1992). Codes are also provided in most commercial numerical libraries, although many excellent codes are in the public domain and are available via *netlib* (Dongarra and Grosse, 1987).

To complete the derivation of f( t; T | 1 ), we add the continuous and discrete components (1) and (2). In total,

$$(3)\quad f(\,t;\,T \mid 1\,) = \left[\lambda\mu t\,\frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}} + \lambda\; I_0\left(2\sqrt{x}\right)\right] e^{-\lambda t-\mu(T-t)} + u_0(T-t)\,e^{-\lambda T}, \qquad 0 \le t \le T.$$

**Machine initially failed**

Using an analogous argument to the one above, we could derive f(t; T | 0), the PDF of uptime over [0, T) conditional on the machine being failed at time 0. However, with a few simple observations we can more easily obtain the result. First, simply note that the PDF of downtime over [0, T) conditional on the machine being failed at time 0 is described by the same stochastic process as f(t; T | 1), with the roles of $\lambda$ and $\mu$ reversed, that is,

$$h(\,t;\,T, \lambda, \mu \mid 0) = f(\,t;\,T, \mu, \lambda \mid 1\,),$$

Therefore,

$$h(t;\,T \mid 0) = \left[\lambda\mu t\,\frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}} + \mu\; I_0\left(2\sqrt{x}\right)\right] e^{-\mu t-\lambda(T-t)} + u_0(T-t)\,e^{-\mu T}, \qquad 0 \le t \le T.$$

Since downtime = T – uptime, the PDFs of uptime and downtime are simply mirror images of one another, thus

(4) $f(t; T \mid 0) = h(T-t; T \mid 0) =$

$$\left[\lambda\mu(T-t)\,\frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}} + \mu\, I_0\left(2\sqrt{x}\right)\right]e^{-\mu(T-t)-\lambda t} + u_0(t)\, e^{-\mu T}, \qquad 0 \le t \le T.$$

Further, we can conclude that the density of *downtime* conditional on the machine working at time 0 is

$h(t; T \mid 1) = f(T-t; T \mid 1) =$

$$\left[\lambda\mu(T-t)\,\frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}} + \lambda\, I_0\left(2\sqrt{x}\right)\right]e^{-\lambda(T-t)-\mu t} + u_0(t)\, e^{-\lambda T}, \qquad 0 \le t \le T.$$

**Machine initially in steady state**

The PDF of uptime with the initial state of the machine randomized (i.e., starting in steady state) can be written as

$$\Pr\{\text{machine initially working}\}\, f(t; T \mid 1) + \Pr\{\text{machine initially failed}\}\, f(t; T \mid 0).$$

Since the steady-state probability that the machine starts out working is $\mu/(\lambda+\mu)$ and failed is $\lambda/(\lambda+\mu)$ (Ross, 1983), $f(t; T)$ can now be seen to equal

$$f(t; T) = \left[\frac{\mu}{\lambda+\mu}\left(\lambda\mu t\,\frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}} + \lambda\, I_0\left(2\sqrt{x}\right)\right) + \frac{\lambda}{\lambda+\mu}\left(\lambda\mu(T-t)\,\frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}} + \mu\, I_0\left(2\sqrt{x}\right)\right)\right] e^{-\lambda t-\mu(T-t)} +$$

$$\frac{\mu}{\lambda+\mu} u_0(T-t)\, e^{-\lambda T} + \frac{\lambda}{\lambda+\mu} u_0(t)\, e^{-\mu T}, \qquad 0 \le t \le T.$$

which, after simplification, is equal to

$$(5)\quad f(t; T) = \frac{\lambda\mu}{\lambda+\mu}\left((\mu t + \lambda(T-t))\,\frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}} + 2\, I_0\left(2\sqrt{x}\right)\right) e^{-\lambda t-\mu(T-t)} +$$
$$\frac{\mu}{\lambda+\mu} u_0(T-t)\, e^{-\lambda T} + \frac{\lambda}{\lambda+\mu} u_0(t)\, e^{-\mu T}, \qquad 0 \le t \le T.$$

**Density with known starting and terminal machine states**

Lastly, we wish to compute f(t; T | α(0) = a, α(T) = b), the PDF of uptime over [0, T) conditional on the machine being in state a at time 0 and state b at time T. We will abbreviate this as f(t; T | ab). From the development above, the results follow immediately, for example,

$$\text{dens}\{t \text{ units of uptime in } [0, T) \text{ \& machine failed at } T \mid 1\} = \lambda\ I_0\left(2\sqrt{x}\right)\ e^{-\lambda t-\mu(T-t)},$$

so that from the law of conditional probability,

$$\text{dens}\{ t \text{ units of uptime in } [0, T) \text{ \& machine failed at } T \mid 1 \} =$$
$$\text{dens}\{ t \text{ units of uptime in } [0, T) \mid 10 \}\ P_{10}(T)$$

where $P_{10}(T)$ is the probability that the machine is failed at time T given that it is working at time 0. Therefore,

$$(6)\quad f(t; T \mid 10) = \frac{1}{P_{10}(T)}\ \lambda\ I_0\left(2\sqrt{x}\right)\ e^{-\lambda t-\mu(T-t)}, \qquad 0 \le t \le T.$$

By similar reasoning,

$$(7)\quad f(t;T \mid 11) = \frac{1}{P_{11}(T)}\left[\lambda\mu t\,\frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}}\,e^{-\lambda t-\mu(T-t)} + u_0(T-t)e^{-\lambda t}\right], \qquad 0 \le t \le T,$$

$$(8)\quad f(t;T \mid 01) = \frac{1}{P_{01}(T)}\,\mu\, I_0\left(2\sqrt{x}\right)\,e^{-\lambda t-\mu(T-t)}, \qquad 0 \le t \le T,$$

$$(9)\quad f(t;T \mid 00) = \frac{1}{P_{00}(T)}\left[\lambda\mu(T-t)\,\frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}}\,e^{-\lambda t-\mu(T-t)} + u_0(t)e^{-\mu T}\right], \qquad 0 \le t \le T.$$

The four probabilities $P_{00}(T)$, $P_{01}(T)$, $P_{10}(T)$ and $P_{11}(T)$ are well-known and are derived classic texts on reliability, such as Barlow and Proschan (1965). They are

$$(10)\quad P_{10}(T) = \frac{\lambda}{\lambda+\mu}\left(1-e^{-(\lambda+\mu)T}\right),$$

$$(11)\quad P_{11}(T) = 1-P_{10}(T) = \frac{\mu}{\lambda+\mu}+\frac{\lambda}{\lambda+\mu}e^{-(\lambda+\mu)T},$$

$$(12)\quad P_{01}(T) = \frac{\mu}{\lambda+\mu}\left(1-e^{-(\lambda+\mu)T}\right),$$

$$(13)\quad P_{00}(T) = 1-P_{01}(T) = \frac{\lambda}{\lambda+\mu}+\frac{\mu}{\lambda+\mu}e^{-(\lambda+\mu)T}.$$

Note that it is now easy to see that $f(t; T \mid 10) = f(t; T \mid 01)$, as a result of the reversibility of the process.

As an aside, it is interesting to note that through this derivation we can write new expressions for certain integrals of modified Bessel functions. For example, since a probability density function must integrate to one,

$$\int_0^T f(t;T \mid 10)\,dt = 1,$$

so that

$$\int_0^T \frac{1}{P_{10}(T)}\ \lambda\ I_0\left(2\sqrt{\lambda\mu t(T-t)}\right)\ e^{-\lambda t-\mu(T-t)}\ dt\ =\ 1.$$

Rearranging this equation, we obtain

$$\int_0^T I_0\left(2\sqrt{\lambda\mu t(T-t)}\right)\ e^{-\lambda t-\mu(T-t)}\ dt\ =\ \frac{P_{10}(T)}{\lambda}$$

Substituting $P_{10}(T)$, it follows that

$$\int_0^T I_0\left(2\sqrt{\lambda\mu t(T-t)}\right)\ e^{-\lambda t-\mu(T-t)}\ dt\ =\ \frac{1}{\lambda+\mu}\left(1-e^{-(\lambda+\mu)T}\right)$$

and

$$\int_0^T I_0\left(2\sqrt{\lambda\mu t(T-t)}\right)\ e^{-(\lambda-\mu)t}\ dt\ =\ \frac{1}{\lambda+\mu}\left(e^{\mu T}-e^{-\lambda T}\right).$$

## 2.2 Laplace transform and moments of uptime over a fixed time interval

In the development that follows we will compute the Laplace transform of the uptime distribution over the period [0, T), conditional on the machine working at time 0,

$$\mathcal{L}\{ f(t; T \mid 1) \} = f^*(s; T \mid 1) = \int_{t=0}^{T} e^{-st} \; f(t; \; T \mid 1) \; dt,$$

and the first and second moments of the distribution. We will also derive results assuming the machine is initially failed or in steady state. From these results, we will also show how to easily obtain results for the case when both the initial and terminal machine states are known.

### Laplace transform of uptime

We have shown in the previous section that $f(t; T \mid 1)$ is

$$f(t; T \mid 1) = e^{-\lambda t - \mu(T-t)} \sum_{n=1}^{\infty} \left[ \lambda\mu t \frac{(\lambda\mu t(T-t))^{n-1}}{(n-1)!\; n!} + \lambda \frac{(\lambda\mu t(T-t))^{n-1}}{(n-1)!\; (n-1)!} \right] + u_0(T-t)\, e^{-\lambda T}.$$

We will find the Laplace transform of f by breaking it into three separate functions $f_1 + f_2 + f_3$, finding the Laplace transform of each of the three functions, and then summing to obtain the overall transform of f. In particular, define

$$f_1(t, T) = e^{-\lambda t - \mu(T-t)} \sum_{n=1}^{\infty} \lambda\mu t \frac{(\lambda\mu t(T-t))^{n-1}}{(n-1)!\; n!} = \sum_{n=1}^{\infty} \frac{\lambda^n t^n e^{-\lambda t}}{n!} \frac{\mu^n (T-t)^{n-1} e^{-\mu(T-t)}}{(n-1)!}$$

$$f_2(t, T) = e^{-\lambda t - \mu(T-t)} \sum_{n=1}^{\infty} \lambda \frac{(\lambda\mu t(T-t))^{n-1}}{(n-1)!\; (n-1)!} = \sum_{n=1}^{\infty} \frac{\lambda^n t^{n-1} e^{-\lambda t}}{(n-1)!} \frac{\mu^{n-1} (T-t)^{n-1} e^{-\mu(T-t)}}{(n-1)!}$$

$$f_3(t, T) = u_0(T-t)\, e^{-\lambda T}.$$

Further, we will find the transforms of $f_1$ and $f_2$ by finding the transform of each term in the infinite series. To derive the transforms, we require the following

***Lemma.*** If g(t), h(T–t) are two functions and f(t,T) = g(t) h(T–t) then the Laplace transform of f(t,T), denoted by f*(s, T), is given by the inverse Laplace transform (with respect to r and introducing the variable T) of g*(r+s) h*(r).

***Proof.*** First, by definition of a Laplace transform,

$$\mathcal{L}\{ f(t,T) \} = f^*(s, T) = \int_{t=0}^{T} g(t)\ h(T-t)\ e^{-st}\ dt.$$

Now note that

$$f^*(s, T) = \left\{g(t)\ e^{-st}\right\} \star \{h(t)\}$$

where ☆ represents the convolution operator and the parameter of convolution is T. Treating T as a variable and taking the transform of both sides with respect to T and introducing the variable r, we obtain

$$f^{**}(s, r) = g^*(r+s)\ h^*(r).$$

Taking the inverse transform of both sides with respect to r and reintroducing the variable T produces the desired result. ❑

To use the lemma, we write the $n^{th}$ term of $f_1(t, T)$ as $g_1(t)\ h_1(T-t)$, where

$$g_1(t) = \frac{\lambda^n t^n e^{-\lambda t}}{n!}$$

and

$$h_1(T-t) = \frac{\mu^n (T-t)^{n-1} e^{-\mu(T-t)}}{(n-1)!}.$$

We also write the $n^{th}$ term of $f_2(t, T)$ as $g_2(t)\, h_2(T-t)$ where

$$g_2(t) = \frac{\lambda^n t^{n-1} e^{-\lambda u}}{(n-1)!}$$

and

$$h_2(T-t) = \frac{\mu^{n-1} (T-t)^{n-1} e^{-\mu(T-t)}}{(n-1)!}.$$

Note that

$$g_1^*(s) = \frac{\lambda^n}{(\lambda+s)^{n+1}},\ h_1^*(s) = \frac{\mu^n}{(\mu+s)^n},\ g_2^*(s) = \frac{\lambda^n}{(\lambda+s)^n},\ \text{and}\ h_2^*(s) = \frac{\mu^{n-1}}{(\mu+s)^n}.$$

Given these definitions and the above lemma, in order to find the Laplace transform of $f_1(t, T)$ we must find

$$\mathcal{L}\{ f_1(t,T) \} = \mathcal{L}^{-1}\{ \sum_{n=1}^{\infty} \frac{\lambda^n \mu^n}{(\lambda+s+r)^{n+1}(\mu+r)^n} \}$$

and in order to find the Laplace transform of $f_2(t, T)$ we must find

$$\mathcal{L}\{ f_2(t,T) \} = \mathcal{L}^{-1}\{ \sum_{n=1}^{\infty} \frac{\lambda^n \mu^{n-1}}{(\lambda+s+r)^{n}(\mu+r)^n} \}$$

where the inverse transforms are with respect to the variable r and introduce the variable T.

We will first focus on finding the transform of $f_2(t, T)$. In the discussion that follows (_._._) refers to an equation in Abramowitz and Stegun (1964). The first term for n = 1 is trivial. By (29.3.12),

$$\mathcal{L}^{-1}\{\frac{\lambda}{(\lambda+s+r)(\mu+r)}\} = \lambda\left(\frac{e^{-\mu T} - e^{-(\lambda+s)T}}{\lambda+s-\mu}\right).$$

For $n \geq 2$, we have from (29.3.50) that

$$\mathcal{L}^{-1}\{\frac{\lambda^n\mu^{n-1}}{(\lambda+s+r)^n(\mu+r)^n}\} = \sqrt{\pi}\ \frac{(\lambda\mu)^n}{\mu\ (n-1)!}\left(\frac{T}{\lambda+s-\mu}\right)^{n-\frac{1}{2}}\ e^{-\frac{1}{2}(\lambda+s+\mu)T}\ I_{n-\frac{1}{2}}\left(\tfrac{1}{2}(\lambda+s-\mu)T\right)$$

where $I_{n-1/2}$ is a modified spherical Bessel function. We now desire an expression for the sum of these terms from n = 2 to infinity. Writing this sum, and then rearranging the terms and reindexing to begin at n = 1, we obtain

$$\sum_{n=2}^{\infty} \mathcal{L}^{-1}\{\frac{\lambda^n\mu^{n-1}}{(\lambda+s+r)^n(\mu+r)^n}\} =$$
$$\lambda T\ e^{-\frac{1}{2}(\lambda+s+\mu)T} \sum_{n=1}^{\infty} \sqrt{\frac{\pi}{2}}\ \left(\tfrac{1}{2}(\lambda+s-\mu)T\right)^{-\frac{1}{2}}\ \frac{1}{n!}\left(\frac{\lambda\mu T}{\lambda+s-\mu}\right)^n\ I_{n+\frac{1}{2}}\left(\tfrac{1}{2}(\lambda+s-\mu)T\right).$$

Infinite sums of modified spherical Bessel functions are governed by the generating function (10.2.31),

$$\sum_{n=0}^{\infty} \sqrt{\frac{\pi}{2}}\ z^{-\frac{1}{2}}\ \frac{t^n}{n!}\ I_{n-\frac{1}{2}}(z) = \frac{\cosh\sqrt{z^2 + 2zt}}{z}.$$

To obtain a new identity that suits our purposes we take the partial derivative of both sides with respect to t and reindex to obtain

$$\sum_{n=0}^{\infty} \sqrt{\frac{\pi}{2}}\, z^{-\frac{1}{2}}\, \frac{t^n}{n!}\, I_{n+\frac{1}{2}}(z) \;=\; \frac{\sinh \sqrt{z^2 + 2zt}}{\sqrt{z^2 + 2zt}},$$

where $\cosh z = (\exp(z) + \exp(-z))/2$ and $\sinh z = (\exp(z) - \exp(-z))/2$. With this new identity we can rewrite our infinite series as

$$\sum_{n=2}^{\infty} \mathcal{L}^{-1}\left\{ \frac{\lambda^n \mu^{n-1}}{(\lambda+s+r)^n (\mu+r)^n} \right\} =$$
$$\lambda T\; e^{-\frac{1}{2}(\lambda+s+\mu)T} \left[ \frac{\sinh \sqrt{\left(\frac{1}{2}(\lambda+s-\mu)T\right)^2 + \lambda\mu T^2}}{\sqrt{\left(\frac{1}{2}(\lambda+s-\mu)T\right)^2 + \lambda\mu T^2}} - \frac{\sinh\left(\frac{1}{2}(\lambda+s-\mu)T\right)}{\left(\frac{1}{2}(\lambda+s-\mu)T\right)} \right].$$

The second term in the square brackets arises because we were missing the zeroth term of the generating function and thus subtracted $\sqrt{\pi/2z}\; I_{1/2}(z)$, which by (10.2.13) is just $\sinh(z) / z$. Adding this expression with the term for n = 1 and simplifying, we obtain the Laplace transform of $f_2(t, T)$,

$$(14)\quad \mathcal{L}\{ f_2(t,T) \} = \lambda T\, \frac{\sinh y}{y\, e^h},$$

where $y = \sqrt{\lambda^2 + 2\lambda\mu + \mu^2 + 2\lambda s - 2\mu s + s^2}\; T / 2$ and $h = (\lambda + \mu + s)\, T / 2$. This completes the subproblem we have focused on.

Recall that the second part of the problem is to compute the Laplace transform of $f_1(t, T)$,

$$\mathcal{L}\{ f_1(t,T) \} = \sum_{n=1}^{\infty} \mathcal{L}^{-1}\left\{ \frac{\lambda^n \mu^n}{(\lambda+s+r)^{n+1} (\mu+r)^n} \right\}.$$

This turns out to be somewhat more cumbersome but is similar to the development above. Here the first term is

$$\mathcal{L}^{-1}\{\frac{\lambda\mu}{(\lambda+s+r)^2(\mu+r)}\} = \lambda\mu\ e^{-(\lambda+s)T}\int_0^T \frac{1-e^{-(\mu-\lambda)\tau}}{\mu-\lambda}\,d\tau$$

$$=\lambda\mu\ e^{-(\lambda+s)T}\left[\frac{-1}{(\lambda+s-\mu)^2}+\frac{e^{(\lambda+s-\mu)T}-(\lambda+s-\mu)T}{(\lambda+s-\mu)^2}\right].$$

For $n \geq 2$, we proceed in three steps, first finding

$$\mathcal{L}^{-1}\{\frac{\lambda^n\mu^n}{r\ r^n\ (\mu-\lambda-s+r)^n}\} =$$

$$\int_{\tau=0}^{\tau=T}\sqrt{\pi}\ \frac{(\lambda\mu)^n}{(n-1)!}\left(\frac{\tau}{\lambda+s-\mu}\right)^{n-\frac{1}{2}}\ e^{\frac{1}{2}(\lambda+s-\mu)\tau}\ I_{n-\frac{1}{2}}\left(\tfrac{1}{2}(\lambda+s-\mu)\tau\right)\,d\tau.$$

To evaluate this integral we first rewrite it as

$$\sqrt{\pi}\ \frac{(\lambda\mu)^n}{(n-1)!}\left(\frac{2}{\lambda+s-\mu}\right)^{2n}\left(\frac{1}{2}\right)^{n-\frac{1}{2}}\int_{u=0}^{u=w} u^{n-\frac{1}{2}}\ e^u\ I_{n-\frac{1}{2}}(u)\ du$$

where $u = \frac{1}{2}(\lambda+s-\mu)\tau$ and $w = \frac{1}{2}(\lambda+s-\mu)T$. Note that $du = \frac{1}{2}(\lambda+s-\mu)d\tau$. Finally, applying (11.3.12), and simplifying we obtain

$$\frac{\sqrt{\pi}}{2}\ \frac{(\lambda\mu)^n}{n!}\left(\frac{1}{\lambda+s-\mu}\right)^{n-\frac{1}{2}}\ T^{n+\frac{1}{2}}\ e^{\frac{1}{2}(\lambda+s-\mu)T}\left[I_{n-\frac{1}{2}}\left(\tfrac{1}{2}(\lambda+s-\mu)T\right)-I_{n+\frac{1}{2}}\left(\tfrac{1}{2}(\lambda+s-\mu)T\right)\right].$$

The second step is to use (29.2.12) to substitute r+λ+s for r to obtain

$$\mathcal{L}^{-1}\left\{ \frac{\lambda^n \mu^n}{(\lambda+s+r)^{n+1}\,(\mu+r)^n} \right\} =$$

$$\frac{\sqrt{\pi}}{2}\,\frac{(\lambda\mu)^n}{n!}\left(\frac{1}{\lambda+s-\mu}\right)^{n-\frac{1}{2}} T^{n+\frac{1}{2}}\; e^{-\frac{1}{2}(\lambda+s+\mu)T}\left[ I_{n-\frac{1}{2}}\left(\tfrac{1}{2}(\lambda+s-\mu)T\right) - I_{n+\frac{1}{2}}\left(\tfrac{1}{2}(\lambda+s-\mu)T\right)\right].$$

The third step, as before, is to evaluate the sum from n=2 to infinity. This produces two infinite series, the first given by

$$w\; e^{-\frac{1}{2}(\lambda+s+\mu)T} \sum_{n=2}^{\infty} I_{n-\frac{1}{2}}(w)\,\sqrt{\frac{\pi}{2w}}\,\frac{v^n}{n!}$$

where $w = ((\lambda+s-\mu)T)/2$ and $v = \lambda\,\mu\,T\,/\,(\lambda+s-\mu)$. Applying the generating function (10.2.31) we obtain

$$e^{-\frac{1}{2}(\lambda+s+\mu)T}\left(\mathbf{cosh}\left(\sqrt{w^2+2wv}\right) - \mathbf{cosh}(w) - v\;\mathbf{sinh}(w)\right)$$

where we note that the generating function sum starts at zero so we have, using (10.2.13) and (10.2.14), subtracted off $\sqrt{\pi/2w}\; I_{-1/2}(w) = \mathbf{cosh}(w)/w$ and $v\,\sqrt{\pi/2w}\; I_{1/2}(w) = v\;\mathbf{sinh}(w)/w$. The second series can be written as

$$-w\; e^{-\frac{1}{2}(\lambda+s+\mu)T} \sum_{n=2}^{\infty} I_{n+\frac{1}{2}}(w)\,\sqrt{\frac{\pi}{2w}}\,\frac{v^n}{n!}.$$

Using our modified generating function and subtracting off the missing terms using (10.2.13) and (10.2.14) as before,

$$-w\; e^{-\frac{1}{2}(\lambda+s+\mu)T}\left(\frac{\mathbf{sinh}\,\sqrt{w^2+2wv}}{\sqrt{w^2+2wv}} - \frac{\mathbf{sinh}\;w}{w} - v\left(\frac{\mathbf{cosh}\;w}{w} - \frac{\mathbf{sinh}\;w}{w^2}\right)\right).$$

Combining these results we can, after a considerable amount of algebra, obtain a simpler expression for the Laplace transform of $f_1(t, T)$,

$$(15)\quad \mathcal{L}\{ f_1(t,T) \} = \frac{\cosh y}{e^h} - w \frac{\sinh y}{y\ e^h} - e^{-(\lambda+s)T}.$$

This completes the computation of the Laplace transform of $f_1(t, T)$ and $f_2(t, T)$. The only remaining step is to compute the Laplace transform of the impulse term $f_3(t, T)$, which is simply

$$(16)\quad \mathcal{L}\{ f_3(t,T) \} = e^{-(\lambda+s)T}.$$

Combining the three transforms (14), (15), and (16), we finally obtain

$$(17)\quad f^*(s,\ T \mid 1) = \frac{(\lambda+\mu-s)T}{2} \frac{\sinh y}{y\ e^h} + \frac{\cosh y}{e^h},$$

or, in terms of exponentials,

$$f^*(s,\ T \mid 1) = \frac{(\lambda+\mu-s)T}{2} \frac{e^{y-h} - e^{-(y+h)}}{2y} + \frac{e^{y-h} + e^{-(y+h)}}{2},$$

where $y = \sqrt{\lambda^2 + 2\lambda\mu + \mu^2 + 2\lambda s - 2\mu s + s^2}\ T / 2$ and $h = (\lambda + \mu + s)\ T / 2$. Note that $y = h$ when $s = 0$, so that the above expression is easily seen to equal 1 at $s = 0$. This is an important check because the density $f(t; T \mid 1)$ must integrate to 1.

Further, we observe that by definition, the Laplace transform of $f(t; T \mid 10)\ P_{10}(T)$ is the Laplace transform of $f_2(t, T)$ and the Laplace transform of $f(t; T \mid 11)\ P_{11}(T)$ is the Laplace

transform of $f_1(t, T) + f_3(t, T)$, all of which have been derived above. As we would expect, the Laplace transform of $f_2(t, T)$ equals $P_{10}(T)$ and the Laplace transform of $f_1(t, T) + f_3(t, T)$ equals $P_{11}(T)$ at $s = 0$.

**Mean uptime**

The mean is obtained by taking the derivative of the transform with respect to s and then setting s equal to zero and negating the result. The process is straightforward although cumbersome, and after much simplification yields

$$(18)\quad E[f \mid 1] = \frac{\mu}{\lambda+\mu}T + \frac{\lambda}{(\lambda+\mu)^2}\left(1 - e^{-(\lambda+\mu)T}\right).$$

This expression agrees with the result of Barlow and Proschan (1965). Also, this can be easily derived from the general Laplace transform result of Takács (1957a). As a simple check we note that for small T, $E[f \mid 1] = T + O(T^2)$; at $T = 0$ the above expression is zero; and as T approaches infinity, $E[f \mid 1] / T$ approaches $\mu/(\lambda+\mu)$, as we would expect.

Up to this point we have supposed that the machine is working at time 0. To derive expressions where the machine is *failed* at time zero, we will be working with $h(t; T)$, the PDF for *downtime* over an interval of length T. To find the mean downtime over an interval of length T when the machine is failed at time 0, we simply reverse the roles of $\lambda$ and $\mu$ to obtain

$$E[h \mid 0] = \frac{\lambda}{\lambda+\mu}T + \frac{\mu}{(\lambda+\mu)^2}\left(1 - e^{-(\lambda+\mu)T}\right).$$

Since $E[f \mid 0] + E[h \mid 0] = T$, we simply subtract the above expression from T to obtain

$$(19)\quad E[f \mid 0] = \frac{\mu}{\lambda+\mu}T - \frac{\mu}{(\lambda+\mu)^2}\left(1-e^{-(\lambda+\mu)T}\right).$$

Since we have found the transform of f(t; T | 10) and f(t; T | 11), we can also derive E[f | 10] and E[f | 11] by taking the derivative of the transform with respect to s and then setting s equal to zero and negating the result. After simplification, we obtain

$$(20)\quad E[f \mid 10] = \left(\frac{\lambda}{\lambda+\mu}\right)^2 \frac{T}{1-e^{(\lambda+\mu)T}} + \left(\frac{\mu}{\lambda+\mu}\right)^2 \frac{T}{1-e^{-(\lambda+\mu)T}} + \frac{\lambda\mu T+\lambda-\mu}{(\lambda+\mu)^2},$$

$$(21)\quad E[f \mid 11] = \frac{2\lambda\mu}{(\lambda+\mu)^2}\left(\frac{1}{\mu+\lambda e^{-(\lambda+\mu)T}} - \frac{1}{\lambda+\mu e^{(\lambda+\mu)T}}\right) + \frac{T}{(\lambda+\mu)^2}\left(\frac{\lambda^3}{\lambda+\mu e^{(\lambda+\mu)T}} + \frac{\mu^3}{\mu+\lambda e^{-(\lambda+\mu)T}}\right) + \frac{\lambda\mu T}{(\lambda+\mu)^2},$$

and not surprisingly, we can also show that

$$(22)\quad E[f \mid 01] = E[f \mid 10],$$

$$(23)\quad E[f \mid 00] = T - \frac{2\lambda\mu}{(\lambda+\mu)^2}\left(\frac{1}{\lambda+\mu e^{-(\lambda+\mu)T}} - \frac{1}{\mu+\lambda e^{(\lambda+\mu)T}}\right) - \frac{T}{(\lambda+\mu)^2}\left(\frac{\mu^3}{\mu+\lambda e^{(\lambda+\mu)T}} + \frac{\lambda^3}{\lambda+\mu e^{-(\lambda+\mu)T}}\right) - \frac{\lambda\mu T}{(\lambda+\mu)^2}.$$

These expressions divided by T approach $\mu/(\lambda+\mu)$ as T approaches infinity, and approach zero as T approaches zero, as we would expect.

**Laplace transform with machine initially failed or in steady state**

Using the same observations as above we can easily find the Laplace transform of f(t; T | 0). We reverse the roles of $\lambda$ and $\mu$, add T to t (which corresponds to multiplying

the transform by **exp**(s T)) and then negate t (which corresponds to negating the transform variable s). The result is

$$(24)\quad f^*(s,\ T \mid 0) = h\ \frac{e^{y-h} - e^{-(y+h)}}{2y} + \frac{e^{y-h} + e^{-(y+h)}}{2}.$$

As a final check on our work, let us examine the case where the machine state at time 0 is unknown, i.e., the machine starts out in steady-state. Then the probability that the machine starts out working is $\mu/(\lambda+\mu)$ and failed is $\lambda/(\lambda+\mu)$. Thus, the expected time the machine is working over an interval of length T is

$$\frac{\mu}{\lambda+\mu}E[f \mid 1] + \frac{\lambda}{\lambda+\mu}E[f \mid 0]$$

Substituting the above expressions, we obtain

$$\frac{\mu}{\lambda+\mu}\left[\frac{\mu}{\lambda+\mu}T + \frac{\lambda}{(\lambda+\mu)^2}\left(1-e^{-(\lambda+\mu)T}\right)\right] + \frac{\lambda}{\lambda+\mu}\left[\frac{\mu}{\lambda+\mu}T - \frac{\mu}{(\lambda+\mu)^2}\left(1-e^{-(\lambda+\mu)T}\right)\right]$$

which is, after simplification,

$$E[f] = T\,\mu/(\lambda+\mu),$$

as we would expect from the theory of alternating renewal processes. In fact, it is well known that this result holds for any repair and failure distributions (Ross, 1983).

**Variance of uptime**

To derive an expression for the variance of uptime over an interval of length T when the machine is working at time 0, we first find the second moment by taking the second derivative of the transform (17) with respect to s and then setting s = 0, which, after simplification, gives

$$\frac{2\lambda^2 - 4\lambda\mu}{(\lambda+\mu)^4}\left(1 - e^{-(\lambda+\mu)T}\right) + \frac{4\lambda\mu}{(\lambda+\mu)^3}T - \frac{2\lambda^2}{(\lambda+\mu)^3}Te^{-(\lambda+\mu)T} + \frac{\mu^2}{(\lambda+\mu)^2}T^2.$$

To find the variance we square our expression for the mean and subtract it from the above expression to obtain (after simplification)

$$(25)\quad \mathrm{Var}[f \mid 1] = \frac{\lambda^2}{(\lambda+\mu)^4}\left(1 - e^{-2(\lambda+\mu)T}\right) - \frac{4\lambda\mu}{(\lambda+\mu)^4}\left(1 - e^{-(\lambda+\mu)T}\right) + \frac{2\lambda\mu}{(\lambda+\mu)^3}T\left(1 + e^{-(\lambda+\mu)T}\right) - \frac{2\lambda^2}{(\lambda+\mu)^3}Te^{-(\lambda+\mu)T}.$$

We can find Var[f | 0] in a similar manner. The result is

$$(26)\quad \mathrm{Var}[f \mid 0] = \frac{\mu^2}{(\lambda+\mu)^4}\left(1 - e^{-2(\lambda+\mu)T}\right) - \frac{4\lambda\mu}{(\lambda+\mu)^4}\left(1 - e^{-(\lambda+\mu)T}\right) + \frac{2\lambda\mu}{(\lambda+\mu)^3}T + \frac{2\mu(\lambda-\mu)}{(\lambda+\mu)^3}Te^{-(\lambda+\mu)T}.$$

Note that as T approaches infinity, both Var[f | 1] / T and Var[f | 0] / T approach $2\lambda\mu/(\lambda+\mu)^3$. This asymptotic result agrees with the general result of Takács (1957a); see also Gnedenko et al. (1969). Further, these authors both show that the asymptotic distribution is Normal.

Lastly, with a substantial amount of algebra, it can be seen that

$$(27)\quad \mathrm{Var}[f\,|\,10] = \frac{1}{(\lambda+\mu)^2} - \frac{8\lambda\mu}{(\lambda+\mu)^4} + \frac{2\lambda\mu T}{(\lambda+\mu)^3\left(1-e^{-(\lambda+\mu)T}\right)^2} - \frac{2\lambda\mu T}{(\lambda+\mu)^3\left(1-e^{(\lambda+\mu)T}\right)^2} - T^2\left(\frac{\lambda-\mu}{\lambda+\mu}\right)^2 \frac{e^{(\lambda+\mu)T}}{\left(1-e^{(\lambda+\mu)T}\right)^2},$$

$$(28)\quad \mathrm{Var}[f\,|\,11] = \frac{6\lambda\mu\left(\mu^2 e^{(\lambda+\mu)T} - \lambda^2\right)\left(1-e^{(\lambda+\mu)T}\right)}{(\lambda+\mu)^4\left(\lambda+\mu e^{(\lambda+\mu)T}\right)^2} + \frac{2\lambda^2\mu^2\left(1-e^{(\lambda+\mu)T}\right)^2}{(\lambda+\mu)^4\left(\lambda+\mu e^{(\lambda+\mu)T}\right)^2} - \frac{2\lambda^3\mu T\left(1+2e^{(\lambda+\mu)T}\right)}{(\lambda+\mu)^3\left(\lambda+\mu e^{(\lambda+\mu)T}\right)^2} + \frac{2\lambda\mu^3 T\left(2e^{(\lambda+\mu)T} + e^{2(\lambda+\mu)T}\right)}{(\lambda+\mu)^3\left(\lambda+\mu e^{(\lambda+\mu)T}\right)^2} + \frac{\lambda\mu T^2(\lambda-\mu)^2 e^{(\lambda+\mu)T}}{(\lambda+\mu)^2\left(\lambda+\mu e^{(\lambda+\mu)T}\right)^2},$$

$$(29)\quad \mathrm{Var}[f\,|\,01] = \mathrm{Var}[f\,|\,10],$$

$$(30)\quad \mathrm{Var}[f\,|\,00] = \frac{6\lambda\mu\left(\lambda^2 e^{(\lambda+\mu)T} - \mu^2\right)\left(1-e^{(\lambda+\mu)T}\right)}{(\lambda+\mu)^4\left(\lambda+\mu e^{(\lambda+\mu)T}\right)^2} + \frac{2\lambda^2\mu^2\left(1-e^{(\lambda+\mu)T}\right)^2}{(\lambda+\mu)^4\left(\lambda+\mu e^{(\lambda+\mu)T}\right)^2} + \frac{2\lambda^3\mu T\left(2e^{(\lambda+\mu)T} + e^{2(\lambda+\mu)T}\right)}{(\lambda+\mu)^3\left(\lambda+\mu e^{(\lambda+\mu)T}\right)^2} - \frac{2\lambda\mu^3 T\left(1+2e^{(\lambda+\mu)T}\right)}{(\lambda+\mu)^3\left(\lambda+\mu e^{(\lambda+\mu)T}\right)^2} + \frac{\lambda\mu T^2(\lambda-\mu)^2 e^{(\lambda+\mu)T}}{(\lambda+\mu)^2\left(\lambda+\mu e^{(\lambda+\mu)T}\right)^2}.$$

We note that each of these expressions divided by T approaches $2\lambda\mu / (\lambda+\mu)^3$ as T approaches infinity, and approaches zero as T approaches zero, as we would expect.

## 2.3 Distribution, transform, mean and variance of time to produce a fixed number of parts

The focus of the following development will be to characterize the time to produce a fixed lot of q parts at some processing speed p, exponentially distributed time between failures with MTBF $1/\lambda$, and exponentially distributed time to repair with MTTR $1/\mu$. Although this problem has been addressed previously by Brouwers (1986) and Kim and Alden (1992), we take a different approach that is simpler and provides insight into the more difficult problem with an arbitrary (general) repair distribution. The key observation is that the random variable of interest can be represented as a power series, from which the transform is easily found. This approach is not new; see, for example, Giffin (1975) for an excellent exposition, or Serfozo (1990) for a rigorous presentation of theoretical results.

### Laplace transform and density when machine initially working

We begin by observing that the time to produce a lot can be divided into two mutually-exclusive, collectively-exhaustive components: the processing time to produce parts, plus the time the machine spends in repair. Note that the first component is deterministic and the second is stochastic. Thus the time to produce a lot is given by

$$q/p + \mathbf{R}$$

where **R** is a random variable representing the time spent in repair. We will thus focus our attention on **R**.

Let us first define b as the arrival rate of failures per batch, given by

$$b = \lambda q / p.$$

Using this notation, two parameters characterize the distribution of **R**: b and $\mu$. The key observation of our derivation is that if the machine is working at time 0, we can model the failure process as a Compound Poisson process of the form

$$\mathbf{R}(\bullet; b \mid 1) = \sum_{i=1}^{N(b)} \mathbf{X}_i$$

where **N**(b) is the number of arrivals in the Poisson failure process with rate b and each $\mathbf{X}_i$ is exponentially distributed with rate $\mu$. Using standard results (Ross, 1983) for Compound Poisson processes,

$$E[\mathbf{R} \mid 1] = b/\mu,$$
$$\mathrm{Var}[\mathbf{R} \mid 1] = 2b/\mu^2.$$

In fact, we can easily find all the moments since the Laplace transform is easily found. We note that R is a *mixture* and can be represented by the power series

$$\mathbf{R}(\bullet;\ b \mid 1) \ = \ \sum_{i=0}^{\infty} p_i(b)\ \mathbf{X}^{i*}$$

where $\{p_i(b)\}$ is the Poisson distribution with rate b and $\mathbf{X}^{i*}$ represents the i-fold convolution of **X**. With this observation, we may write

$$r^*(s;\ b \mid 1) \ = \ \sum_{i=0}^{\infty} p_i \left(x^*(s)\right)^i$$

where $r^*$ and $x^*$ denote the Laplace transforms of **R** and **X**. Letting $\phi$ denote the characteristic function of a random variable,

$$r^*(s;\ b \mid 1) \ = \ \phi_P\left(x^*(s)\right)$$

(by definition of a characteristic function). Since $\phi_P(s) = \mathbf{exp}(-b + bs)$, we conclude that

$$(31)\quad r^*(s;\ b \mid 1) \ = \ \mathbf{exp}\left(-b\left(1 - x^*(s)\right)\right),$$

which agrees with the Laplace transform for the Compound Poisson processes (Ross, 1983). In the case of exponentially distributed repairs, $x^*(s) = \mu/(s+\mu)$, and thus,

$$(32)\quad r^*(s;\ b \mid 1) \ = \ \mathbf{exp}\left(-b + b\frac{\mu}{s+\mu}\right).$$

Calculating the first and second derivatives of $r^*(s; b \mid 1)$ at $s = 0$ for the exponential case validates the two moments obtained above; note that from (31) we can easily obtain the moments for any repair distribution.

From this transform we can obtain r, the density of the time spent in repair, for the exponential case. We will need the fact that the Laplace transform of $I_1(2\sqrt{x})/\sqrt{x}$ is $\mathbf{exp}(1/s) - 1$ (Feller, 1971), which is easily verified by Maclaurin expansion of the transform (Doetsch, 1961). Using this fact and the basic rules of Laplace transforms, we obtain

$$(33)\quad r(t;b \mid 1) \ = \ u_0(t)\mathbf{exp}(-b) \ + \ \mu \ b \ \mathbf{exp}(-\mu t - b) \ I_1\left(2\sqrt{\mu bt}\right) \left(\mu bt\right)^{-\frac{1}{2}},\ t \geq 0,$$

where $u_0$ is the unit impulse function and $I_1$ is the modified Bessel function of order 1. This density has also been obtained by direct probabilistic argument by Feller (1971), Brouwers (1986), and Kim and Alden (1992).

**Laplace transform and density when machine initially failed**

We would now like to find r(t; b | 0), the density of time spent in repair given that the machine is currently failed. This is given by the convolution of r(t; b | 1) with the density of time to repair (i.e., the exponential density with parameter $\mu$). The convolution integral is difficult to evaluate, but the Laplace transform is simply the product of the two transforms, and is

$$(34)\quad r^*(s;\ b\,|\,0) \;=\; \frac{\mu}{s+\mu}\ \exp\!\left(-b+b\frac{\mu}{s+\mu}\right).$$

From this transform we can easily find the first two moments of r(t; b | 0). The result is, not surprisingly,

$$E[\mathbf{R} \mid 0] = b/\mu + 1/\mu,$$
$$\mathrm{Var}[\mathbf{R} \mid 0] = 2b/\mu^2 + 1/\mu^2.$$

To find r(t; b | 0), we need only to invert the above transform. The symbol $\Leftrightarrow$ will be used to represent that the expression on the left is the Laplace transform of the expression on the right. We begin with the following transform identity from standard tables of Laplace transforms (e.g., (29.3.81) in Abramowitz and Stegun),

$$\frac{1}{s^{\nu+1}}\ \exp(1/s) \;\Leftrightarrow\; t^{\nu/2}\ I_\nu\!\left(2\sqrt{t}\right),\quad \nu > -1,$$

where $I_v(z)$ is the modified Bessel function of order v,

$$I_v(z) = \left(\frac{z}{2}\right)^v \sum_{k=0}^{\infty} \frac{\left(\frac{z}{2}\right)^{2k}}{k!\,(k+v)!}$$

(see (9.6.10) in Abramowitz and Stegun). We now proceed with three simple steps to transform the above identity into $r^*(s; b \mid 0)$. First, taking the identity at $v = 0$ and scaling s by $1/\mu b$ gives

$$\frac{1}{\mu b}\left(\frac{\mu b}{s}\right) \exp(\mu b/s) \;\Leftrightarrow\; I_0\left(2\sqrt{\mu b t}\right).$$

Next, we replace s by $s + \mu$, which is equivalent to multiplying the inverse transform by $\exp(-\mu t)$, and then scale both sides by the constant $\mu \exp(-b)$. We obtain

$$\left(\frac{1}{s+\mu}\right) \exp\left(b\frac{\mu}{s+\mu}\right) \;\Leftrightarrow\; \exp(-\mu t)\; I_0\left(2\sqrt{\mu b t}\right),$$

and

$$\left(\frac{\mu}{s+\mu}\right) \exp\left(-b + b\frac{\mu}{s+\mu}\right) \;\Leftrightarrow\; \mu\, \exp(-\mu t - b)\; I_0\left(2\sqrt{\mu b t}\right).$$

Therefore,

(35) $r(t; b \mid 0) = \mu \exp(-\mu t - b)\; I_0\left(2\sqrt{\mu b t}\right),\; t \geq 0.$

We should not be surprised that the impulse term in r(t; b | 1) is not present in r(t; b | 0): since the machine is currently failed, the time spent in repair time is almost surely non-zero.

Since the stochastic process of this section is related to a compound Poisson process, the Laplace transforms (32) and (34) are of a special type, and as a result can be numerically inverted by the rather elegant method of Van Landingham and Shariq (1974). These authors present an efficient method that is specialized for inverting transforms of this type.

We also note that from the above development, we obtain an equivalence between the convolution integral and r(t; b | 0). Writing this equivalence and simplifying, one obtains a surprisingly simple identity,

$$\int_0^z a \, \frac{I_1\left(2\sqrt{at}\right)}{\sqrt{at}} \, dt \;=\; I_0\left(2\sqrt{az}\right) - 1.$$

## 2.4 Cumulative distribution of time to produce a fixed number of parts

To find the CDF R(t; b | 1) one could attempt several different approaches. The most obvious is to integrate the density obtained in the previous section. This is possible for the functional form involved; this method is illustrated on a similar function R(t; b | 11) in the next section. Another possible method is direct probabilistic argument, and this has been successfully accomplished by Kim (1994, unpublished). We take a different approach, utilizing the Laplace transform obtained in the previous section.

If f(t) is any non-negative function of t and f*(s) is its Laplace transform, then f*(s)/s is the transform of the integral of f(t) from zero to t, namely, the CDF. We therefore seek the inverse Laplace transform of r*(s; b | 1)/s, i.e.,

$$\mathcal{L}^{-1}\left\{\frac{1}{s}\,\exp\left(-b+b\frac{\mu}{s+\mu}\right)\right\}$$

As in the previous section, we begin with the transform identity

$$\frac{1}{s^{v+1}}\,\exp(1/s) \;\Leftrightarrow\; t^{v/2}\; I_v\left(2\sqrt{t}\right),\quad v > -1$$

from standard tables of Laplace transforms (e.g., (29.3.81) in Abramowitz and Stegun). We now proceed with a series of simple steps to transform the above identity into R*(s; b | 1)/s. First, scaling s by $1/\mu b$ gives

$$\frac{1}{\mu b}\left(\frac{\mu b}{s}\right)^{v+1}\exp(\mu b/s) \;\Leftrightarrow\; (\mu bt)^{v/2}\; I_v\left(2\sqrt{\mu bt}\right),\quad v > -1.$$

Multiplying both sides by the constant $b^{-v}$ yields

$$\frac{1}{\mu}\left(\frac{\mu}{s}\right)^{v+1}\exp(\mu b/s) \Leftrightarrow \left(\frac{\mu t}{b}\right)^{v/2} I_v\left(2\sqrt{\mu bt}\right), \quad v > -1.$$

Next, we replace s by $s + \mu$, which is equivalent to multiplying the inverse transform by $\exp(-\mu t)$, and then scale both sides by the constant $\exp(-b)$. We obtain

$$\frac{1}{\mu}\left(\frac{\mu}{s+\mu}\right)^{v+1}\exp\left(b\frac{\mu}{s+\mu}\right) \Leftrightarrow \exp(-\mu t)\left(\frac{\mu t}{b}\right)^{v/2} I_v\left(2\sqrt{\mu bt}\right), \quad v > -1,$$

and

$$\frac{1}{\mu}\left(\frac{\mu}{s+\mu}\right)^{v+1}\exp\left(-b+b\frac{\mu}{s+\mu}\right) \Leftrightarrow \exp(-\mu t - b)\left(\frac{\mu t}{b}\right)^{v/2} I_v\left(2\sqrt{\mu bt}\right), \quad v > -1.$$

We are now almost done, as the left-hand side of the above identity is very similar to our desired $R^*(s; b \mid 1)/s$. Although no simple transformation of the above expression will give us the form that we desire, by the additivity of Laplace transforms we can create the identity we seek, by summing over v from zero to infinity. That is,

$$\frac{1}{\mu}\exp\left(-b+b\frac{\mu}{s+\mu}\right)\sum_{v=0}^{\infty}\left(\frac{\mu}{s+\mu}\right)^{v+1} \Leftrightarrow \sum_{v=0}^{\infty}\exp(-\mu t - b)\left(\frac{\mu t}{b}\right)^{v/2} I_v\left(2\sqrt{\mu bt}\right),$$

which, after simplification, gives

$$\frac{1}{\mu}\exp\left(-b+b\frac{\mu}{s+\mu}\right)\left(\frac{\mu}{s}\right) \Leftrightarrow \exp(-\mu t - b)\sum_{v=0}^{\infty}\left(\frac{\mu t}{b}\right)^{v/2} I_v\left(2\sqrt{\mu bt}\right).$$

Canceling the $\mu$ and $1/\mu$ on the left-hand side, we obtain the desired result, namely, that

$$(36)\quad R(t;b\mid 1) \;=\; \exp(-\mu t - b)\sum_{v=0}^{\infty}\left(\frac{\mu t}{b}\right)^{\frac{v}{2}} I_v\left(2\sqrt{\mu b t}\right),\; t \geq 0.$$

This expression agrees with Kim (1994, unpublished). It is at first surprising that there is not an impulse term at zero, such as the one in our expression for r(t; b | 1). Note, however, that at t = 0 the first term of the infinite series is one and all the others are zero, so that R(0; b | 1) is indeed exp(–b). Although this infinite series can not be simplified further, we can evaluate a finite number of terms as an approximation. Press et al. (1989) present an algorithm to compute $I_v(z)$ using downward recurrence in v and the polynomial approximation for $I_0(z)$ given by Abramowitz and Stegun (1964). More sophisticated algorithms exist for computing a sequence of modified Bessel functions, such as the algorithm of Cody (1983), which provides guaranteed error bounds. Several codes are in the public domain and are available via *netlib* (Dongarra and Grosse, 1987); most commercial numerical libraries also provide such routines.

By a simple modification of the above derivation, we can also show that

$$(37)\quad R(t;b\mid 0) \;=\; \exp(-\mu t - b)\sum_{v=1}^{\infty}\left(\frac{\mu t}{b}\right)^{\frac{v}{2}} I_v\left(2\sqrt{\mu b t}\right),\; t \geq 0.$$

## 2.5 Cumulative distribution of parts produced over a fixed period of time

The task of this section will be to obtain the cumulative distribution of parts produced by a machine at processing speed p over the time period [0, T) when interarrivals of failures and repairs are exponentially distributed with means $\lambda$ and $\mu$, respectively.

We now show that this distribution follows immediately from the results of the previous section. By simply noting that the Pr { parts produced in [0, T) $\leq$ q } is equivalent to Pr { time to produce q parts $\geq$ T }, which is equal to Pr { downtime incurred while producing q parts $\geq$ T – q/p }, we can write the following *equivalence property*

(38) $F(q/p; T \mid 1) = 1 - R(T - q/p; \lambda\, q/p \mid 1).$

Therefore, carefully accounting for impulses and endpoints, the CDF F(t; T | 1) is given by

$$(39)\quad F(t;T\mid 1) = \begin{cases} 0 & t = 0 \\ 1 - e^{-\mu(T-t)-\lambda t} \sum_{v=0}^{\infty} \left(\frac{\mu(T-t)}{\lambda t}\right)^{\frac{v}{2}} I_v\left(2\sqrt{\mu\ \lambda\ t\ (T-t)}\right) & 0 < t < T \\ 1 & t \geq T. \end{cases}$$

Barlow and Hunter (1961) give an alternative formula for F(t; T | 1), 0 < t < T,

$$1 - e^{-\lambda t}\left[1 + \sqrt{\mu\ \lambda\ t}\ \int_0^{T-t} e^{-\mu x}\ x^{-1/2}\ I_1\left(2\sqrt{\mu\ \lambda\ t\ x}\right)\ dx\right]$$

due to Takács (1957). This integral is simply $1 - f(t; T \mid 11)\, P_{11}(T)$ with (T-t) replaced by x and then integrated from zero to T–t. This has an intuitive physical interpretation; see the end of this subsection. Unfortunately this integral for has no known closed-form solution, and is therefore not much more useful than the integral of the density $f(t; T \mid 1)$.

We can also conclude with analogous logic that

$$(40)\quad F(t;T \mid 0) = \begin{cases} 0 & t = 0 \\ 1 - e^{-\mu(T-t)-\lambda t} \sum_{v=1}^{\infty} \left(\frac{\mu(T-t)}{\lambda t}\right)^{\frac{v}{2}} I_v\left(2\sqrt{\mu\ \lambda\ t\ (T-t)}\right) & 0 < t < T \\ 1 & t \geq T. \end{cases}$$

We have been able to verify this expression and the expression for $F(t; T \mid 1)$ by brute-force integration of their respective densities $f(t; T \mid 0)$ and $f(t; T \mid 1)$. The basic approach is to recognize that

$$1 - F(t; T \mid 1) = \sum_{n=1}^{\infty} \Pr\{\text{ n failures in t time units }\} \Pr\{\text{ } n^{th} \text{ repair occurs at time} \leq T\text{–t }\},$$

using an argument similar to the one used to derive $f(t; T \mid 1)$. $\Pr\{\text{ } n^{th}$ repair occurs at time $\leq$ T-t } can be written as an infinite series using (6.5.1) and (6.5.29) of Abramowitz and Stegun (1964). Once this is done, one manipulates the two infinite series to produce a single infinite series of modified Bessel functions, and the above result follows immediately. An example of this technique can be seen at the end of this section.

**Distribution function with known starting and terminal machine states**

We now turn our attention to finding $F(t; T \mid 11)$, the cumulative distribution of parts produced by a machine at processing speed p over the time period [0, T) given that the machine is working at time 0 and at time T. This distribution will be important to our dynamic programming models in Chapter 3. To find this distribution, we will, as before, first derive an expression in terms of the distribution function R, and then exploit an equivalence between the distribution functions R and F.

In particular, we will now derive the probability that the downtime while producing a batch of size q is at most t, given that the machine starts working and is also working at time $t + q/p$, where p is the production speed of the machine. In our notation, this probability is $R(t; \lambda\, q/p \mid \alpha(0) = 1, \alpha(t+q/p) = 1)$; we will abbreviate this as $R(t; \lambda\, q/p \mid 11)$. Our derivation is a probabilistic argument based on Bayes' theorem. We begin by writing

$$1 - R(t; \lambda\, q/p \mid 11) = 1 - \Pr\{\text{downtime to produce q parts} \le t \mid 11\}.$$

The key step is to rewrite this as

$$R(t; \lambda\, q/p \mid 11) = \frac{\Pr\{\text{downtime to produce q parts} \le t \text{ and } \alpha(t+q/p) = 1 \mid 1\}}{\Pr\{\alpha(t+q/p) = 1 \mid 1\}}$$

$$= \frac{\int_{y=0}^{t} \text{dens}\{\text{downtime to produce q parts} = y\}\, \Pr\{\alpha(t+q/p-(q/p+y)) = 1 \mid 1\}\, dy}{\Pr\{\alpha(t+q/p) = 1 \mid 1\}}.$$

The reasoning behind the numerator of the last expression is as follows. First, the event { downtime to produce q parts ≤ t } has been rewritten as { downtime to produce q parts = y } where y is integrated from 0 to t. If the downtime to produce q parts is y, the

$q^{th}$ part is completed at time $q/p + y$. The machine must be working at the instant $q/p + y$, so in order for the machine to be working at time $t + q/p$, it must be back in the working state after an interval of length $t + q/p - (q/p + y)$. Therefore,

$$R(t; \lambda\, q/p \mid 11) = \frac{\int_{y=0}^{t} r(y;\ q/p \mid 1)\, P_{11}(t-y)\ dy}{P_{11}(t+q/p)}$$

where $P_{11}(T)$ is the probability that the machine is still working T time units later. $P_{11}(T)$ is simple to derive and appears in many contexts; we first used it in Section 2.1. It is given by

$$P_{11}(T) = \frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu} e^{-(\lambda+\mu)T}.$$

Substituting $P_{11}(T)$ gives

$$R(t; \lambda\, q/p \mid 11) = \frac{\int_{y=0}^{t} r(y;\ q/p \mid 1) \left[ \frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu} e^{-(\lambda+\mu)(t-y)} \right] dy}{\frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu}\, e^{-(\lambda+\mu)(t+q/p)}}$$

$$= \frac{\mu\ R(t;\ q/p \mid 1) + \lambda\ e^{-(\lambda+\mu)t} \int_{y=0}^{t} r(y;\ q/p \mid 1)\, e^{(\lambda+\mu)y}\ dy}{\mu + \lambda e^{-(\lambda+\mu)(t+q/p)}}.$$

The last step is to rewrite the integral in the numerator. Substituting the value for the density r, replacing its Bessel function $I_0$ with the usual infinite series representation, setting $b = \lambda\, q\, /\, p$, and applying (6.5.2) of Abramowitz and Stegun (1964) to express the integral as an incomplete gamma function, we obtain

$$\int_{y=0}^{t} r(y;\ q/p \mid 1)\, e^{(\lambda+\mu)y}\ dy = e^{-b} + \sum_{j=0}^{\infty} \frac{(\mu bt)^{j+1} e^{-b}}{j!\ (j+1)!} (-\lambda t)^{-(j+1)}\ \gamma(j+1, -\lambda t)$$

where $\gamma(a,z) = \int_0^z e^{-t}\ t^{a-1}\ dt$. Rewriting the incomplete gamma function as an infinite series using (6.5.4) and (6.5.29) gives

$$\int_{y=0}^{t} r(y;\ q/p\ |\ 1)\ e^{(\lambda+\mu)y}\ dy\ =\ e^{-b} + \sum_{j=0}^{\infty} \frac{(\mu bt)^{j+1} e^{-b+\lambda t}}{(j+1)!} \sum_{v=0}^{\infty} \frac{(-\lambda t)^{v}}{(j+v+1)!}.$$

Rearranging terms and applying the infinite series representation for $I_v(z)$, we can write

$$\int_{y=0}^{t} r(y;\ q/p\ |\ 1)\ e^{(\lambda+\mu)y}\ dy\ =\ e^{-b} + \sum_{v=0}^{\infty} (-\lambda t)^{v}\ e^{-b+\lambda t} \left[ \frac{I_v\left(2\sqrt{\mu bt}\right)}{\left(\sqrt{\mu bt}\right)^{v}} - \frac{1}{v!} \right].$$

Lastly, recognizing the embedded Taylor series for **exp**$(-\lambda t)$ and simplifying, we obtain

$$\int_{y=0}^{t} r(y;\ q/p\ |\ 1)\ e^{(\lambda+\mu)y}\ dy\ =\ e^{-b+\lambda t} \sum_{v=0}^{\infty} (-1)^{v} \left( \frac{\lambda t}{\mu q/p} \right)^{v/2} I_v\left(2\sqrt{\mu bt}\right).$$

Thus, our final result is, after simplification,

$$(41)\ R(t;\ b\ |\ 11)\ =\ \frac{\mu\ R(t;\ b\ |\ 1) + \lambda\ e^{-b-\mu} \sum_{v=0}^{\infty} (-1)^{v} \left( \frac{\lambda t}{\mu q/p} \right)^{v/2} I_v\left(2\sqrt{\mu bt}\right)}{\mu + \lambda\ e^{-(\lambda+\mu)(t+q/p)}}.$$

To find $R(t;\ b\ |\ 10)$ we simply note using the law of total probability that

$$R(t;\ b\ |\ 1)\ =\ R(t;\ b\ |\ 11)\ P_{11}(t+q/p)\ +\ R(t;\ b\ |\ 10)\ P_{10}(t+q/p),$$

and therefore

$$(42)\quad R(t;\, b \mid 10) = \frac{R(t;\, b \mid 1) - R(t;\, b \mid 11)\, P_{11}(t+q/p)}{P_{10}(t+q/p)}$$

$$= \frac{R(t;\, b \mid 1) - e^{-b-\mu t} \sum_{v=0}^{\infty} (-1)^v \left(\frac{\lambda t}{\mu q/p}\right)^{v/2} I_v\left(2\sqrt{\mu b t}\right)}{1 - e^{-(\lambda+\mu)(t+q/p)}}.$$

Through a similar derivation one can also show that

$$(43)\quad R(t;\, b \mid 01) = \frac{R(t;\, b \mid 0) + e^{-b-\mu t} \sum_{v=0}^{\infty} (-1)^v \left(\frac{\lambda t}{\mu q/p}\right)^{\frac{v+1}{2}} I_{v+1}\left(2\sqrt{\mu b t}\right)}{1 - e^{-(\lambda+\mu)(t+q/p)}},$$

and, of course,

$$(44)\quad R(t;\, b \mid 00) = \frac{R(t;\, b \mid 0) - R(t;\, b \mid 01)\, P_{01}(t+q/p)}{P_{00}(t+q/p)}$$

$$= \frac{\lambda\, R(t;\, b \mid 0) - \mu\, e^{-b-\mu t} \sum_{v=0}^{\infty} (-1)^v \left(\frac{\lambda t}{\mu q/p}\right)^{\frac{v+1}{2}} I_{v+1}\left(2\sqrt{\mu b t}\right)}{\lambda + \mu\, e^{-(\lambda+\mu)(t+q/p)}}.$$

With a little effort, it can also be seen that $R(t;\, b \mid 10) = R(t;\, b \mid 01)$.

We now use the expressions that we have derived for the distribution function R with known starting and terminal machine states to derive new expressions for the distribution function F with known starting and terminal machine states. The modification of the equivalence (38) is immediately obvious,

$$F(q/p;\, t+q/p \mid 11) = 1 - R(t;\, \lambda\, q/p \mid 11),$$

and we therefore can write

$$(45)\quad F(t;T \mid 11) = 1 - \frac{\mu\left(1-F(t;\ T\ \mid\ 1)\right) + \lambda\ e^{-\lambda t-\mu(T-t)} \sum_{v=0}^{\infty} (-1)^v \left(\frac{\lambda(T-t)}{\mu t}\right)^{v/2} I_v\left(2\sqrt{\mu\lambda t(T-t)}\right)}{\mu + \lambda\ e^{-(\lambda+\mu)T}},$$

$$(46)\quad F(t;T \mid 10) = 1 - \frac{\left(1-F(t;\ T\ \mid\ 1)\right) - e^{-\lambda t-\mu(T-t)} \sum_{v=0}^{\infty} (-1)^v \left(\frac{\lambda(T-t)}{\mu t}\right)^{v/2} I_v\left(2\sqrt{\mu\lambda t(T-t)}\right)}{1 - e^{-(\lambda+\mu)T}},$$

$$(47)\quad F(t;T \mid 00) = 1 - \frac{\lambda\left(1-F(t;\ T\ \mid\ 1)\right) - \mu\ e^{-\lambda t-\mu(T-t)} \sum_{v=0}^{\infty} (-1)^v \left(\frac{\lambda(T-t)}{\mu t}\right)^{\frac{v+1}{2}} I_{v+1}\left(2\sqrt{\mu\lambda t(T-t)}\right)}{\lambda + \mu\ e^{-(\lambda+\mu)T}},$$

and

$$(48)\quad F(t;T \mid 10) = F(t;T \mid 01).$$

Note that in all of the distributions that we have derived in this section, the terms of the infinite series alternate in sign. This is fortunate, since we can exploit numerical methods such as Euler's transformation to accelerate the convergence of these series (Press et al., 1989). As discussed earlier, the modified Bessel functions of order v can be computed using one of several available algorithms, such as Cody (1983) or the one described in Press et al. (1989).

**Laplace transform with known starting and terminal machine states**

We now derive an expression for the Laplace transform of $R(t;\lambda\ q/p \mid 10)$,

$$\mathcal{L}\{\frac{R(t;\ b\mid 1) - e^{-b-\mu t}\sum_{v=0}^{\infty}(-1)^v\left(\frac{\lambda t}{\mu q/p}\right)^{v/2} I_v\left(2\sqrt{\mu bt}\right)}{1 - e^{-(\lambda+\mu)(t+q/p)}}\}.$$

We begin by rewriting $1/P_{10}(T)$ as

$$\frac{1}{P_{10}(T)} = \frac{1}{1-e^{-(\lambda+\mu)T}} = \sum_{n=0}^{\infty}\left(e^{-(\lambda+\mu)T}\right)^n \qquad \text{for } \left|e^{-(\lambda+\mu)T}\right| < 1.$$

Note that for all positive T, the convergence condition is satisfied. Our problem can therefore be rewritten as

$$\sum_{n=0}^{\infty}\mathcal{L}\left\{\left(e^{-(\lambda+\mu)(t+q/p)}\right)^n R(t;\ b\mid 1) - \left(e^{-(\lambda+\mu)(t+q/p)}\right)^n e^{-b-\mu t}\sum_{v=0}^{\infty}(-1)^v\left(\frac{\lambda t}{\mu q/p}\right)^{v/2} I_v\left(2\sqrt{\mu bt}\right)\right\}.$$

Noting that multiplying a function of t by **exp**(-at) replaces s by s+a in its transform, we conclude that

$$\begin{aligned}
&\sum_{n=0}^{\infty}\mathcal{L}\{\left(e^{-(\lambda+\mu)(t+q/p)}\right)^n R(t;\ b\mid 1)\} \\
&= \sum_{n=0}^{\infty}\left(e^{-(\lambda+\mu)q/p}\right)^n R^*(s+(\lambda+\mu)n;\ b\mid 1) \\
&= \sum_{n=0}^{\infty}\left(e^{-(\lambda+\mu)q/p}\right)^n \frac{1}{s+(\lambda+\mu)\ n}\ \mathbf{exp}\left(-b+b\frac{\mu}{s+\mu+(\lambda+\mu)\ n}\right) \\
&= \sum_{n=0}^{\infty}\frac{1}{s+(\lambda+\mu)\ n}\ \mathbf{exp}\left(-(\lambda+\mu)n\frac{q}{p}-b+b\frac{\mu}{s+\mu+(\lambda+\mu)\ n}\right)
\end{aligned}$$

The more difficult half of the problem is to find

$$\sum_{n=0}^{\infty} \mathcal{L}\{ \left(e^{-(\lambda+\mu)(t+q/p)}\right)^n e^{-b-\mu t} \sum_{v=0}^{\infty} (-1)^v \left(\frac{\lambda t}{\mu q/p}\right)^{v/2} I_v\left(2\sqrt{\mu b t}\right) \}.$$

We begin by constructing

$$\sum_{n=0}^{\infty} \mathcal{L}\{ e^{-b-\mu t} \sum_{v=0}^{\infty} (-1)^v \left(\frac{\lambda t}{\mu q/p}\right)^{v/2} I_v\left(2\sqrt{\mu b t}\right) \}$$

from the basic rules of Laplace transforms. As before, the symbol $\Leftrightarrow$ will be used to represent that the expression on the left is the Laplace transform of the expression on the right. We saw in the previous section during our derivation of R(t; b | 1) that

$$\frac{1}{\mu b}\left(\frac{\mu b}{s}\right)^{v+1} \mathbf{exp}(\mu b/s) \Leftrightarrow (\mu b t)^{v/2} I_v\left(2\sqrt{\mu b t}\right), \quad v > -1.$$

Multiplying both sides by the constant $\mu^{-v}\,(q/p)^{-v}\,(-1)^v$ yields

$$(-1)^v \lambda^v \left(\frac{1}{s}\right)^{v+1} \mathbf{exp}(\mu b/s) \Leftrightarrow (-1)^v \left(\frac{\lambda t}{\mu q/p}\right)^{v/2} I_v\left(2\sqrt{\mu b t}\right), \quad v > -1.$$

Next, we replace s by $s + \mu$, which is equivalent to multiplying the inverse transform by **exp**$(-\mu t)$, and then scale both sides by the constant **exp**$(-b)$. We obtain

$$(-1)^v \lambda^v \left(\frac{1}{s+\mu}\right)^{v+1} \mathbf{exp}\left(b\frac{\mu}{s+\mu}\right) \Leftrightarrow \mathbf{exp}(-\mu t)\,(-1)^v \left(\frac{\lambda t}{\mu q/p}\right)^{v/2} I_v\left(2\sqrt{\mu b t}\right), \quad v > -1.$$

and

$$(-1)^v \frac{1}{\lambda}\left(\frac{\lambda}{s+\mu}\right)^{v+1} e^{-b+b\frac{\mu}{s+\mu}} \Leftrightarrow e^{-\mu t-b}(-1)^v\left(\frac{\lambda t}{\mu q/p}\right)^{v/2} I_v\left(2\sqrt{\mu b t}\right),\ v>-1.$$

The last step is to sum over v from zero to infinity,

$$\frac{1}{\lambda} e^{-b+b\frac{\mu}{s+\mu}} \sum_{v=0}^{\infty}(-1)^v\left(\frac{\lambda}{s+\mu}\right)^{v+1} \Leftrightarrow e^{-\mu t-b}\sum_{v=0}^{\infty}(-1)^v\left(\frac{\lambda t}{\mu q/p}\right)^{v/2} I_v\left(2\sqrt{\mu b t}\right),$$

which, after simplification, gives

$$\left(\frac{1}{s+\lambda+\mu}\right) e^{-b+b\frac{\mu}{s+\mu}} \Leftrightarrow e^{-\mu t-b}\sum_{v=0}^{\infty}(-1)^v\left(\frac{\lambda t}{\mu q/p}\right)^{v/2} I_v\left(2\sqrt{\mu b t}\right).$$

We now see that

$$\begin{aligned}
&\sum_{n=0}^{\infty}\mathcal{L}\left\{\left(e^{-(\lambda+\mu)(t+q/p)}\right)^n e^{-b-\mu t}\sum_{v=0}^{\infty}(-1)^v\left(\frac{\lambda t}{\mu q/p}\right)^{v/2} I_v\left(2\sqrt{\mu b t}\right)\right\}\\
&\quad=\sum_{n=0}^{\infty}\left(e^{-(\lambda+\mu)q/p}\right)^n \frac{1}{s+(\lambda+\mu)(n+1)}\exp\left(-b+b\frac{\mu}{s+\mu+(\lambda+\mu)n}\right).\\
&\quad=\sum_{n=0}^{\infty}\frac{1}{s+(\lambda+\mu)(n+1)}\exp\left(-(\lambda+\mu)n\frac{q}{p}-b+b\frac{\mu}{s+\mu+(\lambda+\mu)n}\right).
\end{aligned}$$

Our Laplace transform of interest is thus, in total,

$$(49)\quad R^*(s;b\mid 10)=\sum_{n=0}^{\infty}\exp\left(-(\lambda+\mu)n\frac{q}{p}-b+b\frac{\mu}{s+\mu+(\lambda+\mu)n}\right)\times\left[\frac{1}{s+(\lambda+\mu)n}-\frac{1}{s+(\lambda+\mu)(n+1)}\right].$$

Further, we can also show by nearly identical arguments that

$$(50)\quad R^*(s;\, b \mid 11) = \sum_{n=0}^{\infty} \left( -\frac{\lambda}{\mu} e^{-(\lambda+\mu)q/p} \right)^n \exp\left( -b + b \frac{\mu}{s + \mu + (\lambda+\mu)\, n} \right) \times \left[ \frac{1}{s + (\lambda+\mu)\, n} + \frac{\lambda}{\mu} \frac{1}{s + (\lambda+\mu)\,(n+1)} \right].$$

$$(51)\quad R^*(s;\, b \mid 01) = R^*(s;\, b \mid 10).$$

$$(52)\quad R^*(s;\, b \mid 00) = \sum_{n=0}^{\infty} \left( -\frac{\mu}{\lambda} e^{-(\lambda+\mu)q/p} \right)^n \exp\left( -b + b \frac{\mu}{s + \mu + (\lambda+\mu)\, n} \right) \times \frac{\mu}{s + \mu + (\lambda+\mu)\, n} \left[ \frac{1}{s + (\lambda+\mu)\, n} - \frac{1}{s + (\lambda+\mu)\,(n+1)} \right].$$

**A simplified Laplace transform with known starting and terminal machine states**

It is important to note that the above infinite series for $R^*(s;\, b \mid 11)$ may not converge if $\lambda > \mu$, and $R^*(s;\, b \mid 00)$ may not converge if $\mu > \lambda$, since the parenthetical term could be greater than one (Knopp, 1956). However, one possible solution is to work with $R^*(s;\, b \mid 01)$ and $R^*(s;\, b \mid 10)$, and then employ an equation such as

$$R(t;\, b \mid 1) = R(t;\, b \mid 11)\, P_{11}(t+q/p) + R(t;\, b \mid 10)\, P_{10}(t+q/p).$$

Furthermore, we can use these results to derive formulae that are simpler and do not suffer from convergence problems. Rearranging the above equation and taking the Laplace transform of each side gives

$$\mathcal{L}\{ R(t;\, b \mid 11)\, P_{11}(t+q/p) \} = \mathcal{L}\{ R(t;\, b \mid 1) - R(t;\, b \mid 10)\, P_{10}(t+q/p) \}$$

$$= \mathcal{L}\{ R(t;\, b \mid 1) \} - \mathcal{L}\{ R(t;\, b \mid 10) \frac{\lambda}{\lambda+\mu} \left( 1 - e^{-(\lambda+\mu)(t+q/p)} \right) \}$$

$$= \frac{1}{s} \exp\left( -b + b \frac{\mu}{s + \mu} \right) - \frac{\lambda}{\lambda+\mu} \mathcal{L}\{ R(t;\, b \mid 10) \} +$$

$$\frac{\lambda}{\lambda+\mu} e^{-(\lambda+\mu)q/p} \mathcal{L}\{ R(t; b \mid 10) e^{-(\lambda+\mu)t} \}$$

$$= \frac{1}{s} \exp\left( -b + b\frac{\mu}{s+\mu} \right) -$$

$$\frac{\lambda}{\lambda+\mu} \sum_{n=0}^{\infty} \exp\left( -(\lambda+\mu)n\frac{q}{p} - b + b\frac{\mu}{s+\mu+(\lambda+\mu)n} \right) \times$$

$$\left[ \frac{1}{s+(\lambda+\mu)n} - \frac{1}{s+(\lambda+\mu)(n+1)} \right] +$$

$$\frac{\lambda}{\lambda+\mu} \sum_{n=0}^{\infty} \exp\left( -(\lambda+\mu)(n+1)\frac{q}{p} - b + b\frac{\mu}{s+\mu+(\lambda+\mu)(n+1)} \right) \times$$

$$\left[ \frac{1}{s+(\lambda+\mu)(n+1)} - \frac{1}{s+(\lambda+\mu)(n+2)} \right]$$

Noting that the two infinite series are identical except for a shifted index, we can cancel almost all of the terms and obtain

$$\mathcal{L}\{ R(t; b \mid 11) P_{11}(t+q/p) \} =$$

$$\frac{1}{s} \exp\left( -b + b\frac{\mu}{s+\mu} \right) - \frac{\lambda}{\lambda+\mu} \exp\left( -b + b\frac{\mu}{s+\mu} \right) \left[ \frac{1}{s} - \frac{1}{s+(\lambda+\mu)} \right],$$

or

$$(53) \quad \mathcal{L}\{ R(t; b \mid 11) P_{11}(t+q/p) \} = \frac{1}{\lambda+\mu} \exp\left( -b + b\frac{\mu}{s+\mu} \right) \left[ \frac{\mu}{s} + \frac{\lambda}{s+(\lambda+\mu)} \right].$$

The left-hand side can be interpreted as Pr{downtime $\leq t$ & $\alpha(t+q/p) = 1 \mid \alpha(0) = 1$}. We will see later that this probability will be very useful. Of course, if $R(t; b \mid 11)$ is desired instead, it is a simple matter to scale by $P_{11}(t+q/p)$.

From the above development we also see that

$$(54)\quad \mathcal{L}\{\, R(t; b \mid 10)\, P_{10}(t+q/p)\,\} = \frac{\lambda}{\lambda+\mu}\exp\left(-b+b\frac{\mu}{s+\mu}\right)\left[\frac{1}{s} - \frac{1}{s+(\lambda+\mu)}\right].$$

Finally, using the equation

$$R(t; b \mid 0) = R(t; b \mid 01)\, P_{01}(t+q/p) + R(t; b \mid 00)\, P_{00}(t+q/p),$$

we can also obtain, by similar argument,

$$(55)\quad \mathcal{L}\{\, R(t; b \mid 00)\, P_{00}(t+q/p)\,\} = \exp\left(-b+b\frac{\mu}{s+\mu}\right)\left[\frac{\mu}{s+\mu}\frac{1}{s} - \frac{\mu}{\lambda+\mu}\frac{1}{s} + \frac{\mu}{\lambda+\mu}\frac{1}{s+(\lambda+\mu)}\right].$$

and

$$(56)\quad \mathcal{L}\{\, R(t; b \mid 01)\, P_{01}(t+q/p)\,\} = \frac{\mu}{\lambda+\mu}\exp\left(-b+b\frac{\mu}{s+\mu}\right)\left[\frac{1}{s} - \frac{1}{s+(\lambda+\mu)}\right].$$

To use these results to compute the distribution F, we begin with the equation

$$F(q/p; t+q/p \mid 11) = 1 - R(t; \lambda\, q/p \mid 11),$$

and multiplying both sides by $P_{11}(t+q/p)$, obtain

$$P_{11}(t+q/p)\, F(q/p; t+q/p \mid 11) = P_{11}(t+q/p) - P_{11}(t+q/p)\, R(t; \lambda\, q/p \mid 11).$$

This can be used in several ways, for instance,

$$(57)\quad F(q/p; t+q/p \mid 11) =$$

$$1 - \frac{1}{P_{11}(t+q/p)}\,\mathcal{L}^{-1}\left\{\frac{1}{\lambda+\mu}\exp\left(-b+b\frac{\mu}{s+\mu}\right)\left[\frac{\mu}{s}+\frac{\lambda}{s+(\lambda+\mu)}\right]\right\}.$$

Of course, analogous expressions can be written for the other three cases (00, 01, 10).

**An important property of the distribution function**

*Property 1.* $F(t; T \mid 1)$ is a non-increasing function of T.

While this is difficult to prove by calculus, the result follows immediately from the equivalence (38) between the distributions F and R. In particular, for any $\delta > 1$, $T \geq 0$ we wish to show that

$$F(t; \delta T \mid 1) \leq F(t; T \mid 1).$$

We can rewrite this using the equivalence (*) as

$$R(\delta T - t;\ \lambda t \mid 1) \geq R(T - t;\ \lambda t \mid 1).$$

Since $R(t; b \mid 1)$ is a non-decreasing function of t, the result follows.

By the same arguments, the above properties also hold for the CDFs $F(t; T \mid 0)$, $F(t; T \mid 11)$, $F(t; T \mid 10)$, $F(t; T \mid 01)$, and $F(t; T \mid 00)$.

## 2.6 Transient behavior of mean and variance of uptime over a fixed period of time

In this section we explore the transient behavior of the mean and variance of f(t; T | 1)/T (derived in Section 2.2) as we vary the parameter T. We are interested in both the behavior of the asymptotes and how quickly these functions approach their asymptotes.

Figures 2.1 – 2.6 depict the results. In each figure we vary the parameter T from 1 to 25. Figure 2.1 will serve as our base case, in which $\lambda = 1, \mu = 1$. We will subsequently investigate changes in the failure rate $\lambda$ and the repair rate $\mu$. We see that in the base case, the mean approaches the asymptote $\mu / (\lambda + \mu) = 50\%$ somewhat slowly, while the variance approaches its asymptote of $2 \lambda \mu / (\lambda + \mu)^3$ even more slowly. The asymptotic mean is sometimes called the *stand-alone availability.*

Next we increase the stand-alone availability (SAA) to 80% in two different ways. In Figure 2.2 we increase $\mu$ to 4, and in Figure 2.3 we decrease $\lambda$ to 0.25. We see vastly different results in each case. Increasing $\mu$ leads to a great reduction in the variance asymptote and results in much quicker convergence of both the mean and the variance to their asymptotes. Decreasing $\lambda$, however, yields a slight *increase* in the variance asymptote, reduces the rate of convergence of the variance to its asymptote, and does not improve the rate of convergence of the mean as much as increasing $\mu$.

Figures 2.4 and 2.5 tell a similar story. In each case we decrease the SAA to 20%, by increasing $\lambda$ to 4 in Figure 2.4, and decreasing $\mu$ to 0.25 in Figure 2.5. Increasing $\lambda$ is seen to somewhat improve the rate of convergence of the mean to its asymptote, greatly reduce the variance asymptote, and almost completely eliminate the transient effect

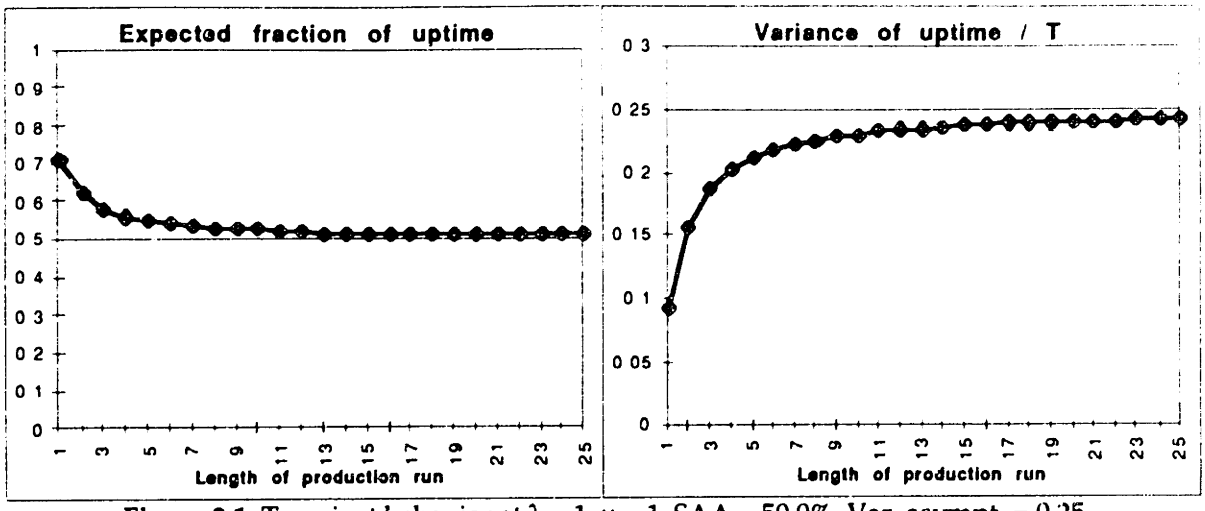


**Figure 2.1** Transient behavior at $\lambda = 1$, $\mu = 1$, SAA = 50.0%, Var. asympt. = 0.25

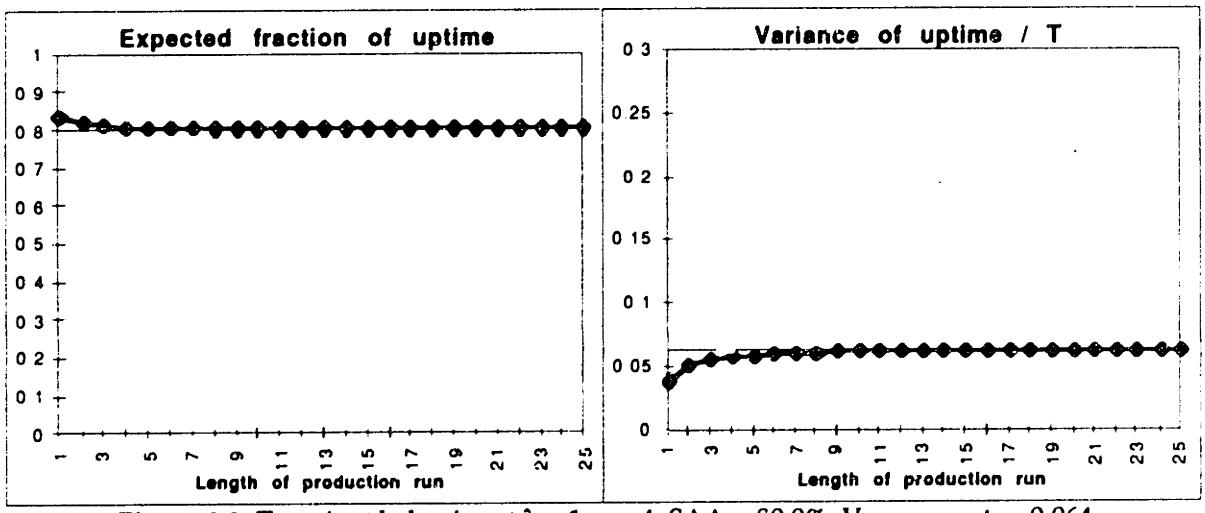


**Figure 2.2** Transient behavior at $\lambda = 1$, $\mu = 4$, SAA = 80.0%, Var. asympt. = 0.064

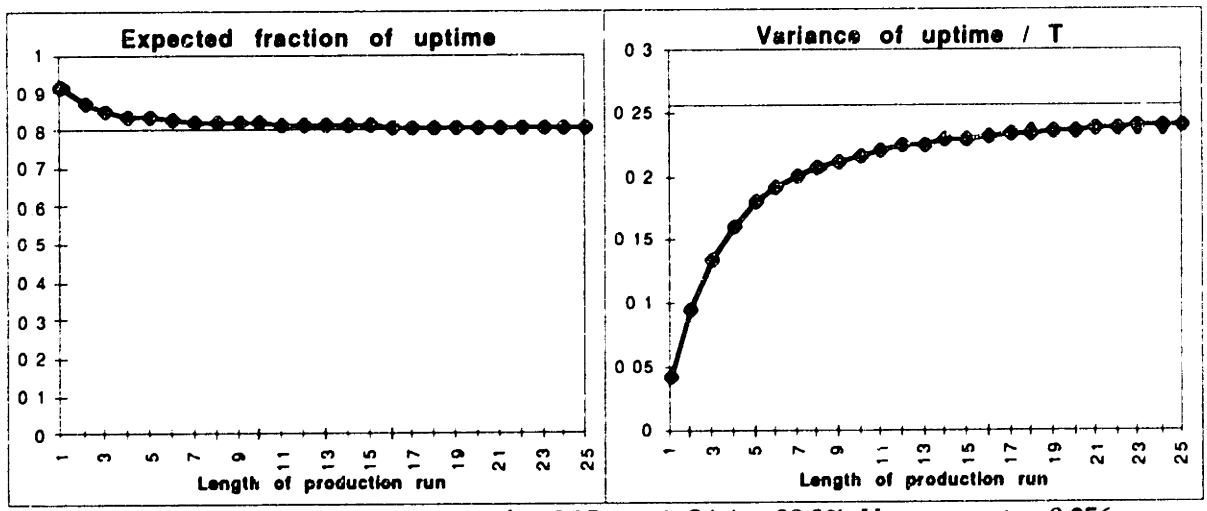


**Figure 2.3** Transient behavior at $\lambda = 0.25$, $\mu = 1$, SAA = 80.0%, Var. asympt. = 0.256

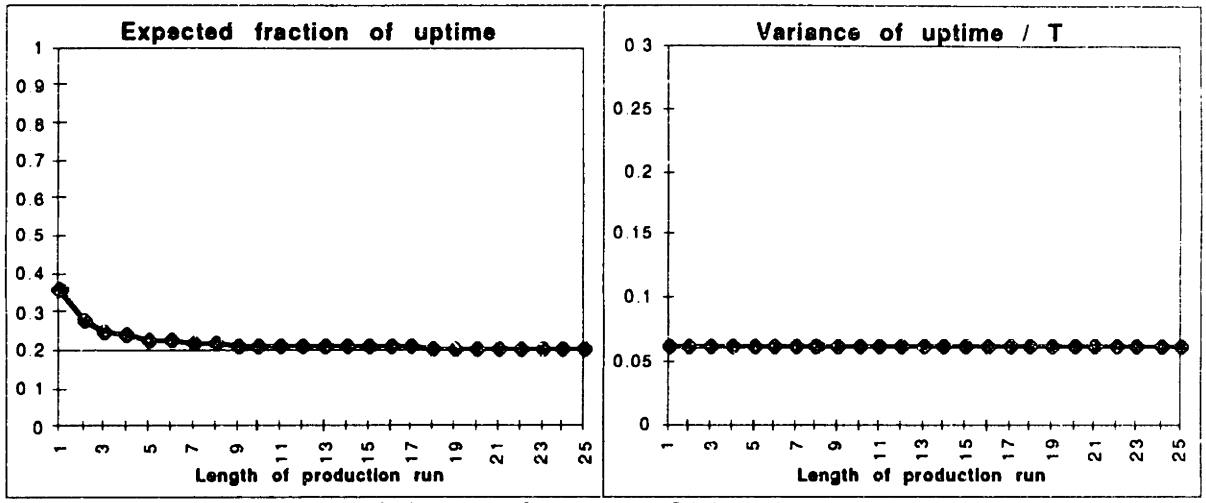


**Figure 2.4** Transient behavior at $\lambda = 4$, $\mu = 1$, SAA = 20.0%, Var. asympt. = 0.064

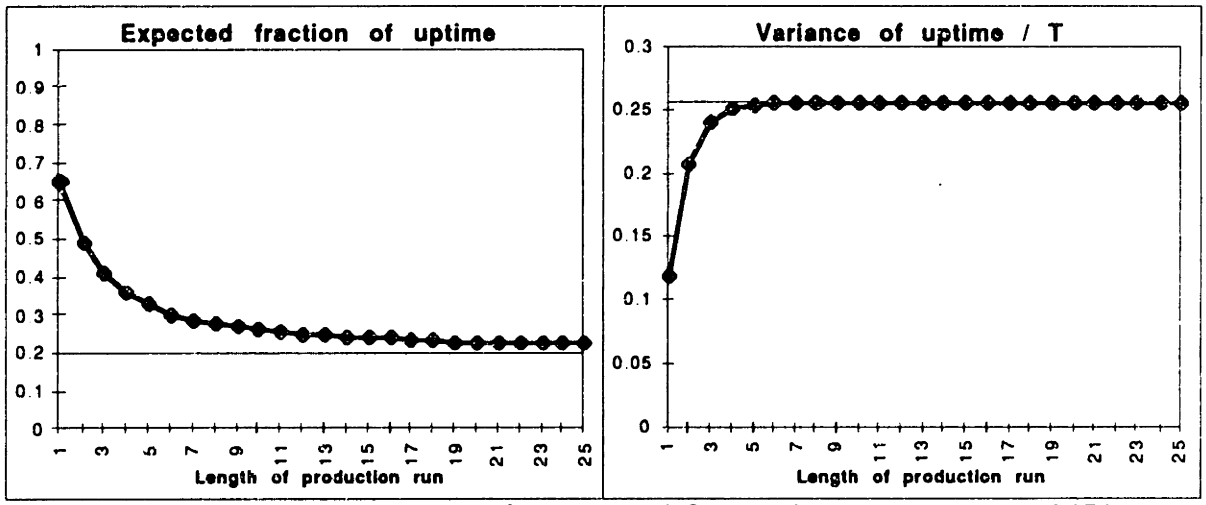


**Figure 2.5** Transient behavior at $\lambda = 1$, $\mu = 0.25$, SAA = 20.0%, Var. asympt. = 0.256

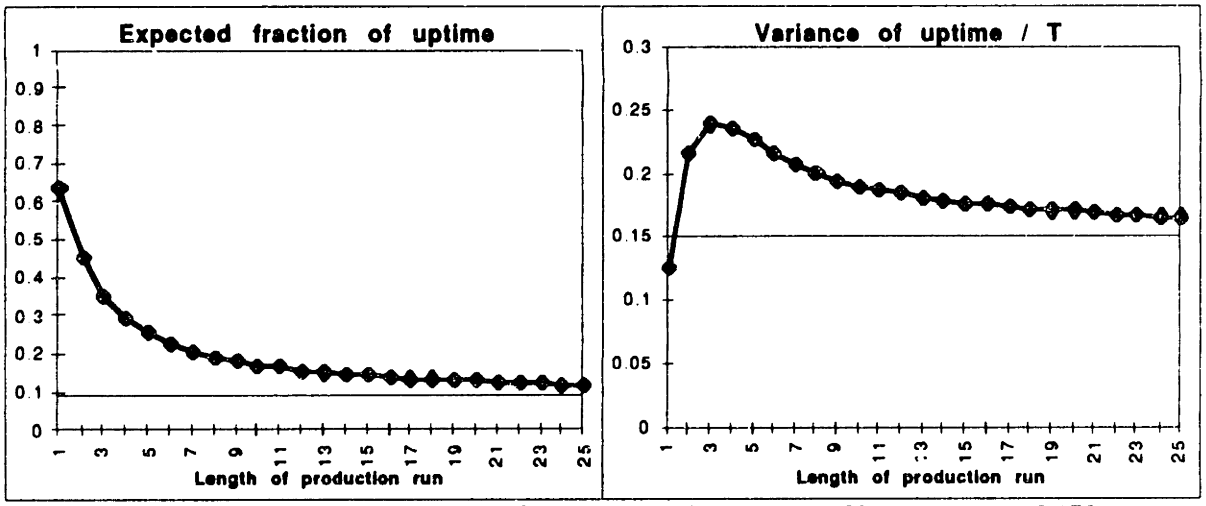


**Figure 2.6** Transient behavior at $\lambda = 1$, $\mu = 0.1$, SA 9.1%, Var. asympt. = 0.150

associated with the variance. In sharp contrast, decreasing $\mu$ yields a slight *increase* in the variance asymptote, does not dramatically improve the rate of convergence of the variance to its asymptote, and worsens the rate of convergence of the mean. We will analytically examine this phenomena below.

Lastly, we observe in Figure 2.6 some of the unusual behavior that can exist at a very low SAA (9%). Here we observe that the variance is initially below the asymptote (at T = 1), increases above the asymptote, then decreases to the asymptote as T increases to infinity. We can understand this behavior intuitively, recognizing that the MTTR is 10 hours, that is, any failure leaves the system failed for a long period of time. This, in combination with the fact that the machine is not failed at time 0 and may still be working two or three hours later (since the MTBF is 1 hour), has the effect of significantly increasing the variability due to the initial startup effect.

There are two ways to improve the reliability of the machine. One is to increase the repair rate $\mu$, and the other is to lower the failure rate $\lambda$. The above exploration suggests that for any fixed SAA, we would prefer to have a higher $\mu$ instead of a lower $\lambda$. We now show this analytically. Recall from Section 2.2 that the asymptotic variance of $f(t; T \mid 1)/T$ is $2\lambda\mu/(\lambda + \mu)^3$. If we increase $\mu$ to $\mu+\Delta$, then we must increase $\lambda$ to $\lambda\,(\mu+\Delta)/\mu$ in order to maintain a constant SAA. As a result, the asymptotic variance of $f(t; T \mid 1)/T$ becomes

$$\frac{2\lambda\mu}{(\lambda+\mu)^3}\,\frac{\mu}{\mu+\Delta}$$

which is a decreasing function of $\Delta$. As a result, increasing $\mu$ while holding the SAA constant decreases the asymptotic variance of $f(t; T \mid 1)/T$. It also follows that

decreasing $\lambda$ while holding the SAA constant increases the asymptotic variance. It is also true (but harder to show) that the same is true of the transient variance given by (25).

We now explore relaxation time as one metric for the rate of convergence of the stochastic process, as described by Keilson (1979). First, consider the discrete state Markov Process with two states denoted zero and one. Let the transitions from state zero to state one occur with rate $\mu$, and the transitions from one to zero with rate $\lambda$ (we do not permit self-transitions), and let the system be in state one at time 0. This discrete state Markov Process is then equivalent to the machine failure process that is the subject of this chapter if we interpret the time that the process spends in state one (zero) as machine uptime (downtime).

Given a function $f$ defined on the Markov chain state space $N$, Keilson defines the covariance function $r_f(\tau)$ as

$$r_f(\tau) = \sum_{m \in N} \sum_{n \in N} f(m)\, e_m\, (p_{mn}(\tau) - e_n)\, f(n)$$

where $e_m$ are the ergodic probabilities and $p_{mn}(\tau)$ represents the probability that the chain is in state n at time $t+\tau$ if the chain is in state m at time t. Let us define the function $f$ such that $f(0) = 0$ and $f(1) = 1$. Thus the function $f$ serves as an indicator function for machine uptime. In this case $r_f(\tau)$ reduces to

$$r_f(\tau) = e_1\, (p_{11}(\tau) - e_1)$$

Furthermore, for our chain,

$$e_1 = \frac{\mu}{\lambda + \mu}$$

and it is easily shown (Barlow and Proschan, 1965 or Gross and Harris, 1985) that

$$p_{11}(\tau) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)\tau}.$$

Thus, given our definition of *f*, the covariance function for this process after simplification is

$$r_f(\tau) = \frac{\lambda\mu}{(\lambda + \mu)^2} e^{-(\lambda+\mu)\tau}.$$

The relaxation time for the process is then defined as

$$T_{REL} = \int_0^\infty \frac{r_f(\tau)\ d\tau}{r_f(0)}$$

which is easily seen to equal $1/(\lambda + \mu)$ in this case. This is consistent with our empirical observations above: the rate of convergence of expected fraction of machine uptime appears affected equally by $\lambda$ and $\mu$. This is also consistent with results of Baxter (1985), who shows that for a machine starting in steady state, the autocorrelation of the indicator function *f* is given by $\exp(-(\lambda + \mu)\ |\tau|)$. Furthermore, it is easy to see from the expressions for $P_{11}(T)$ and $P_{01}(T)$ that the rate of convergence of the so-called *availability coefficient* $\Pr\{\ \alpha(t) = 1\ \}$ is exponential with rate determined by $\lambda + \mu$. See Gnedenko et al. (1969) for a further discussion.

Keilson points out the familiarity of the relaxation time expression with a survival function, and notes that the relaxation time is essentially a survival function for the dependence of the process on its initial condition. Keilson also shows that the survival function can be rewritten in terms of the fundamental matrix of the process and then it is easily seen that the relaxation time derived above is in fact the largest eigenvalue of the fundamental matrix. For another discussion of relaxation time, see Morse (1958).

## 2.7 Normal approximation to the distribution of parts produced over a fixed period of time

In this section we briefly explore the accuracy of approximating f(t; T | 1) by a Normal distribution. Takács (1957a, 1957b) has proven that this distribution is asymptotically Normal as a function of T. We will therefore approximate f(t; T | 1) by its first two moments, as derived in Section 2.2. Those results will serve as the mean and variance in our Normal approximation.

To facilitate numerical evaluation of the Normal approximation, we propose two metrics. The first is the so-called Kolmogorov distance (which we denote by $\overline{K}$), the largest absolute difference between the two cumulative distributions. The second is the average absolute difference between the two cumulative distributions over [0, T) (which we will denote by $\overline{E}$). For a rigorous discussion of these metrics, see Kalashnikov (1994).

In Figure 2.7 we plot f(t; T | 1) and the resulting Normal approximation for $\lambda = 2$, $\mu = 4$, and T = 5. We see that the approximation is quite good over the full range of the distribution ($\overline{E} = 0.5\%$, $\overline{K} = 2.0\%$). However, as T is decreased (with $\lambda$ and $\mu$ fixed), the approximation worsens. Figures 2.8 and 2.9 show the results for T = 2 ($\overline{E} = 1.3\%$, $\overline{K} = 3.7\%$) and T = 1 ($\overline{E} = 2.8\%$, $\overline{K} = 7.1\%$). A maximum absolute error of 7% in the cumulative distribution suggests that the approximation should not be used for numerical work other than first order approximation.

Intuitively, we would expect that it is not the magnitude of T that dictates the accuracy of the approximation, but rather, the number of failure/repair cycles that occur within the interval [0, T). Figure 2.10 confirms this, where T is held at 1 while $\lambda$ is increased to

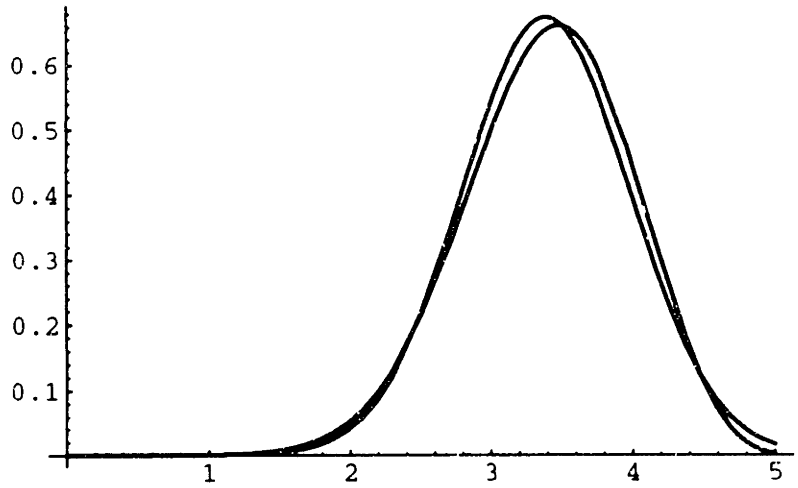


**Figure 2.7** f(t;T | 1) and normal approximation at $\lambda = 2, \mu = 4, T = 5$

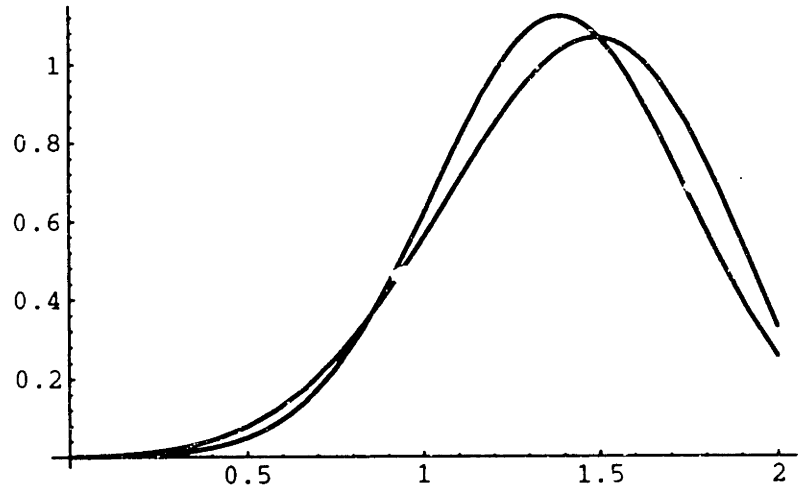


**Figure 2.8** f(t;T | 1) and normal approximation at $\lambda = 2, \mu = 4, T = 2$

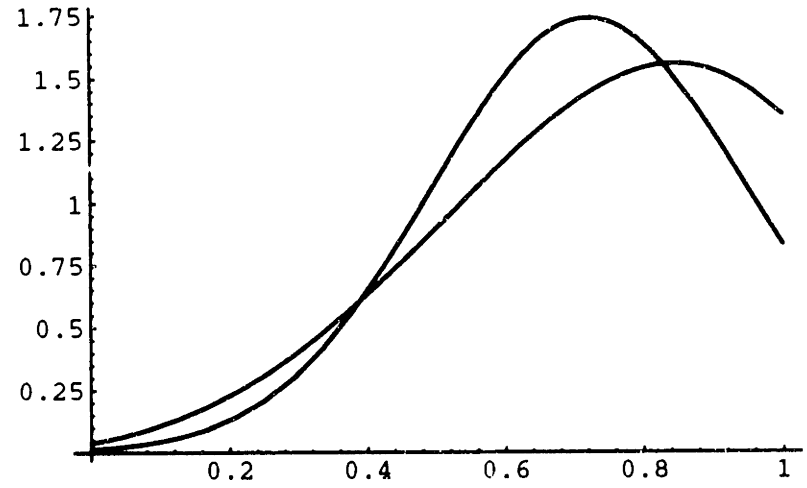


**Figure 2.9** f(t;T | 1) and normal approximation at $\lambda = 2, \mu = 4, T = 1$

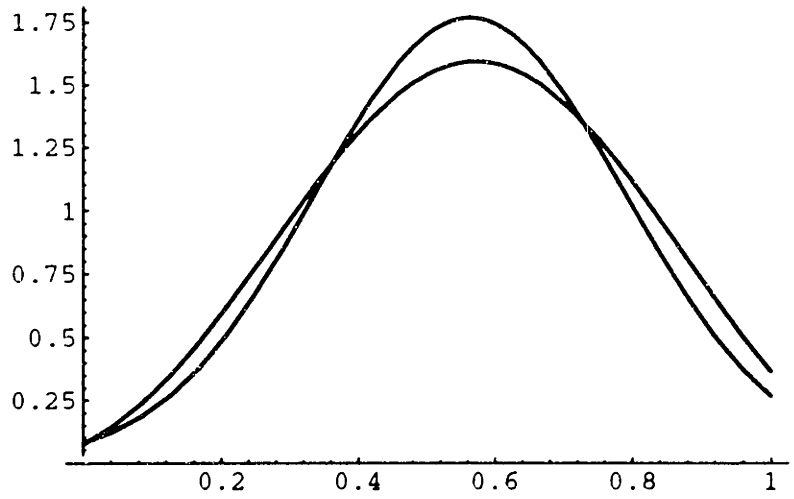


**Figure 2.10** f(t;T | 1) and normal approximation at $\lambda = 4$, $\mu = 4$, $T = 1$

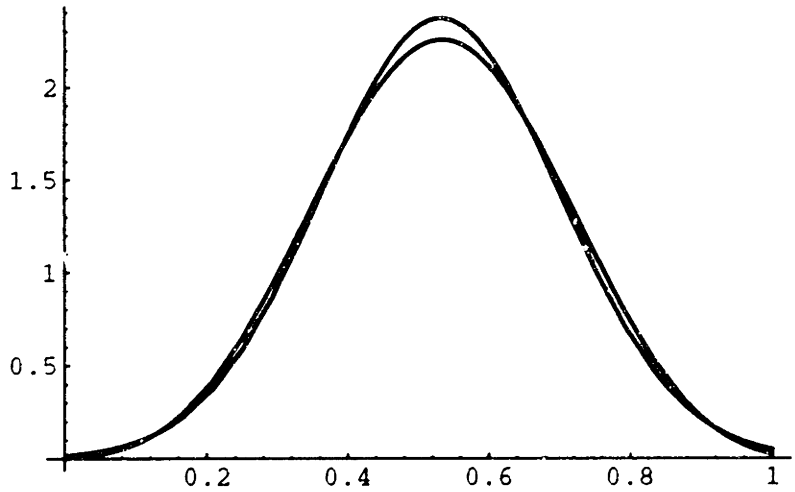


**Figure 2.11** f(t;T | 1) and normal approximation at $\lambda = 8$, $\mu = 8$, $T = 1$

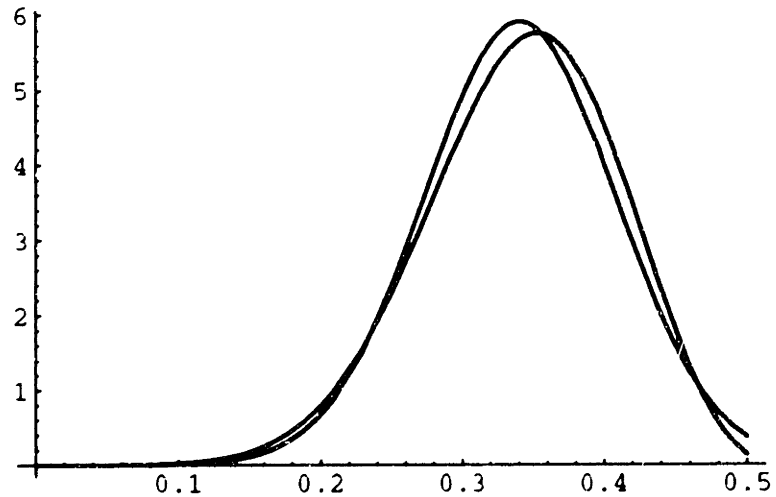


**Figure 2.12** f(t;T | 1) and normal approximation at $\lambda = 15$, $\mu = 30$, $T = 0.5$

4, and a dramatic improvement results ($\overline{E}$ = 1.1%, $\overline{K}$ = 2.3%). In Figure 2.11, $\lambda$ and $\mu$ are increased to 8 (with T held at 1) and the approximation improves further ($\overline{E}$ = 0.4%, $\overline{K}$ = 1.0%).

Figure 2.12 further demonstrates this principle. Decreasing T to 0.5 but increasing $\lambda$ to 15 and $\mu$ to 30 results in an approximation which is once again reasonable over the full range of the distribution ($\overline{E}$ = 0.7%, $\overline{K}$ = 2.4%).

To see this analytically, recall from equation (3) that

$$f(t; T \mid 1) = \left[\lambda\mu t \, \frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}} + \lambda \, I_0\left(2\sqrt{x}\right)\right] e^{-\lambda t - \mu(T-t)} + u_0(T-t)\, e^{-\lambda T}, \qquad 0 \le t \le T.$$

It can be seen that if t and T are scaled by k, and $\lambda$ and $\mu$ by 1/k, then $f(t; T \mid 1) / T$ is unchanged. Therefore, the shape of the density $f(t; T \mid 1)$ is determined not by T alone, but rather, by the relative magnitude of T in relation to $\lambda$ and $\mu$.

We have shown that in certain circumstances, the Normal approximation can be quite good even in short time intervals. Conversely, the approximation can be quite poor even over long time intervals. The conclusion we reach is that care must be taken before the approximation is used.

## 2.8 Distribution of time to produce multiple batches of parts

We now turn our attention to the problem of finding the distribution of time to produce multiple batches on a single machine. The probability density function of the time to produce n different batches is the convolution of n probability density functions of type r(t; b). Since the Laplace transform r*(s; b) of r(t; b) is known, we can find the Laplace transform of the density of time to produce multiple batches by simply multiplying the transforms of the probability density function of time to produce each batch. Given the transform of the density of time to produce multiple batches, it is then easy to obtain the moments of the time to produce multiple batches. The remainder of this section deals with the more difficult problem of finding the probability density function of time to produce multiple batches.

First consider the simplest possible problem: two batches with equal failure and repair rates; that is, $b_1 = b_2 = b$ and $\mu_1 = \mu_2 = \mu$. This problem is equivalent to finding the two-fold convolution of r(t; b) which, from its Laplace transform, is easily seen to be equivalent to a one-batch problem with failure rate 2b. The intuition behind this result is the following. Since each process is a Compound Poisson process on the interval [0, 1] with rate b, the superposition of the two processes is a Compound Poisson process on the interval [0, 1] with rate 2b. It is also easily seen from the Laplace transform that this easily generalizes to the case $b_1 \neq b_2$, in which the problem is equivalent to a one-batch problem with failure rate $b_1 + b_2$. These results are a consequence of the fact that the superposition of two Poisson processes with rates $\lambda_1$ and $\lambda_2$ is itself a Poisson process of rate $\lambda_1 + \lambda_2$ (Ross, 1989). Furthermore, this result extends directly to $n > 2$ batches.

The case with different repair rates ($\mu_1 \neq \mu_2$) is much more difficult. The Laplace transforms easily multiply but the product is not easily inverted. One reasonable guess

is to assume that this problem is equivalent to the one-batch problem with failure rate $b_1 + b_2$ and repair rate given by

$$\mu_1 \frac{b_1}{b_1 + b_2} + \mu_2 \frac{b_2}{b_1 + b_2}.$$

This approximation is exact for $\mu_1 = \mu_2$ and worsens as $|\mu_1 - \mu_2|$ grows. In fact, for reasonable values of $\mu_1$, $\mu_2$, $b_1$ and $b_2$, the approximation is not very good.

**A two moment approximation**

We instead propose the following: let us assume that the two-batch distribution can be represented as an equivalent one-batch distribution; this is a reasonable guess since the case $\mu_1 = \mu_2$ reduced to a one-batch distribution. Further, since we can find the moments of the two-batch distribution, we can find the $\mu_0$ and $b_0$ for the one-batch distribution such that the first two moments of the one-batch distribution are the same as the first two moments of the two-batch distribution. This equates to solving two equations (one for each moment) in two unknowns ($\mu_0$ and $b_0$):

$$\frac{b_0}{\mu_0} = \frac{b_1}{\mu_1} + \frac{b_2}{\mu_2},$$
$$\frac{2b_0}{\mu_0^2} = \frac{2b_1}{\mu_1^2} + \frac{2b_2}{\mu_2^2},$$

whose solution is

$$b_0 = \frac{(b_1\mu_2 + b_2\mu_1)^2}{b_1\mu_2^2 + b_2\mu_1^2},$$
$$\mu_0 = \frac{(b_1\mu_2 + b_2\mu_1)(\mu_1\mu_2)}{b_1\mu_2^2 + b_2\mu_1^2}.$$

This approximation has the desirable property that in the case $\mu_1 = \mu_2 = \mu$ for which the exact result is known, the approximation produces the correct exact result $b_0 = b_1 + b_2$ and $\mu_0 = \mu$.

It can be shown that the third moment of the two-batch distribution does not equal the third moment of the one-batch distribution with $\mu_0$ and $b_0$ given above. From this we can conclude that *in general, there does not exist an equivalent one-batch distribution for the multiple batch distribution.* In other words, the resulting stochastic process is no longer Compound Poisson.

This two-batch procedure can be applied iteratively to approximate the distribution of $n \geq 3$ batches as follows: compute the one-batch approximation to the two-batch distribution yielding a one-batch distribution. Use this result along with the parameters for the third batch to compute the one-batch approximation to the three-batch distribution, and so forth.

**Accuracy of two moment approximation**

We would now like to evaluate the accuracy of the two moment approximation described above. Let us first take a moment to consider the limiting behavior of the multiple batch distribution. First note that as $b_1$ increases, the distribution of time to produce batch 1 approaches the Normal distribution. The same is true for batch 2 as $b_2$ increases. Therefore as $b_1$ and $b_2$ increase, the convolution of the two distributions approaches the Normal distribution, since the convolution of two Normal distributions is itself a Normal distribution. This result is discussed with great rigor and depth by Feller (1971). Since the Normal distribution is completely described by its first two moments, we can conclude that our two moment approximation is asymptotically exact.

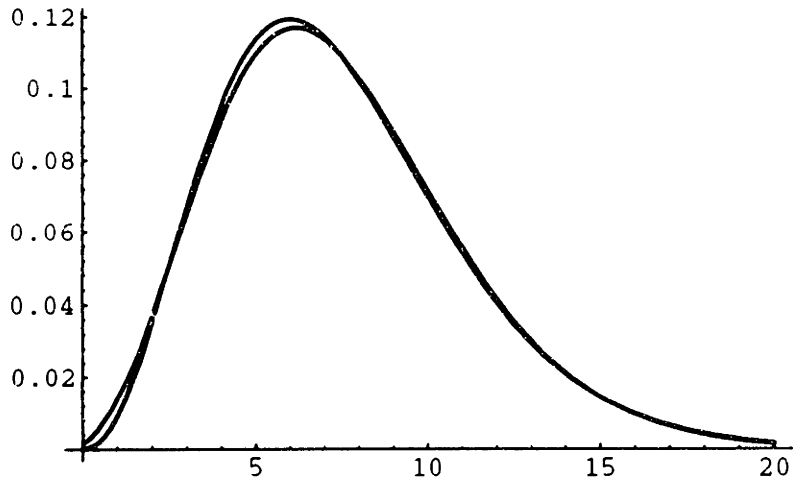


**Figure 2.13** Exact and approximate convolution of two densities of type r(t; b | 1) with parameters $b_1 = 6$, $b_2 = 6$, $\mu_1 = 4$, $\mu_2 = 1$

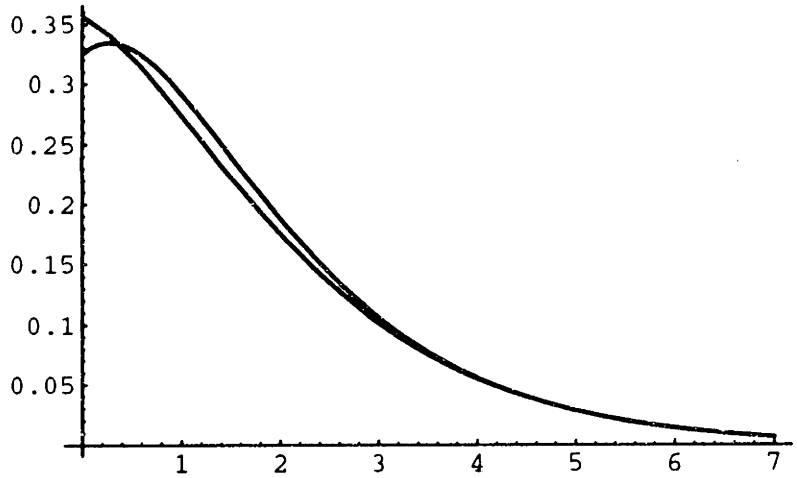


**Figure 2.14** Exact and approximate convolution of two densities of type r(t; b | 1) with parameters $b_1 = 1$, $b_2 = 1$, $\mu_1 = 1$, $\mu_2 = 2$

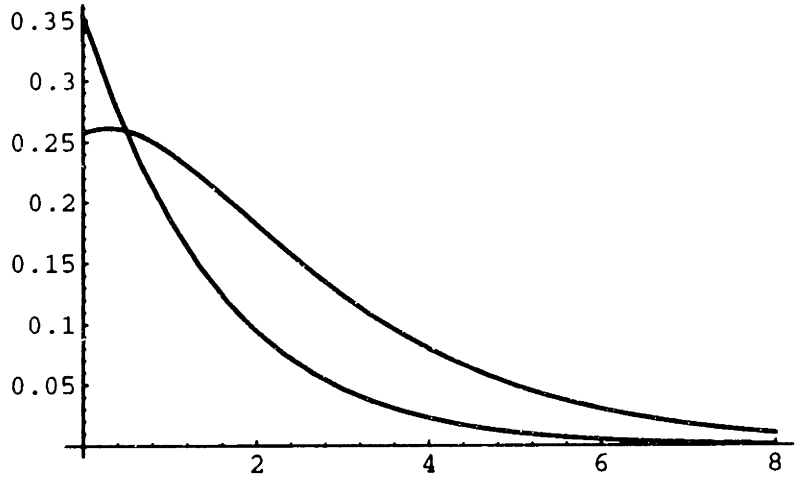


**Figure 2.15** Exact and approximate convolution of two densities of type r(t; b | 1) with parameters $b_1 = 2$, $b_2 = 0.1$, $\mu_1 = 4$, $\mu_2 = 0.4$

Accordingly, we limit our attention to modest values of b. To evaluate the accuracy of the approximation we compute the Kolmogorov distance $\overline{K}$, the largest absolute difference between the exact cumulative distribution and the approximation*. Three exact convolved densities with varied parameters and their approximations are plotted on the previous page. In all examples we assume that the machine is initially working. Figure 2.13 shows that the approximation is very good for $b_1 = b_2 = 6$, $\mu_1 = 4$, $\mu_2 = 1$ ($\overline{K}$ = 0.8%). This would correspond to running two batches for three hours with MTBF = 30 minutes, and MTTR = 15 minutes for one part and 60 minutes for the other. Since the approximation is asymptotically correct and is also exact when $\mu_1 = \mu_2$, it is encouraging that the approximation is quite good even with a relatively small b and with $\mu_1$ and $\mu_2$ differing by a factor of 4. Figure 2.14 shows the result in the case $b_1 = b_2 = 1$, $\mu_1 = 1$, $\mu_2 = 2$. We see that in this case, even with a small b, the quality of the approximation appears reasonable ($\overline{K}$ = 3.5%). The quality of the approximation does degrade if we continue to increase the difference between $\mu_1$ and $\mu_2$ while keeping b and $\mu$ small. Figure 2.15 shows that the approximation breaks down at $b_1 = 2$, $b_2 = 0.1$, $\mu_1 = 4$, $\mu_2 = 0.4$ ($\overline{K}$ = 37.2%).

**An equivalent convolution**

We have shown above how to approximate the distribution of time to produce multiple batches. This subsection examines the probability that we are able produce two batches with differing parameters in an interval of fixed length. We will show how to write this probability as a convolution, which is important for computational purposes. Without such a result, the evaluation of this probability would require a n-fold integral. If numerical integration were used, the computational effort would grow exponentially in n. Because we are able to express the probability of interest as a convolution of n

---

* Note that the average absolute difference between the two cumulative distributions $\overline{E}$ will be zero since the distribution and its approximation both have infinitely long right tails.

distributions, we can write the Laplace transform of this probability as a product of n terms. As a result, if numerical Laplace transform inversion is used, the computational effort grows only linearly in n.

We now wish to evaluate the probability that we are able to produce two batches of size $q_1$ and $q_2$ in an interval of length T, where the processing speeds are $p_1$ and $p_2$, the failure rates are $\lambda_1$ and $\lambda_2$, and the repair rates are $\mu_1$ and $\mu_2$. Assuming that we produce batch one first, denote the probability that we produce at most $q_2$ parts by the distribution function $G(q_2 ; T, q_1, p_1, p_2, \lambda_1, \lambda_2, \mu_1, \mu_2)$, which we will abbreviate as $G(q_2 ; T, q_1)$. This distribution function can be written as

$$G(q_2 ; T, q_1) = \Pr\{ \text{time to produce } 1^{st} \text{ batch} > T - q_2/p_2 \} +$$
$$\int_{q_1/p_1}^{T-q_2/p_2} \Pr\{ \#\text{ of parts produced in } 2^{nd} \text{ batch} \le q_2 \mid \text{time to produce } 1^{st} \text{ batch} = y \} \times \text{dens}\{ \text{time to produce } 1^{st} \text{ batch} = y \}\, dy$$

$$= \left[1 - R_1\left(T - \frac{q_1}{p_1} - \frac{q_2}{p_2};\ \lambda_1 \frac{q_1}{p_1}\right)\right] + \int_{q_1/p_1}^{T-q_2/p_2} F_2\left(\frac{q_2}{p_2};\ T - y\right) r_1\left(y - \frac{q_1}{p_1};\ \lambda_1 \frac{q_1}{p_1}\right) dy.$$

By the equivalence property,

$$= \left[1 - R_1\left(T - \frac{q_1}{p_1} - \frac{q_2}{p_2};\ \lambda_1 \frac{q_1}{p_1}\right)\right] +$$
$$\int_{q_1/p_1}^{T-q_2/p_2} \left[1 - R_2\left(T - \frac{q_2}{p_2} - y;\ \lambda_2 \frac{q_2}{p_2}\right)\right] r_1\left(y - \frac{q_1}{p_1};\ \lambda_1 \frac{q_1}{p_1}\right) dy,$$

$$= \left[1 - R_1\left(T - \frac{q_1}{p_1} - \frac{q_2}{p_2};\ \lambda_1 \frac{q_1}{p_1}\right)\right] +$$
$$R_1\left(T - \frac{q_1}{p_1} - \frac{q_2}{p_2};\ \lambda_1 \frac{q_1}{p_1}\right) - r_1\left(T - \frac{q_1}{p_1} - \frac{q_2}{p_2};\ \lambda_1 \frac{q_1}{p_1}\right) \bigstar R_2\left(T - \frac{q_1}{p_1} - \frac{q_2}{p_2};\ \lambda_2 \frac{q_2}{p_2}\right),$$

where ☆ denotes the convolution operator, and the argument of the convolution is $T - q_1/p_1 - q_2/p_2$. Not surprisingly, it is now seen that

$$G(q_2\,;\,T,\,q_1) = 1 - r_1\left(T - \frac{q_1}{p_1} - \frac{q_2}{p_2};\ \lambda_1\frac{q_1}{p_1}\right) \text{☆}\, R_2\left(T - \frac{q_1}{p_1} - \frac{q_2}{p_2};\ \lambda_2\frac{q_2}{p_2}\right),$$

that is, a convolution of the type that is the subject of this section. Through a similar derivation, we can also show that

$$G(q_2\,;\,T,\,q_1) = 1 - r_2\left(T - \frac{q_1}{p_1} - \frac{q_2}{p_2};\ \lambda_2\frac{q_2}{p_2}\right) \text{☆}\, R_1\left(T - \frac{q_1}{p_1} - \frac{q_2}{p_2};\ \lambda_1\frac{q_1}{p_1}\right).$$

Although we have derived this result for the case of two batches, the result extends by induction to any number of batches.

In the discussion above we have not described how to handle known initial (and possibly ending) machine states. Suppose for example that we know that the machine is initially working but is failed T time units later. Then

$$G(q_2\,;\,T,\,q_1 \mid 10) = 1 - r_1\left(T - \frac{q_1}{p_1} - \frac{q_2}{p_2};\ \lambda_1\frac{q_1}{p_1} \,\middle|\, 1\right) \text{☆}\, R_2\left(T - \frac{q_1}{p_1} - \frac{q_2}{p_2};\ \lambda_2\frac{q_2}{p_2} \,\middle|\, 10\right).$$

In general, the density for the first part should be conditioned on the initial state of the machine. The distribution for the last part should be conditioned on 11 or 10 depending if the ending machine state is working or failed (respectively). The distribution for each intermediate part should be conditioned on the machine initially working, since the

machine must have been working at the end of the previous batch (a failed machine can not complete a batch).

To prevent misunderstanding, we wish to highlight the fact that *G is not a convolution of two F distributions*. Such a convolution would correspond to the probability that we produce at most q parts given two production opportunities, one of length $T_1$ and the other of length $T_2$. Although such a convolution may have important practical uses, we do not need it for our work and do not consider it here. We mention only that the convolution of two F distributions is itself a distribution of type F if and only if the failure *and* repair rates for the two distributions are identical.

## Appendix: Algorithms for numerical Laplace transform inversion

Since the Laplace transforms derived earlier in this chapter must be inverted numerically, identifying effective inversion algorithms is important for implementation of our results. The classic paper by Davies and Martin (1979) compares a variety of different Laplace transform inversion methods and measures their applicability to a variety of different types of inversion problems. Their broad conclusion is that Laguerre polynomial methods are the most effective, although no one method is optimal in all circumstances. We are not aware of a more recent survey that has followed the improvements in algorithms for numerical Laplace transform inversion over the last 15 years. In this section we present a small study of our own, briefly describing our experience using two relatively new algorithms for Laplace transform inversion that have appeared in the literature: an implementation of Talbot's Method (a contour integration method) and an implementation of Weeks' Method (a Laguerre polynomial method).

### Talbot's Method

Murli and Rizzardi (1990) have developed an implementation of Talbot's (1979) method for numerically inverting Laplace transforms via contour integration. The method requires that the Laplace transform f*(s) satisfy the following criteria:

(1) The locations of the singularities $s_1, s_2, \ldots$ of f*(s) must be known; let $\gamma_0 \equiv \max_j \mathbf{Re}(s_j)$

(2) $| f^*(s) | \to 0$ uniformly as $| s | \to \infty$ in $\mathbf{Re}(s) < \gamma_0$,

(3) $| \mathbf{Im}(s_j) | < K \ \forall \ j$, and K is known.

Further, Murli and Rizzardi's algorithm performs best when the following additional conditions are satisfied:

(4) $| \mathbf{Re}(s_j) | \leq 0 \ \forall \ j$.

(5) No singularities exist at zero.

Our tests were performed with the function

$$R^*(s;\ b \mid 1) \ = \ \frac{1}{s} \ \mathbf{exp}\left(-b + b\frac{\mu}{s+\mu}\right).$$

Based on the behavior of this function in the complex plane (Copson, 1935), we can conclude that this function is ideally suited for use with Murli and Rizzardi's algorithm. The two singularities are a simple pole at zero and an essential singularity at $-\mu$, and thus $\gamma_0 = 0$. Condition (2) is satisfied since the exponential part of R*(s) approaches **exp**(-b) as $| s | \rightarrow \infty$, and therefore $R^*(s) \rightarrow 0$. Condition (3) is satisfied for any small positive $\varepsilon$, and fortunately (4) is satisfied as well, although (5) is not. The authors report that their method is influenced near singularities, so that the singularity in R*(s) at zero can affect the results for large t.

The results of three experiments are shown in Tables 2.1 - 2.3. Table 2.1 represents the base case with $b = 2$, $\mu = 4$, and the argument t varying from 0.001 to 100. This case is intended to be representative of the typical inputs one might expect, e.g., mean time between failures = 30 minutes, q = 100 parts, p = 100 parts/hour, mean time to repair = 15 minutes. The other two cases are intended to stress the inversion code. In Table 2.2 we set $b = 20$, $\mu = 0.4$ and in Table 2.3 we set $b = 0.2$, $\mu = 40$. For each case, the range of the argument t was selected so that the extremes of the tails of $R(t; T \mid 1)$ were reached.

In these experiments, the exact values reported were obtained via numerical integration of the density r(t; T | 1) using *Mathematica* (Wolfram, 1988) to 14 decimal digits of precision, as reported in tables. In all of the runs, 13 decimal digits of precision were requested of the inversion code. The code used was identical to that used by Murli and Rizzardi in their experiments, except for adjustment of machine dependent parameters, and modifications that we made to utilize double-precision real and complex arithmetic.

We see from Table 2.1 that the algorithm performs extremely well over the entire range of t. Observe that at t=10, the CDF has reached unity to within 12 decimal digits of precision, and at t=1E-14, the inverse transform is **exp**(-b) to precision within the last reported decimal digit. Further, for each value of t, only 25 evaluations of the function R*(s) were required; this metric is sometimes used as an indication of the efficiency of an inversion code.

Table 2.2 shows the results of increasing b to 20 and decreasing $\mu$ to 0.4. We see that for this case, the algorithm performs extremely well for small t, but begins to produce significant errors as t grows large (>50). This is likely due to the singularity in R*(s) at zero. Talbot suggests a simple solution: increase the parameter $\lambda$ in the algorithm. $\lambda$ is a geometric parameter that in part determines the shape of the contour; increasing $\lambda$ will shift the contour of integration away from the singularity. Increasing $\lambda$ carelessly can, according to Murli and Rizzardi, result in a significant increase in computation time.

As an example of this method, we increased $\lambda$ to $2\lambda$ for t = 100. The resulting approximation was then 0.99638150355243, for a relative error of -3.58E-6. Increasing $\lambda$ to $3\lambda$ resulted in an approximation of 0.99638508385793, for a relative error of 1.90E-13.

| t | Approximation | Exact | Rel. error | Pct. error |
|---|---|---|---|---|
| 1E-14 | 0.13533528323662 | 0.13533528323662 | 0 | 0 |
| 1E-13 | 0.13533528323672 | 0.13533528323672 | 0 | 0 |
| 1E-12 | 0.13533528323769 | 0.13533528323770 | -1.00E-14 | -7.38E-12 |
| 1E-11 | 0.13533528324744 | 0.13533528324744 | 0 | 0 |
| 1E-10 | 0.13533528334488 | 0.13533528334488 | 0 | 0 |
| 1E-09 | 0.13533528431929 | 0.13533528431929 | 0 | 0 |
| 1E-08 | 0.13533529406343 | 0.13533529406344 | -1.00E-14 | -7.38E-12 |
| 1E-07 | 0.13533539150484 | 0.13533539150484 | 0 | 0 |
| 1E-06 | 0.13533636591888 | 0.13533636591888 | 0 | 0 |
| 1E-05 | 0.13534611005927 | 0.13534611005927 | 0 | 0 |
| 1E-04 | 0.13544355146224 | 0.13544355146224 | 0 | 0 |
| 1E-03 | 0.13641796454108 | 0.13641796454108 | 0 | 0 |
| 0.01 | 0.14616115308902 | 0.14616115308902 | 0 | 0 |
| 0.02 | 0.15698138198830 | 0.15698138198830 | 0 | 0 |
| 0.03 | 0.16779053519637 | 0.16779053519637 | 0 | 0 |
| 0.04 | 0.17858339797306 | 0.17858339797306 | 0 | 0 |
| 0.05 | 0.18935497016687 | 0.18935497016688 | -1.00E-14 | -5.28E-12 |
| 0.06 | 0.20010046086437 | 0.20010046086437 | 0 | 0 |
| 0.07 | 0.21081528310081 | 0.21081528310081 | 0 | 0 |
| 0.08 | 0.22149504863448 | 0.22149504863448 | 0 | 0 |
| 0.09 | 0.23213556278690 | 0.23213556278691 | -1.00E-14 | -4.30E-12 |
| 0.1 | 0.24273281935103 | 0.24273281935103 | 0 | 0 |
| 0.2 | 0.34556977358660 | 0.34556977358660 | 0 | 0 |
| 0.3 | 0.44088974695003 | 0.44088974695003 | 0 | 0 |
| 0.4 | 0.52708093194976 | 0.52708093194976 | 0 | 0 |
| 0.5 | 0.60350096061199 | 0.60350096061199 | 0 | 0 |
| 0.6 | 0.67017687350024 | 0.67017687350024 | 0 | 0 |
| 0.7 | 0.72757277323626 | 0.72757277323626 | 0 | 0 |
| 0.8 | 0.77641534042106 | 0.77641534042106 | 0 | 0 |
| 0.9 | 0.81756690473868 | 0.81756690473868 | 0 | 0 |
| 1 | 0.85193635694241 | 0.85193635694241 | 0 | 0 |
| 2 | 0.98527653589128 | 0.98527653589128 | 0 | 0 |
| 3 | 0.99888038022251 | 0.99888038022242 | 9.00E-14 | 9.01E-12 |
| 4 | 0.99992806460700 | 0.99992806460307 | 3.93E-12 | 3.93E-10 |
| 5 | 0.99999589578403 | 0.99999589578587 | -1.84E-12 | -1.84E-10 |
| 6 | 0.99999978591957 | 0.99999978591942 | 1.50E-13 | 1.50E-11 |
| 7 | 0.99999998959658 | 0.99999998959718 | -6.00E-13 | -6.00E-11 |
| 8 | 0.99999999952322 | 0.99999999952291 | 3.10E-13 | 3.10E-11 |
| 9 | 0.99999999997913 | 0.99999999997915 | -2.00E-14 | -2.00E-12 |
| 10 | 0.99999999999912 | 0.99999999999913 | -1.00E-14 | -9.99E-13 |

**Table 2.1** Results of Murli and Rizzardi's algorithm for b = 2, $\mu$ = 4

| t | Approximation | Exact | Rel. error | Pct. error |
|---|---|---|---|---|
| 1E-15 | 2.0611536224386E-9 | 2.0611536224386E-9 | 0 | 0 |
| 1E-14 | 2.0611536224387E-9 | 2.0611536224387E-9 | 0 | 0 |
| 1E-13 | 2.0611536224402E-9 | 2.0611536224402E-9 | 0 | 0 |
| 1E-12 | 2.0611536224550E-9 | 2.0611536224550E-9 | 0 | 0 |
| 1E-11 | 2.0611536226034E-9 | 2.0611536226035E-9 | -1.00E-22 | -4.86E-12 |
| 1E-10 | 2.0611536240875E-9 | 2.0611536240875E-9 | 0 | 0 |
| 1E-09 | 2.0611536389278E-9 | 2.0611536389278E-9 | 0 | 0 |
| 1E-08 | 2.0611537873308E-9 | 2.0611537873309E-9 | -1.00E-22 | -4.86E-12 |
| 1E-07 | 2.0611552713617E-9 | 2.0611552713618E-9 | -1.00E-22 | -4.86E-12 |
| 1E-06 | 2.0611701116972E-9 | 2.0611701116972E-9 | 0 | 0 |
| 1E-05 | 2.0613185176964E-9 | 2.0613185176964E-9 | 0 | 0 |
| 1E-04 | 2.0628028421636E-9 | 2.0628028421636E-9 | 0 | 0 |
| 1E-03 | 2.0776725529970E-9 | 2.0776725529970E-9 | 0 | 0 |
| 0.01 | 2.2290350045993E-9 | 2.2290350045993E-9 | 0 | 0 |
| 0.1 | 4.0285688996247E-9 | 4.0285688996247E-9 | 0 | 0 |
| 1 | 7.7718249473702E-8 | 7.7718249473702E-8 | 0 | 0 |
| 2 | 4.8418621380097E-7 | 4.8418621380097E-7 | 0 | 0 |
| 3 | 1.8638678233342E-6 | 1.8638678233342E-6 | 0 | 0 |
| 4 | 5.5448015001308E-6 | 5.5448015001308E-6 | 0 | 0 |
| 5 | 1.3951529953470E-5 | 1.3951529953470E-5 | 0 | 0 |
| 6 | 3.1128461416655E-5 | 3.1128461416655E-5 | 0 | 0 |
| 7 | 6.3366983528633E-5 | 6.3366983528633E-5 | 0 | 0 |
| 8 | 1.1991655923654E-4 | 1.1991655923654E-4 | 0 | 0 |
| 9 | 2.1374957527165E-4 | 2.1374957527165E-4 | 0 | 0 |
| 10 | 3.6234082052527E-4 | 3.6234082052527E-4 | 0 | 0 |
| 20 | 1.3027549946722E-2 | 1.3027549946739E-2 | -1.70E-14 | -1.30E-10 |
| 30 | 9.0853144998461E-2 | 9.0853144737580E-2 | 2.61E-10 | 2.87E-07 |
| 40 | 0.27969010773392 | 0.27969010486059 | 2.87E-09 | 1.03E-06 |
| 50 | 0.53163935704165 | 0.53163913993762 | 2.17E-07 | 4.08E-05 |
| 60 | 0.75157619547114 | 0.75157367485922 | 2.52E-06 | 3.35E-04 |
| 70 | 0.89095222433758 | 0.89095380605233 | -1.58E-06 | -1.78E-04 |
| 80 | 0.95957587129784 | 0.95957851871227 | -2.65E-06 | -2.76E-04 |
| 90 | 0.98709474700440 | 0.98709153853762 | 3.21E-06 | 3.25E-04 |
| 100 | 0.99638377411774 | 0.99638508385774 | -1.31E-06 | -1.31E-04 |
| 110 | 0.99909886512638 | 0.99909869355890 | 1.72E-07 | 1.72E-05 |
| 120 | 0.99979760432365 | 0.99979733975725 | 2.65E-07 | 2.65E-05 |
| 130 | 0.99995828713949 | 0.99995845751716 | -1.70E-07 | -1.70E-05 |
| 140 | 0.99999217241571 | 0.99999216508717 | 7.33E-09 | 7.33E-07 |
| 150 | 0.99999864556966 | 0.99999862976019 | 1.58E-08 | 1.58E-06 |
| 160 | 0.99999977440395 | 0.99999977628191 | -1.88E-09 | -1.88E-07 |
| 170 | 0.99999996487716 | 0.99999996570170 | -8.25E-10 | -8.25E-08 |
| 180 | 0.99999999504371 | 0.99999999503739 | 6.32E-12 | 6.32E-10 |
| 190 | 0.99999999934825 | 0.99999999931933 | 2.89E-11 | 2.89E-09 |
| 200 | 0.99999999991661 | 0.99999999991796 | -1.35E-12 | -1.35E-10 |
| 210 | 0.99999999998919 | 0.99999999998893 | 2.60E-13 | 2.60E-11 |
| 220 | 0.99999999999867 | 0.99999999999821 | 4.60E-13 | 4.60E-11 |
| 230 | 0.99999999999984 | 0.99999999999986 | -2.00E-14 | -2.00E-12 |
| 240 | 0.99999999999999 | 1.00000000000000 | -9.99E-15 | -9.99E-13 |
| 250 | 1.00000000000000 | 1.00000000000000 | 0 | 0 |

**Table 2.2** Results of Murli and Rizzardi's algorithm for b = 20, $\mu$ = 0.4

Furthermore, recomputing the results of all the entries in Table 2.2 resulted in absolute errors smaller than 1.24E-12 for all t reported.

Although we are encouraged by the success of increasing $\lambda$ to $3\lambda$, we should heed the warning of Murli and Rizzardi and first investigate the impact of such a change on the speed of the algorithm. Timing tests were conducted on a Power Macintosh 7100/80 in emulation mode with SANE-based math instructions. The parameters used were the same as for the results of Table 2.2. For the base case ($\lambda$ unadjusted), 1000 inversions required approximately 26 seconds, or 0.26 seconds per inversion. Increasing $\lambda$ to $3\lambda$ resulted in no measurable increase in computation time. Lastly, an unrelated test was conducted to determine the impact of decreasing the requested accuracy from 13 significant decimal digits to 6. The result was a two-fold performance improvement, decreasing the time per inversion to approximately 0.13 seconds.

The results of the third experiment are reported in Table 2.3. Here we see that for b = 0.2, $\mu$ = 40, $\lambda$ unadjusted, the algorithm once again performs extremely well. At t = 1E-15, the inverse transform is equal to **exp**(-b) and at t = 0.9 the inverse transform equals unity, to accuracy within the last decimal digit reported.

Further experiments were conducted and showed that an increase in $\lambda$ was helpful to improve accuracy whenever b was several (e.g., 3) orders of magnitude larger than $\mu$.

Experiments were also performed at b = 20, $\mu$ = 40, and the results were identical to those obtained at b = 20, $\mu$ = 0.4, except with t 100 times smaller. Similarly, the results for b = 0.2, $\mu$ = 0.4 were identical to those obtained for b = 0.2, $\mu$ = 40, except with t 100 times larger.

**Weeks' Method**

Garbow et al. (1988) have developed an implementation of Weeks' method, which approximates a function from its Laplace transform by expansion in a Laguerre polynomial. The method requires that the function has continuous derivatives of all orders.

The algorithm consists of two distinct stages. The first stage computes the coefficients of the Laguerre polynomial to achieve a desired accuracy level. Once the coefficients are determined, the inversion for any particular value of the function is accomplished simply by evaluating the Laguerre polynomial. This two stage approach means that the algorithm of Garbow et al. will be particularly efficient when the same function needs to be evaluated for many different values of t.

Timing experiments were performed on the same hardware as before, on the problem of Table 2.1. Like many numerical inversion algorithms, geometric parameters can be specified which can influence the accuracy of the result. In our experiments we set $\sigma_0$ to one and allowed the algorithm to set $\sigma$ and b; see Lyness and Giunta (1986) for theoretical details. The first stage of the algorithm computed a Laguerre polynomial with 128 coefficients in 0.14 seconds. The algorithm also determined that accuracy could not be improved further (i.e., by computing more coefficients). Using this polynomial, 1000 inversions were performed in 8 seconds, which is about 0.008 seconds per inversion.

For this particular function and choice of $\lambda$ and $\mu$, we see that the algorithm of Garbow et al. is faster than that of Murli and Rizzardi *even if only one function value is required.* This need not always be true. We have found that some Laguerre polynomial expansions require a much larger number of coefficients to achieve a high degree of

| t | Approximation | Exact | Rel. error | Pct. error |
|---|---|---|---|---|
| 1E-15 | 0.81873075307798 | 0.81873075307799 | -1E-14 | -1E-12 |
| 1E-14 | 0.81873075307804 | 0.81873075307805 | -1E-14 | -1E-12 |
| 1E-13 | 0.81873075307863 | 0.81873075307864 | -1E-14 | -1E-12 |
| 1E-12 | 0.81873075308453 | 0.81873075308453 | 0 | 0 |
| 1E-11 | 0.81873075314348 | 0.81873075314348 | 0 | 0 |
| 1E-10 | 0.81873075373296 | 0.81873075373297 | -1E-14 | -1E-12 |
| 1E-09 | 0.81873075962782 | 0.81873075962783 | -1E-14 | -1E-12 |
| 1E-08 | 0.81873081857643 | 0.81873081857643 | 0 | 0 |
| 1E-07 | 0.81873140806140 | 0.81873140806141 | -1E-14 | -1E-12 |
| 1E-06 | 0.81873730280611 | 0.81873730280611 | 0 | 0 |
| 1E-05 | 0.81879623974991 | 0.81879623974991 | 0 | 0 |
| 1E-04 | 0.81938456011584 | 0.81938456011585 | -1E-14 | -1E-12 |
| 1E-03 | 0.82516409833765 | 0.82516409833766 | -1E-14 | -1E-12 |
| 0.01 | 0.87373116093797 | 0.87373116093798 | -1E-14 | -1E-12 |
| 0.02 | 0.91208423778107 | 0.91208423778107 | 0 | 0 |
| 0.03 | 0.93881527657833 | 0.93881527657834 | -1E-14 | -1E-12 |
| 0.04 | 0.95743704762819 | 0.95743704762819 | 0 | 0 |
| 0.05 | 0.97040356552837 | 0.97040356552838 | -1E-14 | -1E-12 |
| 0.06 | 0.97942821305528 | 0.97942821305529 | -1E-14 | -1E-12 |
| 0.07 | 0.98570660604161 | 0.98570660604161 | 0 | 0 |
| 0.08 | 0.99007261462709 | 0.99007261462710 | -1E-14 | -1E-12 |
| 0.09 | 0.99310751643234 | 0.99310751643235 | -1E-14 | -1E-12 |
| 0.1 | 0.99521631196769 | 0.99521631196769 | 0 | 0 |
| 0.2 | 0.99987807193596 | 0.99987807193596 | 0 | 0 |
| 0.3 | 0.99999697260719 | 0.99999697260720 | -1E-14 | -1E-12 |
| 0.4 | 0.99999992636591 | 0.99999992636588 | 3E-14 | 3E-12 |
| 0.5 | 0.99999999823878 | 0.99999999823881 | -3E-14 | -3E-12 |
| 0.6 | 0.99999999995846 | 0.99999999995846 | 0 | 0 |
| 0.7 | 0.99999999999903 | 0.99999999999903 | 0 | 0 |
| 0.8 | 0.99999999999997 | 0.99999999999998 | -1E-14 | -1E-12 |
| 0.9 | 1.00000000000000 | 1.00000000000000 | 0 | 0 |

**Table 2.3** Results of Murli and Rizzardi's algorithm for b = 0.2, $\mu$ = 40

accuracy, and in these cases the first stage of the algorithm can take many times longer than in the present case. Of course, if a sufficiently large number of function values is desired, then the algorithm of Garbow et al. will always be faster.

A naive attempt to implement the algorithm of Garbow et al. for the problem of Table 2.2 (b = 20, $\mu$ = 0.4) produces unacceptable results. The algorithm performed extremely well for small to moderate values of t, but for $t \geq 40$ the algorithm did not output meaningful answers. The problem is easily corrected by scaling the problem. In

particular, we multiplied $\mu$ by b and the argument t by 1/b. This resulted in less accuracy for small t but uniformly good results over the entire range of t. The results are summarized in Table 2.4. To achieve even better results, one could employ a combination of a scaled and unscaled usage of the algorithm depending on parameter and argument values.

Timing experiments were also conducted for the problem of Table 2.4. The first stage of the algorithm computed a Laguerre polynomial with 256 coefficients in 0.24 seconds, and determined that accuracy could not be improved further by computing more coefficients. Using this polynomial, 1000 inversions were performed in 15 seconds, which is about 0.015 seconds per inversion.

| t | Approximation | Exact | Rel. error | Pct. error |
|---|---|---|---|---|
| 1E-15 | 2.0611536183944E-9 | 2.0611536224386E-9 | -4E-18 | -2E-07 |
| 1E-14 | 2.0611536197266E-9 | 2.0611536224387E-9 | -3E-18 | -1E-07 |
| 1E-13 | 2.0611536188208E-9 | 2.0611536224402E-9 | -4E-18 | -2E-07 |
| 1E-12 | 2.0611536186600E-9 | 2.0611536224550E-9 | -4E-18 | -2E-07 |
| 1E-11 | 2.0611536184981E-9 | 2.0611536226035E-9 | -4E-18 | -2E-07 |
| 1E-10 | 2.0611536208407E-9 | 2.0611536240875E-9 | -3E-18 | -2E-07 |
| 1E-09 | 2.0611536347846E-9 | 2.0611536389278E-9 | -4E-18 | -2E-07 |
| 1E-08 | 2.0611537845659E-9 | 2.0611537873309E-9 | -3E-18 | -1E-07 |
| 1E-07 | 2.0611552677738E-9 | 2.0611552713618E-9 | -4E-18 | -2E-07 |
| 1E-06 | 2.0611701080297E-9 | 2.0611701116972E-9 | -4E-18 | -2E-07 |
| 1E-05 | 2.0613185140643E-9 | 2.0613185176964E-9 | -4E-18 | -2E-07 |
| 1E-04 | 2.0628028388098E-9 | 2.0628028421636E-9 | -3E-18 | -2E-07 |
| 1E-03 | 2.0776725493649E-9 | 2.0776725529970E-9 | -4E-18 | -2E-07 |
| 0.01 | 2.2290350018216E-9 | 2.2290350045993E-9 | -3E-18 | -1E-07 |
| 0.1 | 4.0285688983664E-9 | 4.0285688996247E-9 | -1E-18 | -3E-08 |
| 1 | 7.7718249472495E-8 | 7.7718249473702E-8 | -1E-18 | -2E-09 |
| 2 | 4.8418621379980E-7 | 4.8418621380097E-7 | -1E-18 | -2E-10 |
| 3 | 1.8638678233330E-6 | 1.8638678233342E-6 | -1E-18 | -6E-11 |
| 4 | 5.5448015001291E-6 | 5.5448015001308E-6 | -2E-18 | -3E-11 |
| 5 | 1.3951529953469E-5 | 1.3951529953470E-5 | -1E-18 | -7E-12 |
| 6 | 3.1128461416655E-5 | 3.1128461416655E-5 | 0 | 0 |
| 7 | 6.3366983528632E-5 | 6.3366983528633E-5 | -1E-18 | -2E-12 |
| 8 | 1.1991655923653E-4 | 1.1991655923654E-4 | -1E-17 | -8E-12 |
| 9 | 2.1374957527165E-4 | 2.1374957527165E-4 | 0 | 0 |
| 10 | 3.6234082052526E-4 | 3.6234082052527E-4 | -1E-17 | -3E-12 |
| 20 | 1.3027549946739E-2 | 1.3027549946739E-2 | 0 | 0 |
| 30 | 9.0853144737558E-2 | 9.0853144737580E-2 | -2E-14 | -2E-11 |
| 40 | 0.27969010486059 | 0.27969010486059 | 0 | 0 |
| 50 | 0.53163913993762 | 0.53163913993762 | 0 | 0 |
| 60 | 0.75157367485920 | 0.75157367485922 | -2E-14 | -3E-12 |
| 70 | 0.89095380605233 | 0.89095380605233 | 0 | 0 |
| 80 | 0.95957851871226 | 0.95957851871227 | -1E-14 | -1E-12 |
| 90 | 0.98709153853762 | 0.98709153853762 | 0 | 0 |
| 100 | 0.99638508385774 | 0.99638508385774 | 0 | 0 |
| 110 | 0.99909869355937 | 0.99909869355890 | 5E-13 | 5E-11 |
| 120 | 0.99979733975724 | 0.99979733975725 | -1E-14 | -1E-12 |
| 130 | 0.99995845751714 | 0.99995845751716 | -2E-14 | -2E-12 |
| 140 | 0.99999216508717 | 0.99999216508717 | 0 | 0 |
| 150 | 0.99999862976019 | 0.99999862976019 | 0 | 0 |
| 160 | 0.99999977628198 | 0.99999977628191 | 7E-14 | 7E-12 |
| 170 | 0.99999996570231 | 0.99999996570170 | 6E-13 | 6E-11 |
| 180 | 0.99999999503693 | 0.99999999503739 | -5E-13 | -5E-11 |
| 190 | 0.99999999931367 | 0.99999999931933 | -6E-12 | -6E-10 |
| 200 | 0.99999999990377 | 0.99999999991796 | -1E-11 | -1E-09 |
| 210 | 1.00000000000000 | 0.99999999998893 | 1E-11 | 1E-09 |
| 220 | 0.99999999999743 | 0.99999999999821 | -8E-13 | -8E-11 |
| 230 | 0.99999999998623 | 0.99999999999986 | -1E-11 | -1E-09 |
| 240 | 0.99999999955673 | 1.00000000000000 | -4E-10 | -4E-08 |
| 250 | 0.99999999932218 | 1.00000000000000 | -7E-10 | -7E-08 |

**Table 2.4** Results of the algorithm of Garbow et al. for b = 20, $\mu$ = 0.4

## References for Chapter 2


Abate, Joseph and Ward Whitt. "Numerical Inversion of Probability Generating Functions". Operations Research Letters, 12, pp. 245-251, 1992.

Abramowitz, Milton and Irene A. Stegun. Handbook of Mathematical Functions, Applied Mathematics Series, vol. 55. Washington: National Bureau of Standards, 1964 (reprinted by Dover Publications, New York, 1965).

Barlow, Richard E. and Larry C. Hunter. "Reliability Analysis of a One-Unit System". Operations Research, 9, pp. 200-208, 1961.

Barlow, Richard E. and Frank Proschan. Mathematical Theory of Reliability. New York: John Wiley & Sons, Inc., 1965.

Baxter, Laurence A. "Availability Measures for a Two-State System". Journal of Applied Probability, 18, pp. 227-235, 1981a.

Baxter, Laurence A. "Some Remarks on Numerical Convolution". Communications in Statistics: Simulation and Computation, B10(3), pp. 281-288, 1981b.

Baxter, Laurence A. "Some Notes on Availability Theory". Microelectronics and Reliability, 25(5), pp. 921-926, 1985.

Bharucha-Reid, A. T. Elements of the Theory of Markov Processes and Their Applications. New York: McGraw-Hill, 1960.

Boisvert, Ronald F. and Bonita V. Saunders. "Algorithm 713: Portable Vectorized Software for Bessel Function Evaluation". ACM Transactions on Mathematical Software, 18(4), pp. 456-469, December 1992.

Brouwers, J. J. H. "Probabilistic Descriptions of Irregular System Downtime". Reliability Engineering, 15, pp. 263-281, 1986.

Cléroux, R. and D. J. McConalogue. "A Numerical Algorithm for Recursively-Defined Convolution Integrals Involving Distribution Functions." Management Science, 22, pp. 1138-1146, 1976.

Cody, W. J. "Algorithm 597: Sequence of Modified Bessel Functions of the First Kind." ACM Transactions on Mathematical Software, 9(2), pp. 242-245, June 1983.

Copson, E. T. An Introduction to the Theory of Functions of a Complex Variable. Oxford: Oxford University Press, 1935.

Cox, D. R. Renewal Theory. London: Methuen & Co., Ltd., 1962.

Crump, Kenny S. "Numerical Inversion of Laplace Transforms Using a Fourier Series Approximation". Journal of the Association for Computing Machinery, 23(1), pp. 89-96, 1976.

Davies, Brian and Brian Martin. "Numerical Inversion of the Laplace Transform: a Survey and Comparison of Methods". Journal of Computational Physics, 33(1), pp. 1-32, 1979.

De Hoog, F. R., J. H. Knight, and A. N. Stokes. "An Improved Method for Numerical Inversion of Laplace Transforms". SIAM Journal of Scientific and Statistical Computing, 3(3), pp. 357-366, 1982.

Dickey, J. M. "The Renewal Function for an Alternating Renewal Process, Which Has a Weibull Failure Distribution and a Constant Repair Time". Reliability Engineering and System Safety, 31, pp. 321-343, 1991.

Doetsch, G. Guide to the Application of Laplace Transforms. New York: Van Nostrand, 1961.

Dongarra, J. J. and E. Grosse. "Distribution of Mathematical Software via Electronic Mail". Communications of the ACM, 30(5), pp. 403-407, May 1987.

Feller, William. An Introduction to Probability Theory and Its Applications, Volume 2, Second Edition. New York: John Wiley & Sons, Inc., 1971.

FitzHugh, Richard. "Statistical Properties of the Asymmetric Random Telegraph Signal, with Applications to Single-Channel Analysis". Mathematical Biosciences, 64, pp. 75-89, 1983.

Garbow, B. S., G. Giunta, J. N. Lyness and A. Murli. "Software for an Implementation of Weeks' Method for the Inverse Laplace Transform Problem". ACM Transactions on Mathematical Software, 14(2), pp. 163-170, June 1988.

Gershwin, Stanley B. Manufacturing Systems Engineering. Englewood Cliffs, NJ: Prentice-Hall, 1994.

Gnedenko, B. V., Yu K. Belyayev and A. D. Solovyev. Mathematical Methods of Reliability Theory. New York: Academic Press, 1969.

Gross, Donald and Carl M. Harris. Fundamentals of Queueing Theory. New York: John Wiley & Sons, Inc., 1985.

Grundy, R. E. "Laplace Transform Inversion Using Two-point Rational Approximants". Journal of the Institute of Mathematics and Its Applications, 20, pp. 299-306, 1977.

Kabak, Irwin W. "System Availability and Some Design Implications". Operations Research, 17, pp. 827-837, 1969.

Kalashnikov, Vladimir V. Mathematical Methods in Queueing Theory. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1994.

Keilson, Julian. Markov Chain Models – Rarity and Exponentiality. New York: Springer-Verlag, 1979.

Kim, David S. Personal communication, 1994.

Kim, David S. and Jeffrey M. Alden. "Estimating the Distribution of Time Required to Produce a Fixed Lot Size on a Serial Production Line". Research Report GMR-7811, General Motors Research Laboratories, 1992.

Knopp, Konrad. Infinite Sequences and Series. New York: Dover, 1956.

Krylov, Vladimir Ivanovich and Nadezhda Sergeevna Skoblya. Handbook of Numerical Inversion of Laplace Transforms. Jerusalem: Israel Program for Scientific Translations, 1969.

Lie, C. H., C. L. Hwang, and F. A. Tillman. "Availability of Maintained Systems: A State-of-the-Art Survey". AIIE Transactions, 9(3), pp. 247-259, 1977.

Lyness, J. N. and G. Giunta. "A Modification of the Weeks Method for Numerical Inversion of the Laplace Transform". Mathematics of Computation, 47(175), pp. 313-322, July 1986.

Martz, H. F. Jr. "On Single-Cycle Availability". IEEE Transactions on Reliability, R-20(1), pp. 21-23, 1971.

McConalogue, D. J. "Convolution Integrals Involving Probability Distribution Functions (Algorithm 102)". Computer Journal, 21(3), pp. 270-272, 1978.

McConalogue, D. J. "Numerical Treatment of Convolution Integrals Involving Distributions With Densities Having Singularities at the Origin". Communications in Statistics: Simulation and Computation, B10(3), pp. 265-280, 1981.

Morse, Philip M. Queues, Inventories and Maintenance. New York: John Wiley & Sons, Inc., 1958.

Mortensen, R. E. "Alternating Renewal Process Models for Electric Power System Loads". IEEE Transactions on Automatic Control, 35(11), pp. 1245-1249, 1990.

Munford, A. G. "Moments of a Filtered Binary Process". IEEE Transactions on Information Theory, 32(6), pp. 824-826, 1986.

Murli, A. and M. Rizzardi. "Algorithm 682: Talbot's Method for the Laplace Inversion Problem". ACM Transactions on Mathematical Software, 16(2), pp. 158-168, June 1990.

Nahmias, Stephen. Production and Operations Analysis. Homewood, Illinois: Irwin, 1989.

Piessens, R. and R. Huysmans. "Algorithm 619: Automatic Numerical Inversion of the Laplace Transform". ACM Transactions on Mathematical Software, 10(3), pp. 348-353, September 1984.

Press, William H., Brian P. Flannery, Saul A. Teukolsky and William T. Vetterling. Numerical Recipes in Pascal. Cambridge: Cambridge University Press, 1989.

Ross, Sheldon M. Stochastic Processes. New York: John Wiley & Sons, Inc., 1983.

Ross, Sheldon M. Introduction to Probability Models, Fourth Edition. San Diego, CA: Academic Press, Inc., 1989.

Serfozo, R. F. "Point Processes", Chapter 1 in Heyman, D. P. and M. J. Sobel, eds., Stochastic Models, Handbooks in Operations Research & Management Science, Vol. 2, Amsterdam: Elsevier (North-Holland), 1990.

Sericola, Bruce. "Closed-form Solution for the Distribution of the Total Time Spent in a Subset of States of a Homogeneous Markov Process During a Finite Observation Period". Journal of Applied Probability, 27, pp. 713-719, 1990.

Shaked, Moshe and J. George Shanthikumar. "Reliability and Maintainability", Chapter 13 in Heyman, D. P. and M. J. Sobel, eds., Stochastic Models, Handbooks in Operations Research & Management Science, Vol. 2, Amsterdam: Elsevier (North-Holland), 1990.

Sookne, D. J. "Bessel Functions of Real Argument and Integer Order". National Bureau of Standards Journal of Research B, 77A, pp. 125-132, 1973.

Takács, Lajos. "Occurrence and Coincidence Phenomena in Case of Happenings with Arbitrary Distribution Law of Duration". *Acta Mathematica* (Academiae Scientiarum Hungaricae), 2, pp. 275-298, 1951.

Takács, Lajos. "On Certain Sojurn Time Problems in the Theory of Stochastic Processes". *Acta Mathematica* (Academiae Scientiarum Hungaricae), 8, pp. 169-191, 1957a.

Takács, Lajos. "On Limiting Distributions Concerning a Sojourn Time Problem". *Acta Mathematica* (Academiae Scientiarum Hungaricae), 8, pp. 279-294, 1957b.

Takács, Lajos. "On a Sojurn Time Problem in the Theory of Stochastic Processes". Transactions of the American Mathematical Society, 93, pp. 531-540, 1959.

Talbot, A. "The Accurate Numerical Inversion of Laplace Transforms". Journal of the Institute of Mathematics and Its Applications, 23, pp. 97-120, 1979.

Wolfram, Stephen. *Mathematica*: A System for Doing Mathematics by Computer. Redwood City, CA: Addison-Wesley, 1988.

# 3. Dynamic overtime decision model

## Introduction

In this chapter we develop models to evaluate a production plan for an unreliable machine, and determine when it is cost optimal to run overtime. To motivate this discussion and place these models in context, let us consider the following (intentionally oversimplified) example. Suppose we must deliver 500 units of our product to our customer tomorrow morning, but we only have 400 units in inventory, so we must manufacture 100 units. It is now 4:00 PM; there is one hour left in the work day; our machine is currently set up to produce this product; and the machine can produce 200 units per hour. Unfortunately, the machine fails on average every 30 minutes and when it fails, requires 15 minutes on average to fix. If the machine did not fail, we could produce the 100 units in half an hour. However, due to machine failures, there is some probability that we will not be able to produce the 100 units by the deadline.

The production manager is now faced with several questions. What is the probability that we will be able to meet our demand? What is the expected shortfall? At 4:01 PM, the machine fails. Now what is the probability that we will be able to meet our demand? The production manager now considers using overtime. Suppose union rules dictate that plant management must decide by 4:30 PM if overtime will be run for one hour at a cost of $200. What should the decision be? Suppose instead the amount of overtime can be chosen, up to 4 hours. How much should be chosen? Suppose instead that after running one hour of overtime, the production manager can stop overtime at any point. When should we stop? Suppose that we can delay shipping the product until noon if we pay $5 per unit extra for express freight shipping. How does this change the decision?

Next, consider a case where the plant manufactures two products. Suppose we are manufacturing product #1 and have just produced enough units to satisfy our demand for tomorrow. Our production schedule, however, dictates that we continue manufacturing product #1 for another 200 units, at which point we are scheduled to perform a changeover to begin manufacturing product #2. Due to random machine failures, we do not know what time we will finish producing product #1, but we *expect* to begin producing product #2 around 4:00 PM. As before, this would leave us one hour to manufacture 100 units that we must ship by tomorrow morning.

The production manager is now faced with an even more difficult set of questions. Now what is the probability that we will be able to meet our demand, and what is the expected shortfall? Should we stop production of product #1 before we build the full economic lot size? How would that impact our ability to meet our next shipping deadline?

This chapter will develop a series of models that will assist a decision maker in answering questions such as the ones posed above. These models could be used as part of a manufacturing control system in a real manufacturing operation. One can (and should) envision these models embedded in a software tool that would receive data in real-time from the shop floor and assist plant management in decision making.

**Literature review**

In this subsection we briefly review the literature that is related to the problem of deciding when to run overtime on an unreliable machine. Although no paper

addresses this specifically, many papers have addressed some aspect of this problem. We will divide our literature review into two parts: those that incorporate overtime opportunities, and those that model an unreliable machine. Our intent is not to cite every paper that has been written on these subjects, but rather to give the reader a sense for the types of models that have been studied by others. The interested reader is also referred to the literature review in Chapter 1.

Unreliable machine

The presence of machine unreliability in a manufacturing system has been studied in a variety of different contexts, including problems of sequencing, scheduling, and lot sizing. We briefly review each of these areas.

There has been some limited work on sequencing of jobs on an unreliable machine. The earliest work is that of Glazebrook (1984), who models the problem as a rather general cost-discounted Markov decision process. He shows the conditions under which the optimal policy is of an index type (i.e., the job to be processed is the one with the smallest *Gittins index;* see Gittins, 1979). Pinedo and Rammouz (1988) find the optimal non-preemptive policies for several objective functions in the case of a Poisson failure process. For a general failure process and a discrete time model, Birge and Glazebrook (1988) find bounds on the error of following the strategy that is optimal when the failure process is memoryless. Birge et al. (1990) study in greater detail the problem of minimizing weighted flow-time and obtain results that are consistent with and complementary to Pinedo and Rammouz. For a detailed and current overview of this research area, see Pinedo (1995).

There has also been some work on lot sizing on an unreliable machine. Groenevelt et al. (1992a, 1992b) extend the basic economic manufacturing quantity (EMQ) model

to incorporate the effects of machine breakdowns. The first paper assumes that repairs are instantaneous but bear a fixed cost. The second paper assumes (as we do) that repairs are not instantaneous but instead consume machine time. This model permits any repair time distribution, but assumes that the time between failures is exponentially distributed. Under the assumption of lost sales, the authors seek an optimal lot size and safety stock level to minimize cost subject to a constraint on the service level. They require, however, some awkward assumptions regarding safety stock to achieve separability in the optimization of the lot sizes and safety stock level. The authors do not explore the impacts of multiple parts sharing the same machine.

Other authors, such as Sethi and Zhang (1994) have approached the problem from a control theoretic perspective. These authors consider the problem of finding an optimal setup schedule (a sequence of parts and the times at which the changeovers will occur) for an unreliable machine. They show that in the limit (as the length of the horizon tends to infinity), the stochastic problem can be reduced to a deterministic problem, and show how to obtain the optimal control policy. The authors also cite many other similar works.

Reiman and Wein (1994) study a two customer class, single server system with setups. The authors use heavy traffic diffusion approximations to analyze a system with a renewal arrival process, general service times, and either setup costs or setup times. They solve a control problem to minimize a linear function of the queue length plus setup costs, if any. Within these heavy traffic diffusion approximations one could model the unreliability of the machine within the service time distribution.

Overtime opportunities

There has been little work on modeling manufacturing systems where overtime opportunities exist. The research that we have found is quite different from the problem context presented here. Some of these models have treated overtime decisions as a tactical planning problem. For example, Gelders and Kleindorfer (1974, 1975) present a coordinated planning and scheduling model for a one-machine job shop with overtime opportunities. The planning problem is to determine overtime usage levels in each period over a finite horizon, where costs can be time varying. The scheduling model determines job release dates to minimize tardiness plus flow time costs. The authors present a branch and bound scheme and discuss many properties of the optimal solution.

In the area of scheduling, Matsuo (1988) has studied the problem of job sequencing on a single machine to minimize weighted total tardiness plus overtime costs. The author presents an approximate algorithm based on solving a capacitated transportation problem.

Adshead and Price (1989) investigate, via simulation, the impact of different overtime adjustment rates and rules for determining the amount of overtime and where to use it in a make-to-stock shop. They treat the shop as deterministic and stationary, with the exception of the demand pattern, which they obtained from real, non-stationary data. These authors find little value in frequently adjusting overtime levels, which is not surprising in light of the deterministic assumptions they have made.

Many authors have studied queueing systems in which the server is not always available, perhaps due to machines failures or overtime (or lack thereof). These models could be used to analyze a make-to-order system in which jobs arrive to the system from the outside. Federgruen and Green (1986, 1989) and Sengupta (1990) present a general model for a single machine and develop bounds and approximations for typical performance measures. Sengupta also gives exact results for the case of exponentially distributed off times. Bitran and Tirupati (1991) study an open network of queues with fixed overtime opportunities. Based on their earlier works, the authors develop an approximation for the work-in-process levels (queue lengths) at each work center.

**Overview of this chapter**

In the next section we describe many of our assumptions and introduce much of the notation that we will use throughout the chapter. In Section 3.2 we show how to evaluate the expected cost of a given production plan. Section 3.3 describes how to formulate a dynamic program that extends the model of Section 3.2 to include a simple overtime decision problem. This model forms the basic building block that we extend and explore in subsequent sections. We describe the computational complexity of the algorithm and then exercise the model under a variety of scenarios and show its behavior under a variety of scenarios.

We then characterize the structure of the costs and optimal solution of the model and discuss a computational issue associated with this model in Sections 3.4 and 3.5. These sections can be omitted by the reader without loss of continuity.

In Section 3.6 we consider static optimal solutions, that is, the optimal solution where all decisions must be made at time zero and cannot be changed over the

horizon. We begin by showing how to find the static optimal solution by numerical integration or by numerical Laplace transform inversion. We then describe an approach that can more quickly identify the optimal solution in certain circumstances. Lastly, the cost of the dynamic solution is compared to the cost of the static optimal solution. We show that even under moderate uncertainty and a short horizon, there can be significant benefits to dynamic optimization.

In Section 3.7 we consider a variety of extensions to the basic model of Section 3.3. The first extension we consider is early overtime authorization. In some situations it is necessary to decide whether or not to run overtime earlier than the point at which overtime actually begins. We show how to accommodate this situation. In some cases, a simple revision of the inputs to the model is all that is required. At worst, a minor modification to the algorithm is required.

Previous sections assumed that the overtime opportunities are of fixed length. We consider two extensions that relax this restriction. The first extension allows overtime to be consumed in a series of discrete blocks. After a block of overtime is purchased, the overtime is performed and the resulting state of the system is observed before a decision must be made whether or not to purchase additional overtime. We show how, by adding additional stages, the dynamic programming algorithm can be used to incorporate this extension, provided that the overtime costs are convex and increasing as more overtime is consumed. The second extension allows choosing among a set of possible overtime opportunities of varying lengths. This corresponds to the situation in which the amount of overtime must be chosen before any overtime is begun. This second extension does not have a convexity restriction on the overtime costs. We first describe how the solution of the dynamic program provides us with information to easily evaluate

an overtime opportunity at time zero of variable size. We then show how to modify the dynamic programming algorithm to accommodate the case where there is a set of possible overtime opportunities of varying lengths in the middle of the horizon.

The last extension that we consider in Section 3.7 is a constraint on the number of overtime opportunities used over the horizon. We show how to modify the dynamic programming algorithm without a large increase in computation time. With limited additional computational effort, the resulting dynamic programming solution can also provide information about reductions in the number of opportunities available. We also show that without additional computational burden we can accommodate more elaborate constraint structures, such as a constraint on the number of overtime opportunities used over the first half of the horizon, and a separate constraint on the number of overtime opportunities used in the second half. Lastly we describe how, by similar methods, to incorporate a constraint on the quantity of overtime used (e.g., no more than eight hours per week). These extensions are not computationally burdensome.

In Section 3.8 we examine the impact of the finite horizon assumption that we have made in the preceding sections. First, we show empirically how the optimal decisions are affected by increasing the length of the horizon, and the factors that influence the rate at which the steady state is attained.

In Section 3.9 we explore certain types of rescheduling and sensitivity analysis. We begin by describing how to compute the marginal benefit of shifting production between two scheduled production batches. This information can help decide when it makes sense to "cut short", i.e., shift some of today's workload to a future time,

and when to "get ahead". This is essentially sensitivity analysis on the production quantities. We show how this information can be used to estimate the shadow prices of the lengths of the overtime opportunities. We describe how to compute these marginal benefits with minimal computational effort if the machine reliability is the same across all parts. Lastly, we show that with minimal computational effort we can compute the sensitivity of the total cost to the demand quantities and to the overtime and shortage costs, irrespective of whether or not machine reliability is part dependent.

In Section 3.10 we attempt to make some progress toward modeling overtime decisions when demand is stochastic. We consider two special cases. The first incorporates stochastic demands in the special case where only one part is produced on the machine. The second special case we consider assumes that the demand for all parts occurs at the same point in time, there is only one such point over the horizon, and the uncertainty in the demand quantity is not revealed until the last moment. We show that this is essentially a multi-item newsvendor problem where the amount that can be ordered is constrained (due to available machine time), and the amount that is received is uncertain (due to machine unreliability). Given a production sequence, we show how to numerically find cost minimizing production quantities. We then show how to dynamically update this strategy based on the realized output of the machine. In particular, for any point in time we show how to find a critical inventory level, above which the production of the current part should be stopped so that production of the next part in sequence can begin. Lastly, we show how to determine a cost minimizing overtime decision.

## 3.1 Problem statement and notation

In this chapter we will focus on a single machine that repetitively produces a set of parts. We will only consider cases in which batching is necessary on the machine, presumably because setups consume precious machine time, are expensive, or both. We will assume that this machine is unreliable, and further, that breakdown is the only source of uncertainty over a short horizon. We will consider a finite horizon and assume that the time and quantity of demand is known over this horizon.

The models described in this chapter will assume that a production schedule (described below) is given as input. In the next section we will describe how to evaluate the expected shortfall cost of a given production schedule. We will then expand this discussion in Section 3.3 to include options to run overtime, and describe how to determine when it is optimal to run overtime to minimize the expected overtime and shortfall costs.

Each of these assumptions was discussed with various individuals responsible for production planning and scheduling at a General Motors metal stamping plant. The overall conclusion was that these assumptions were reasonably consistent with their manufacturing system. First, each metal part is usually assigned to a single machine on which it will be produced, as machines are different and the dies and automation used to produce the parts are tailored to a specific machine. A single machine might be assigned as few as two or as many as twenty different metal parts. Between production runs, the machine must be stopped and a specialized changeover crew must set up the machine to produce the next part. Thus, changeovers are both costly and consume machine time. Some of the machines do fail quite often (many times per day) and incur highly variable repair times (a few

seconds to a few hours). Lastly, the requirements on the machine are often known with a reasonably high degree of certainty over a period of two weeks, during which each part would certainly be produced at least once. The schedulers conveyed that within a two week period, machine unreliability was the greatest source of disruption to the schedule, and that schedule disruptions were a common occurrence. For a more detailed description of a stamping line, see Kletter (1994). For a good overview of a real automobile stamping plant, see Brooke (1993).

**Notation and assumptions**

In this chapter we will focus on a single machine that produces a set of parts indexed by $k = 1, \ldots, K$. When the machine is working it produces parts at a deterministic rate, but is subject to random failures and random repair times. We assume that the failure times and repair times are each i.i.d. exponential random variables. We assume operation dependent failures, i.e., the machine can not fail while it is under repair, nor can it fail when it is not working or in changeover.

Our model will consider decisions over a finite horizon. The model takes as input a plan for production over this horizon. There are two parts to the production plan. The first is a production sequence that defines the number of production runs (and therefore, the number of changeovers) and which part will be produced during each production run. Note that because the machine is unreliable, the time at which each production run begins is not known in advance. The second half of the production plan is the quantity that is planned for each production run. Without loss of generality we assume that the production runs are indexed by $i = 1, \ldots, N$ in the order that they are planned. Changeover times between production runs can be sequence dependent but are assumed to be deterministic. Within these changeover times we can include the time for any planned maintenance.

Based on the above assumptions, let us define the following *inputs* to the model

$P_i$ = production rate during the $i^{th}$ production run,

$\lambda_i$ = failure rate during the $i^{th}$ production run,

$\mu_i$ = repair rate during the $i^{th}$ production run,

$S_i$ = changeover time required to begin the $i^{th}$ production run,

$Q_i$ = planned production quantity for the $i^{th}$ production run, in parts,

$IK_i$ = part to be produced during the $i^{th}$ production run.

Logically, we would expect that if $IK_a = IK_b$, then $P_a = P_b$, $\lambda_a = \lambda_b$ and $\mu_a = \mu_b$, although there is nothing in the model that requires this to be so. It will be assumed throughout that all times and rates are expressed in a common time unit.

We assume that demand for each part is known with certainty over the horizon, and that all of the demands occur at known points in time. Without loss of generality, index the demand points by j = 1, ..., M in the order in which they occur. Let

$JK_j$ = part demanded at the $j^{th}$ demand point,

$D_j$ = cumulative number of parts of type $JK_j$ demand at the $j^{th}$ demand point,

$TD_j$ = time of the $j^{th}$ demand point.

To ensure that our definition of $D_j$ is clear, let us consider an example. Suppose the first four demand points are for parts 1, 2, 1, and 2, respectively, for quantities of 15, 7, 2, and 3. Then $D_1 = 15$, $D_2 = 7$, $D_3 = 17$, and $D_4 = 10$.

If there are not enough parts in inventory of type $JK_j$ by the deadline $TD_j$, there is a stockout charge $cs_j$ per unit not filled. The stockout charge is a one time penalty and therefore is not a function of the length of time that the unfilled demand is outstanding. The planned production quantities are not affected when stockouts occur; we assume that backordered demand must still be satisfied. These assumptions would be appropriate in a remote metal stamping plant, for example, where all demand must be filled, so extra freight costs must be paid for express shipment whenever a shipping deadline is missed, so that the shipment will arrive on time. At this point we do not assume any relationship between the production plan and the demand requirements.

We require that at any point in time, the current state of the system is known: current inventory levels and the machine state are assumed to be given as inputs. Accordingly, define

$I_k(t)$ = inventory of part k at time t,

$\alpha(t) = 1$ if the machine is working at time t, 0 if it is failed.

We emphasize that if the machine is in changeover, $\alpha(t) = 1$ by assumption.

## 3.2 Evaluation of a production plan

In this section we will describe two ways in which we can evaluate a given production plan. The first is an algorithm that will be central to the development in the remainder of this chapter. We will also describe a simple calculus-based approach that relies on numerical methods.

### Algorithmic approach

We now describe one method to evaluate a production plan, as defined in Section 3.1. This first model is intended only to evaluate the expected cost of a particular production plan. Since we will not consider revenues in our model, the appropriate metric is minimization of total expected cost. Since this model considers decisions over a short horizon, we do not concern ourselves with discounting future costs, although this assumption could be relaxed without loss of generality. The only costs we include in the model at this point are shortfall costs incurred at the shipping deadlines $TD_j$.

The key step for the evaluation of the production plan is how we model the state space. We are able to represent the state of the system at time t by two variables. The first is a quantity $\tau(t)$ that denotes the amount of the production plan that has been completed by time t expressed in terms of machine time. The second is the (binary) state of the machine $\alpha(t)$. We will denote the state of the system at time t by $(t, \tau, \alpha)$. Our solution algorithm will require that a discretization of $\tau$ be chosen. For the simplicity of our examples, we will discretize $\tau$ in unit increments, although any discretization could be chosen.

Before we mathematically describe this system, we consider a few different visual interpretations. In Figure 3.1 we plot time on the horizontal axis and $\tau$ on the vertical axis. By definition we start at $\tau = 0$ at $t = 0$. Suppose the first demand point is at $t = 5$. The value of $\tau$ that we reach at $t = 5$ depends on the amount of time the machine has spent in the failed state. If the time axis and the $\tau$-axis are measured in the same units, then the largest value of $\tau$ that we can achieve by $t = 5$ is $\tau = 5$ (if the machine does not fail). The stack of six circles at $t = 5$ represent the feasible values of $\tau$, i.e., $\tau = 0, 1, \ldots, 5$.

Suppose the next demand point is at $t = 8$. Irrespective of the value of $\tau$ at $t = 5$, we can achieve at most 3 units of work in the interval between $t = 5$ and $t = 8$. Therefore the maximum value of t that we can achieve by $t = 8$ is $\tau = 8$. Further, it is also possible (although perhaps improbable) that $\tau = 0$ at $t = 8$. Note that the points at which demand occurs need not occur at regular intervals.

In Figure 3.1 we represent the possible transitions in the state space from one point in time to the next by a straight line. We have not drawn all of the transitions that are possible from each state. We have only drawn the possible transitions from $\tau = 0$ at $t = 0$, from $\tau = 2$ at $t = 5$, and from $\tau = 4$ at $t = 8$.

In Figure 3.2 we show one realization of this stochastic process. We plot the value of $\tau$ for each point in time as a heavy black line. When the machine is working it produces parts at a (part-dependent) constant rate, so $\tau$ increases linearly, and when the machine is failed $\tau$ remains constant. This results in an upward sloping step-like function.

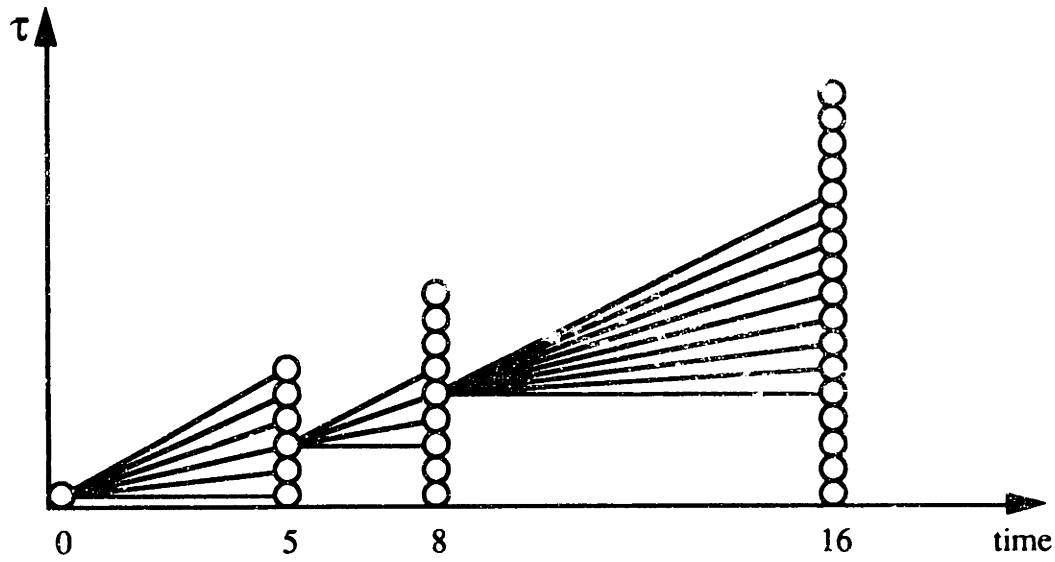


**Figure 3.1** State space representation

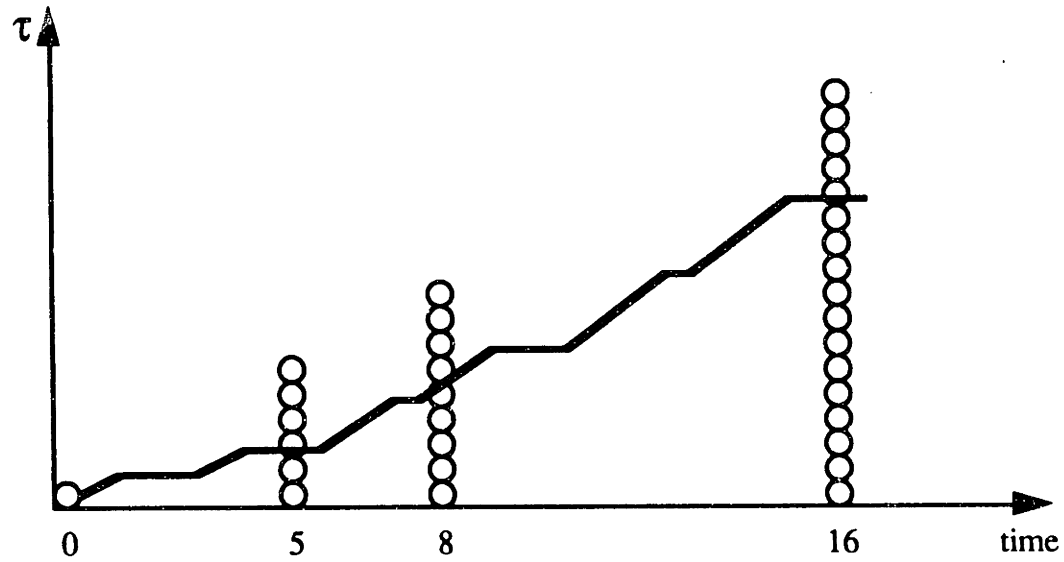


**Figure 3.2** State space with a realization of machine output

To evaluate a production plan we will not require all of the detail that is shown in Figure 3.2. We will only need to know the value of $\tau$ at the demand points, since it is only at the demand points that penalty costs may be incurred. At a demand point, if a sufficient number of parts have been produced to satisfy all demand at the demand point, there will be no penalty costs. This means that there is a threshold value of $\tau$ above which the immediate penalty cost at the state is zero. Below this

value of $\tau$, penalty costs increase as $\tau$ is decreased, until the value of $\tau$ is reached such that no demand is satisfied. The states in which penalty costs occur are darkened in Figure 3.3, where white indicates that no penalty was incurred, black indicates that no demand was satisfied, and shades of gray indicate the quantity of demand that was satisfied.

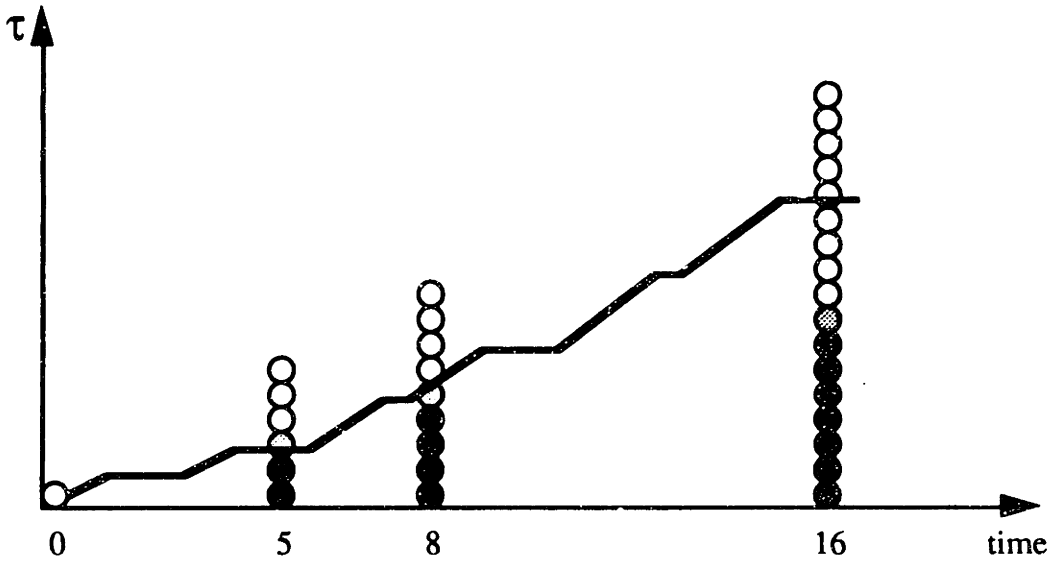


**Figure 3.3** State space with penalty costs

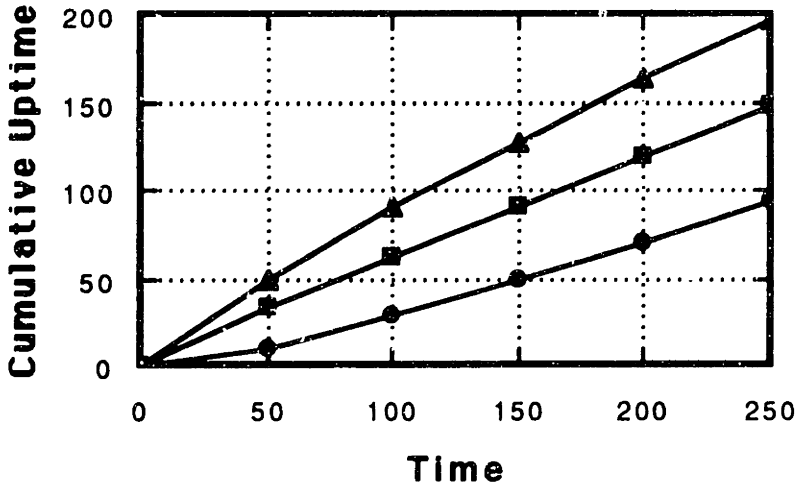


**Figure 3.4** Confidence interval of machine uptime

From the results of Chapter 2 we can determine the probability that machine uptime over an interval is above (or below) any value. This allows us to trace the boundary of a confidence interval for machine uptime as a function of time. Figure 3.4 shows an example of this. The triangles, squares, circles are the 95%, 50% and 5%-iles of the cumulative output of the machine as of time zero, for parameters $\lambda = 3/60$, $\mu = 4/60$. The superposition of the penalty costs from Figure 3.3 and the confidence interval from Figure 3.4 gives the decision maker a visual indication of when and where stockouts are likely to occur.

We now formalize these notions by describing an algorithm for computing the expected future shortfall cost. Let $c_j(\tau, \alpha)$ be the expected future shortfall cost of the state $(TD_j, \tau, \alpha)$ and let $c_0(\tau, \alpha)$ be the expected future shortfall cost as of time zero. The algorithm proceeds backwards, starting at the last demand point. Let $\sigma_j(\tau)$ denote the shortfall at demand point j as a function of the value of $\tau$. $c_M(\tau, \alpha)$ is thus simply equal to the known penalty costs in each state, $cs_M\, \sigma_M(\tau)$. At the M-$j^{th}$ step of the algorithm we compute the expected future shortfall cost for the $j^{th}$ demand point for each state from

$$c_j(\tau_0, \alpha_0) = \sum_{\alpha_1=0}^{1} \sum_{\tau_1 \geq \tau_0} \text{trans}(\tau_0, \tau_1; TD_{j+1} - TD_j \mid \alpha_0\alpha_1)\; c_{j+1}(\tau_1, \alpha_1) + cs_j\, \sigma_j(\tau_0)$$

where $\text{trans}(\tau_0, \tau_1; T \mid \alpha_0\alpha_1)$ is the probability of transitioning from $\tau_0$ to $\tau_1$ in an interval of length T if the initial machine state is $\alpha_0$ and the terminal machine state is $\alpha_1$.

To determine the complexity of the algorithm, let s denote the number of discretized values that $\tau$ can take. Then at each of the M demand points we must

compute the expected future shortfall cost of each possible state, of which there are $O(s)$. This requires computing $O(s^2)$ transition probabilities. Given the transition probabilities, the expected future shortfall cost for any state is then found with a single vector multiplication requiring $O(s)$ multiplications and additions. In total, this algorithm requires $O(s^2 M)$ multiplications and additions, and the computation of $O(s^2 M)$ transition probabilities. Thus, the time to compute the transition probabilities will dictate the running time of the algorithm.

The remainder of this subsection will provide the details for this algorithm, namely, how to compute $\tau$, the immediate penalty costs incurred as a function of $\tau$, the transition probabilities, and the computational effort required to find the transition probabilities. We address each of these in turn.

Determination of $\tau$

Denote the minimum time required to complete production runs 1, ..., i as $U_i$. Then

$$U_i = \sum_{a=1}^{i} S_a + \frac{Q_a}{P_a}.$$

If we are currently producing the $q+1^{st}$ part of the $i^{th}$ production run, then $\tau = U_{i-1} + S_i + q/P_i$. Similarly, if we are s minutes into setting up for production run i, then $\tau = U_{i-1} + s$. In effect, $\tau$ is a measure of cumulative output, measured in time units of machine uptime.

Penalty costs

If the current state is $\tau$, the number of parts of production run i that have been produced is

$$N_i(\tau) = \min\left\{P_i \left[\tau - (U_{i-1} + S_i)\right]^+, Q_i\right\},$$

where $[x]^+$ denotes the greater of zero and x. Thus, each value of $\tau$ uniquely defines how much of the production plan is completed. At demand point j, the shortfall is the cumulative demand minus the cumulative production of part k ($k = JK_j$) minus any starting inventory. Therefore, the shortfall at demand point j is

$$\sigma_j(\tau) = \left[D_j - \sum_{i \in A_j} N_i(\tau) - I_k(0)\right]^+$$

where $A_j = \{i : IK_i = k\}$, the index set of production runs for part k. The penalty costs incurred at demand point j are then $cs_j\, \sigma_j(\tau)$.

Transitions between states

We will denote the current state as $(t_o, \tau_o, \alpha_0)$ and consider transitions to some future state $(t_1, \tau_1, \alpha_1)$ where $t_1 > t_o$ and $\tau_1 \geq \tau_o$. Assume that $\tau_o$ is such that at time $t_0$ we are producing or setting up for the $i^{th}$ production run, and $\tau_1$ is such that at time $t_1$ we are still in the $i^{th}$ production run or setting up for the $i+1^{st}$. With these assumptions, the time available for production during $[t_o, t_1)$ is

$$t_1 - t_o - \left[\tau_1 - U_i\right]^+ - \left[U_{i-1} + S_i - \tau_o\right]^+$$

where the expressions in brackets are zero if we are not setting up at the beginning or end of the interval. Further, in order to reach $\tau_1$ by time $t_1$, we require the uptime over the interval $[t_o, t_1)$ to be

$$\tau_1 - \tau_o - [\tau_1 - U_i]^+ - [U_{i-1} + S_i - \tau_o]^+$$

where again the expressions in brackets are zero if we are not setting up at the beginning or end of the interval. More generally, if $\tau_o$ is such that at time $t_0$ we are producing or setting up the $i^{th}$ production run, and $\tau_1$ is such that at time $t_1$ we are producing the $j^{th}$ production run or setting up for the $j+1^{st}$, $j > i$, then the time available for production during $[t_o, t_1)$ is

$$t_1 - t_o - [\tau_1 - U_j]^+ - [U_{i-1} + S_i - \tau_o]^+ - \sum_{k=i+1}^{j} S_k$$

and the required uptime over the interval $[t_o, t_1)$ is

$$\tau_1 - \tau_o - [\tau_1 - U_j]^+ - [U_{i-1} + S_i - \tau_o]^+ - \sum_{k=i+1}^{j} S_k .$$

Given these results, we can easily state that the condition for feasibility of transition from $(t_o, \tau_o)$ to $(t_1, \tau_1)$: the time available for production must be no less than the required uptime.

Now that we have found the time available for production and the required uptime, we can compute the transition probabilities. Let us consider a simple numerical example. Suppose a transition from $\tau_0$ to $\tau_1$ means that over $[t_0, t_1)$ we complete production of the last 20 units of part 1, incur a 30 minute setup, produce a

batch of 300 units of part 2, incur another 30 minute setup, and produce the first 10 units of part 3. If all time units are expressed in minutes and (for simplicity) all production rates are one per minute, then in the notation of Chapter 2, the probability of transition from $(t_o, \tau_o, \alpha_0)$ to state $(t_1, \tau_1, \alpha_1)$ is

$$G_3(\, 10;\, t_1-t_0-60,\, 20,\, 300 \mid \alpha_0\alpha_1\,) \;-\; G_3(\, 9;\, t_1-t_0-60,\, 20,\, 300 \mid \alpha_0\alpha_1\,),$$

where $G_i(x;\, T, T_1, \ldots, T_{i-1} \mid \alpha(0)=\alpha_0 \;\&\; \alpha(T)=\alpha_1\,)$ = Pr{ x or fewer parts have been produced in the $i+1^{st}$ production run | total time available for production = T, first run requires $T_1$, ..., $i\text{-}1^{st}$ run requires $T_{i-1}$, machine is initially in state $\alpha_0$ and ends in state $\alpha_1$ }. Note that the transition from $(t_o, \tau_o, \alpha_0)$ to state $(t_1, \tau_1+1, \alpha_1)$ is thus

$$G_3(\, 11;\, t_1-t_0-60,\, 20,\, 300 \mid \alpha_0\alpha_1\,) \;-\; G_3(\, 10;\, t_1-t_0-60,\, 20,\, 300 \mid \alpha_0\alpha_1\,),$$

and since we have already computed $G_3(\, 10;\, t_1-t_0-60,\, 20,\, 300 \mid \alpha_0\alpha_1\,)$, we must compute one additional value of $G_i(\cdot)$ for each discretized interval of $\tau$.

We have assumed in the above discussion that the discretization of the state space for $\tau$ occurs in single part increments, although a more fine or more coarse discretization can be chosen.

When machine failures and repairs are i.i.d. exponential (but with possibly different machine reliability parameters $\lambda_k$ and $\mu_k$ for each part), then the distribution $G_i(x;\, T, T_1, \ldots, T_{i-1})$ can be written as a convolution of i distributions of type R, as described in Chapter 2. However, if the machine reliability parameters $\lambda_k$ and $\mu_k$ are the *same* for all parts $k = IK_1, \ldots, IK_{i+1}$, then

$$G_i(x; T, T_1, \ldots, T_{i-1}) = F(x; T - T_1 - \cdots - T_{i-1});$$

that is, $G_i(x; T, T_1, \ldots, T_{i-1})$ is a distribution of type F(t; T) with machine parameters $\lambda_k$ and $\mu_k$, where $k = IK_1$.

As described in the Appendix to Chapter 2, a distribution such as G or F can be evaluated at a point by Laplace transform inversion on a desktop computer in a fraction of a second. For example, on a Power Macintosh 7100/80 in emulation mode with SANE-based math instructions, the time required to evaluate a distribution of type F(t; T) to a high degree of accuracy (absolute error less than $10^{-15}$) is on the order of 0.1 seconds. The computational effort required to evaluate $G_i(x; T, T_1, \ldots, T_{i-1})$ at a point by numerical Laplace transform inversion will be comparable to that for F(t; T), except that the effort grows linearly in i. The rate of growth will depend on the computational effort required to evaluate the Laplace transform at a point.

As a final remark to this subsection, we note that the expected completion time of the production plan is

$$\rho = \sum_{i=1}^{N} S_i + \frac{Q_i}{SAA_i \; P_i}$$

where $SAA_i$ is the *stand-alone availability* $\mu_i / (\lambda_i + \mu_i)$*. If we scale all time units such that the end of the horizon is at time 1, ρ can also be interpreted as the

* This is actually only an approximation if the initial state of the machine is known. However, if the length of the horizon is large relative to the MTBF and MTTR, the quality of the approximation will be excellent.

utilization of the machine required to complete all production by the end of the horizon. In this way ρ gives us some indication for the criticality of the load on the machine. Although this is an important metric, it is not a substitute for the evaluation procedure that we have just described since it can not tell us the likelihood that we will make our shipments on time nor, perhaps more importantly, the expected shortfall.

**Formulation using calculus**

The evaluative model can also be written as a summation of linear loss integrals that compute the expected shortfall cost at each demand point. In particular, the total expected cost can be written as

$$\sum_{j=1}^{M} cs_j \sum_{a=1}^{|A_j|} \left[ L_{aj} + \left(D_j - I_{JK_j}(0) - Q_{a-1,j}\right)^+ \times \left( G_{A_j(a)}\left(0;\ T_{A_j(a),j},\ \frac{Q_1}{P_1},\cdots,\frac{Q_{A_j(a-1)}}{P_{A_j(a-1)}}\right) - G_{A_j(a-1)}\left(\frac{Q_{A_j(a-1)}}{P_{A_j(a-1)}};\ T_{A_j(a-1),j},\ \frac{Q_1}{P_1},\cdots,\frac{Q_{A_j(a-2)}}{P_{A_j(a-2)}}\right)\right)\right],$$

where

$$L_{aj} = \int_0^{Q_{A_j(a)}} \left(D_j - I_{JK_j}(0) - Q_{a-1,j} - x\right)^+ g_{A_j(a)}\left(\frac{x}{P_{A_j(a)}};\ T_{A_j(a),j},\ \frac{Q_1}{P_1},\ \cdots,\ \frac{Q_{A_j(a-1)}}{P_{A_j(a-1)}}\right) dx,$$

$Q_{aj} = Q_{A_j(1)} + \cdots + Q_{A_j(a)}$, and $T_{ij} = TD_j - S_1 - \cdots - S_i$. We define $Q_{0j} = 0$ and the cumulative distribution $G_{A_j(0)}(0;\ T_{A_j(0),j})$ to equal 0. Recall that $A_j$ is the index set of production runs for the part demanded at the $j^{th}$ demand point. We have assumed that the members of the set $A_j$ are $A_j(1), A_j(2), \ldots, A_j(|A_j|)$, indexed such that $IK_{A_j(a)} >$

$IK_{A_j(b)}$ if a > b. The notation $(\ )^+$ denotes the greater of zero and the expression in parentheses.

Although notationally cumbersome, the above expression has a very simple interpretation. $D_j - I_{JK_j}(0) - Q_{a-1,j}$ is the shortfall at demand point j if production runs 1, ..., a-1 of part $JK_j$ are completed but no parts have been produced in production run a. Therefore $L_{aj}$ is the expected shortfall at demand point j given that production run a is still in progress at the demand point. The second term in the square brackets is the expected shortfall at demand point j given production runs 1, ..., a-1 of part $JK_j$ are completed but production run a has not yet started. When summed over all production runs in the set $A_j$ and summed over all demand points j, this gives the total expected shortfall cost.

Provided that we can compute $G_i(\cdot)$ and $g_i(\cdot)$ without difficulty, the numerical challenge in computing the expected total cost from the above expression lies in computing the $L_{aj}$. This can be accomplished by numerical integration or by numerical Laplace transform inversion, as described in Chapter 2. Although our expression for total expected cost says that the number of $L_{aj}$ integrals that we must compute is

$$\sum_{j=1}^{M} |A_j|,$$

there will typically be at most one production run intended to satisfy the demand at any one demand point. Therefore, the number of non-trivial $L_{aj}$'s that must be computed in practice is closer to M.

In summary, we have developed both an analytic and an algorithmic method for evaluating the cost of any particular production plan. These will be important "building blocks" as we explore this model further.

## 3.3 Deciding whether or not to run overtime

In this section we extend the model of the previous section to allow for one or more opportunities to run overtime between now and the end of the horizon. Although we will consider more complex extensions later, for now we extend the model of the previous section in the following way. At certain known points in time the decision maker has the option of purchasing a fixed size block of overtime at a fixed cost. Suppose there are $N_{OT}$ such opportunities, where

$TO_p$ = time of the $p^{th}$ opportunity to run overtime,
$OT_p$ = length of $p^{th}$ overtime opportunity,
$co_p$ = cost of the $p^{th}$ overtime opportunity, $p = 1, \ldots, N_{OT}$.

We assume that the overtime opportunities are indexed such that $a > b$ iff $TO_a > TO_b$.

The problem is to decide whether or not to run overtime to minimize expected stockout and overtime costs. Figure 3.5 is a modification of Figure 3.1 to account for overtime opportunities. We have assumed that there is an overtime opportunity of length 3 somewhere between $t = 10$ and $t = 16$. As a result, the maximum output achievable over the interval if overtime is purchased is now $6 + 3 = 9$. Suppose for simplicity that the time axis and the $\tau$-axis are measured in the same units, and the discretization of the $\tau$-axis is chosen to be in unit increments. Then three additional transitions are possible if overtime is purchased; these are represented by dotted lines in Figure 3.5. Even if overtime is purchased, there is still some positive probability that there is no output over the interval.

In Figure 3.6 we show a modification of Figure 3.2 in which we have an overtime opportunity at t = 12. Since the time available for overtime is not represented on the time axis, the output achieved during overtime is seen as a vertical "jump" at t = 12, which we have represented with a dotted line. The size of this jump is a random variable of type $F(t; OT_p)$ discussed in Chapter 2.

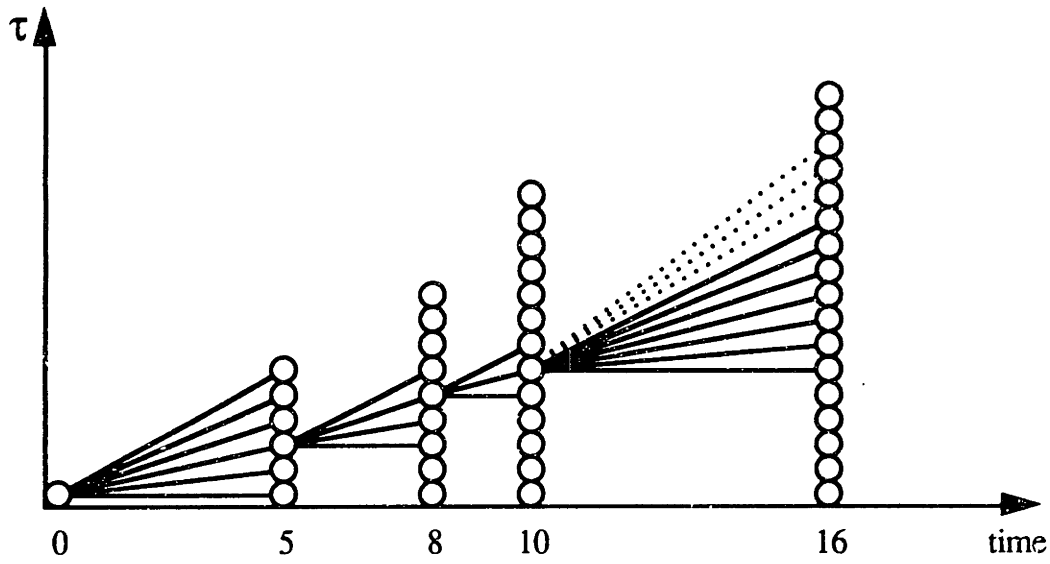


**Figure 3.5** State space representation with overtime opportunity

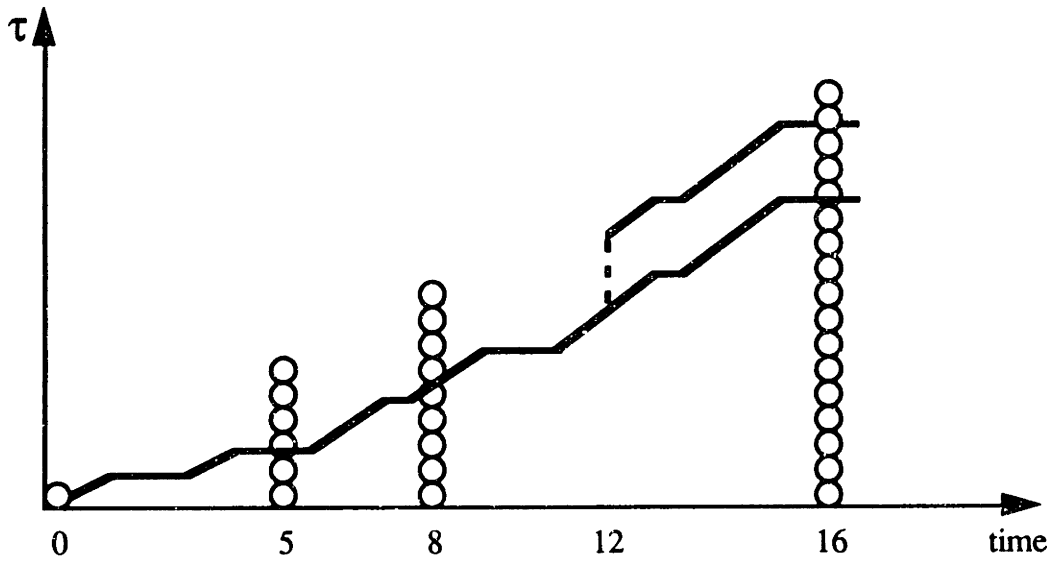


**Figure 3.6** Realization of machine output under with and without overtime

Although we have shown only one overtime opportunity in these examples, we permit an arbitrary number of opportunities located anywhere within the horizon and of any cost.

### Dynamic programming formulation

We will show how to determine whether or not to run overtime at each opportunity by formulating the problem as a dynamic program. Dynamic programming is a well-established methodology for solving problems using digital computers (Bellman, 1957), which we now briefly introduce. The essential idea is to summarize the "state of the world" in one or more *state variables,* where each state has an associated cost (or benefit). At each *stage* of the dynamic program we are told the current state and may need to decide what *action* to take. Before we can find an optimal strategy we must write a recursive relationship to compute the cost of any action given the current state under the assumption that in the future we will behave in a cost minimizing manner. We now introduce some notation for the purposes of this discussion only. Let the stages be indexed by n in reverse chronological order, $\mathbf{s}_n \in \mathbf{S}_n$ be the state at stage n, $\mathbf{a}_n \in \mathbf{A}_n$ be a vector describing the action taken at stage n, and $c_n(\mathbf{s}_n)$ be the optimal expected cost to go with n stages remaining* if the current state is $\mathbf{s}_n$. Then

$$c_n(\mathbf{s}_n) = \min_{\mathbf{a}_n \in \mathbf{A}_n} \left\{ \theta_n(\mathbf{s}_n, \mathbf{a}_n) + \alpha \sum_{\mathbf{s}_{n-1} \in \mathbf{S}_{n-1}} Pr(\mathbf{s}_{n-1} \mid \mathbf{s}_n, \mathbf{a}_n, n)\, c_{n-1}(\mathbf{s}_{n-1}) \right\}$$

where $\alpha$ is a discount rate ($0 \le \alpha \le 1$), $\theta_n(\mathbf{s}_n, \mathbf{a}_n)$ is the cost of being in state $\mathbf{s}_n$ and taking action $\mathbf{a}_n$ at stage n, and $Pr(\mathbf{s}_{n-1} \mid \mathbf{s}_n, \mathbf{a}_n, n)$ is the *transition probability* (the

---

* We adopt the term "cost to go" from Bertsekas (1987). Other authors have called this the "remaining cost".

probability that the next state is $\mathbf{s}_{n-1}$ given that the current stage is n, the current state is $\mathbf{s}_n$ and the action taken is $\mathbf{a}_n$). If the transition probabilities are known and the state and action spaces are finite, then it is a simple matter (at least in principle) to program a computer to find the optimal action for each stage and each possible state by solving the backwards recursion and computing the expected cost for each possible action at each stage. This brief overview of dynamic programming does not even scratch the surface of the well developed theory of the field. The interested reader is referred to one of many excellent introductory texts such as Bertsekas (1987), Denardo (1982) or Bellman and Dreyfus (1962).

We now describe our dynamic programming formulation. Although we will abandon the notation used above to describe dynamic programming, we now describe, in turn, the set of possible states $\mathbf{S}_n$; the stages 1, ..., n; the set of permissible actions $\mathbf{A}_n$; the immediate cost function $\theta_n(\mathbf{s}_n, \mathbf{a}_n)$; and the transition probabilities $\Pr(\mathbf{s}_{n-1} \mid \mathbf{s}_n, \mathbf{a}_n, n)$.

State space

The dynamic program that we construct will bear a close resemblance to the formulation in the previous section. The state space for the dynamic program will have two dimensions $\tau$ and $\alpha$, as described in the previous section. Recall that $\tau$ is a measure of cumulative output and $\alpha$ is the (binary) state of the machine (working or failed).

Stages

The $M + N_{OT} + 2$ stages of the dynamic program will represent the beginning of the first production run at time zero, the M demand points, the $N_{OT}$ points in time

where overtime can be purchased, and the end of the horizon. We index the stages in reverse chronological order.

Actions and immediate costs

At the stages representing overtime opportunities, there are two possible actions: whether or not to purchase a fixed-size block of overtime. In this model, the decision maker will be motivated to purchase overtime if and only if it results in a reduction in total expected costs. At the stages representing overtime opportunities, the immediate costs are a function only of the action taken. If the $p^{th}$ block of overtime is purchased, then we assess an immediate cost $co_p$ at the stage. Otherwise there is no cost assessed.

At the stages representing the demand points, no decisions are made. The immediate costs at these stages are a function only of the state variable $\tau$, as described in the previous section.

A number of different assumptions could be made regarding the terminal costs, i.e., the immediate costs at the stage that represents the end of the horizon. One natural assumption would be to charge a penalty cost equal to the expected cost of the amount of overtime required to complete any unfinished portion of the production plan. The dynamic programming algorithm will be capable of handling any set of terminal costs, although logically speaking, one would expect that the terminal costs would be non-increasing as a function of $\tau$.

Transition probabilities

The state transition probabilities to the next stage are also computed as in the previous section. If the $p^{th}$ block of overtime is purchased, then we add $OT_p$ to the

time available for production when evaluating the transition probabilities from $(TO_p, \tau, \alpha)$ to a state at the next stage.

Optimization

The dynamic program is solved by backward recursion beginning with the second to last stage (closest to the end of horizon) and computing the expected cost to go for each possible value of the state variables $\tau$ and $\alpha$. At stages that represent overtime opportunities, the decision whether or not to run overtime in a particular state is determined by which choice results in the least expected cost to go. The recursion proceeds backwards until the first stage is reached, telling us the expected cost to go at time zero.

Typically, the optimal decisions at each overtime opportunity can be described by a pair of values*. The larger of the two, which we will call the *critical overtime level,* is the value of $\tau$ above which it is not optimal to run overtime. Therefore, the critical overtime level at the $p^{th}$ overtime opportunity is the largest value of $\tau$ at which the overtime cost $co_p$ is exactly equal to the expected reduction in total cost to go if the overtime opportunity is purchased.

The fact that a critical overtime level exists is somewhat intuitive: as the value of $\tau$ decreases, we generally expect the benefit of overtime (in terms of reduced shortfall costs) to increase. Because we consider problems over a finite horizon, this need not be true. It could be that if we are so hopelessly behind schedule (i.e., at a very low value of $\tau$) that we will never catch up by the end of the horizon, so that stockouts

* In Section 3.5 we describe one example we have found where this is not true. This example has extremely large MTBF and MTTR relative to the times between the stages.

are unavoidable even if overtime is purchased. In these cases, purchasing overtime results in a strictly negative expected benefit, since overtime costs are incurred yet there is little or no reduction in expected shortfall. If there exists a $\tau > 0$ such that it is *not* optimal to run overtime below this value of $\tau$, then this is the second of the pair of values. We will call such a value the "lower envelope".

Not running overtime when "hopelessly behind schedule" is clearly not a sensible action. In these situations one can conclude that the production plan is not realistic and should be reconsidered. In such situations, actions are often taken which can not (and we would argue should not) be modeled, such as re-negotiating deadlines or arranging for alternative sources of supply. Although the model can suggest a course of action which is not sensible, this should not be interpreted as an indication that the model is flawed, but rather that this model should not be applied in such a situation. Note that once the lower envelope is found, it is easy to superimpose these levels on the confidence interval for machine output (as shown in Figure 3.4) to ascertain the likelihood of falling "hopelessly behind schedule".

**Computational complexity**

The complexity of the dynamic programming algorithm that we have proposed can be determined in much the same way as the evaluative algorithm of the previous section. Let s denote the number of discretized values that $\tau$ can take. Then at each of the $M + N_{OT} + 1$ stages we must compute the expected cost to go for each possible state, of which there are $O(s)$. This requires computing $O(s^2)$ transition probabilities if the machine reliability parameters are part dependent, and $O(s)$ transition probabilities if not. Given the transition probabilities, the expected cost to go for any state is then found with a single vector multiplication requiring $O(s)$ multiplications and additions. At the stages that represent overtime opportunities, we must do

twice the work, although this does not affect the computational complexity. In total, the algorithm requires $O(s^2 (M + N_{OT}))$ multiplications and additions, the computation of $O(s^2 (M + N_{OT}))$ transition probabilities if the machine reliability parameters are part dependent, and $O(s (M + N_{OT}))$ transition probabilities if not. As before, the time to compute the transition probabilities will dictate the running time of the algorithm.

**Empirical results**

We now present the results of some experiments performed using a computer program (written in Pascal and FORTRAN) that allows a user to perform numerical experiments with a variety of inputs*.

The base case that we will consider is as follows. First we will assume that all parts to be produced have the same parameters (demand, machine reliability, etc.) This is not a necessary assumption of the model, it is made only for simplicity of this discussion. There are five parts to be produced once each over the horizon of length 1000 time units. The production quantity for each part is 120 units and the production rate for each part is one. There are five demand points, one for each of the five parts, for 60 units at intervals of 200 time units. The parts are produced in the order in which they are demanded. If we think of the time horizon as one week, this set of inputs corresponds to a production schedule in which we plan to produce each part every other week.

* The percentiles of the cumulative output of the machine were obtained using Weeks' Method, and the state transition probabilities were obtained using Talbot's Method, as described in the Appendix to Chapter 2.

We set the machine reliability parameters $1/\lambda$ (the mean time between failures, or MTBF) equal to 25, and $1/\mu$ (the mean time to repair, or MTTR) equal to 15. Accordingly, the stand-alone availability (SAA) is MTBF / (MTBF + MTTR) = 0.625, and the (expected) utilization of the machine is (5 * 120 / 0.625) / 1000 = 96%. There are five overtime opportunities of length 20, located 15 time units before each demand point. The costs are normalized so that the per unit backorder cost is always 1.0. The overtime cost is 0.33 per time unit. This data is summarized in Table 3.1.

In all experiments we will assume that setup times are zero. This is done only to make interpretation of the plots easier, and should not be interpreted as a change to the fundamental assumption that we are modeling a production line with non-trivial setup times and/or setup costs such that batching is a practical necessity.

In each experiment, the terminal costs are set to the expected cost of the amount of overtime required to complete any unfinished portion of the production plan. For the base case, the terminal costs are set to $0.33 \times (5 \times 120 - \tau) / 0.625 + 0.33 \times (1-\alpha) / \mu$. The first term is the expected cost of producing on overtime until $\tau = 5 \times 120$, and the second term accounts for the expected cost of the additional overtime that is required if the machine must be repaired before production can resume.

Figure 3.7 shows the confidence interval of machine output if the machine is working at time zero, and the critical overtime levels if the machine is working at the decision point. As in Figure 3.4, the triangles, squares, circles are the 95%, 50% and 5%-iles of the cumulative output of the machine as of time zero. The critical overtime levels are shown as plusses, and can be interpreted as the level above which it is never optimal to run overtime. There is also a "lower envelope" of

| Demand points | | | Overtime Opportunities | | |
|---|---|---|---|---|---|
| Part | Time | Quantity | # | Time | Length |
| 1 | 200 | 60 | 1 | 175 | 20 |
| 2 | 400 | 60 | 2 | 375 | 20 |
| 3 | 600 | 60 | 3 | 575 | 20 |
| 4 | 800 | 60 | 4 | 775 | 20 |
| 5 | 1000 | 60 | 5 | 975 | 20 |

Horizon length = 1000

| | |
|---|---|
| Production batch size = 120 | MTBF = 25 |
| Production rate = 1 | MTTR = 15 |
| Utilization = 96% | SAA = 62.5% |
| Backorder cost = 1 | OT Cost = 0.33 |

**Table 3.1** Data for base case

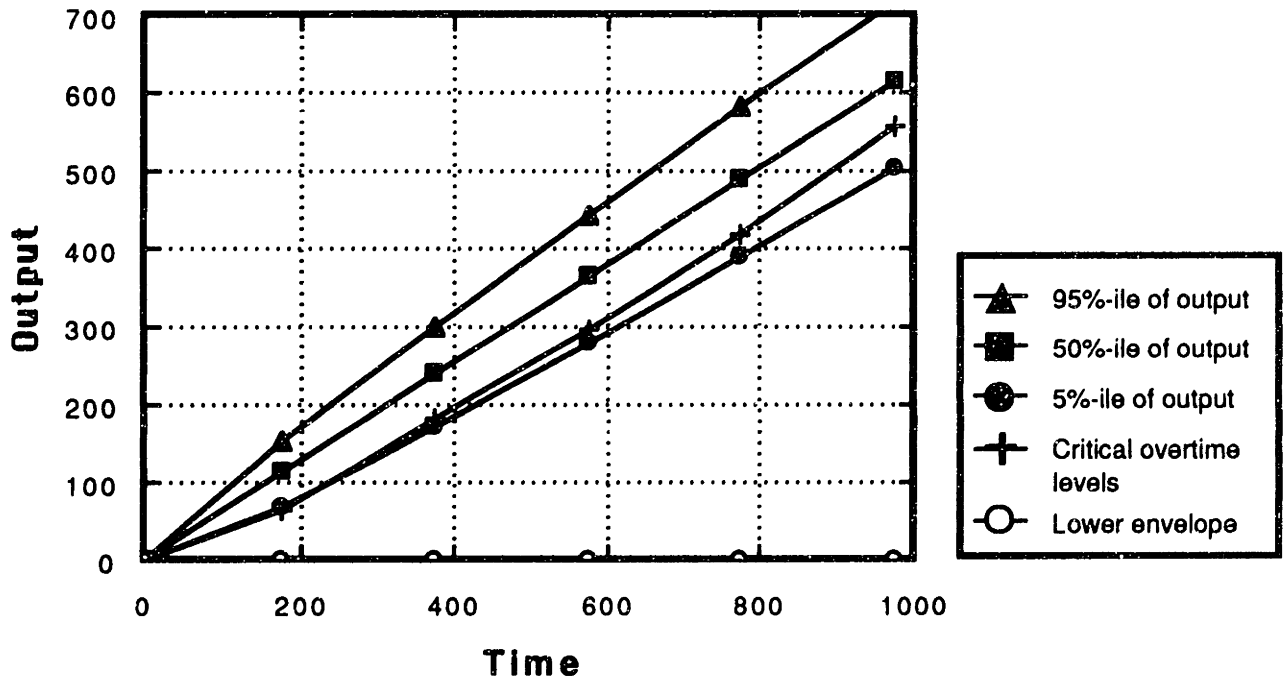


**Figure 3.7** Critical overtime levels when machine is working. Base case. Cost to go = 9.6

overtime levels, represented by hollow circles, below which it is never optimal to run overtime. We see that the critical overtime levels are increasing and convex, indicating that, in this case, we become more willing to run overtime as we approach the end of the horizon. The lower envelope of overtime levels are all zero.

Figure 3.8 shows the confidence interval of machine output if the machine is failed at time zero, and the critical overtime levels if the machine is failed at the decision point. We first note that the cost to go as of time zero increases 29% from 9.6 to 12.4 if the machine is initially failed. We also observe that the critical overtime levels are generally higher at the decision points if the machine is down at that decision point. The critical overtime levels are 14% larger earlier in the horizon, but are 3% smaller at the last decision point. Further, the lower envelope takes on a positive value (440) at the last decision point. This is simply an end of horizon effect.

All experiments were performed on a Power Macintosh 7100/80 running in emulation mode with SANE-based math instructions. To create Figure 3.7, the 5%, 50% and 95%-iles of machine output were each evaluated at five points. These 15 points required a total of 43 seconds to compute. The dynamic program required 3 minutes and 7 seconds to solve with a discretization of the state variable $\tau$ of size 1, resulting in 700 possible discretized values of $\tau$. The computational refinement to be described in Section 3.5 was implemented, although we did not exploit the fact that only two sets of transition probabilities need to be calculated since the time between any two stages is either 20 or 180. See Section 3.5 for a further discussion.

We now consider, in turn, a number of different changes to the base case. In the experiments that follow we show the critical overtime levels only for the case where

the machine is working. When appropriate we will comment if the effect of the change is substantively different if the machine is failed. In all of the experiments that we will describe, a two critical number policy was optimal.

The first change we consider is doubling the number of demand points while keeping the utilization of the machine constant. We accomplish this by creating a 10 part problem with 10 demand points at 100 time unit intervals. The lot size is halved to 60 and the demand at each demand point to 30. We also double the number of overtime opportunities to 10, still located 15 time units before each demand point, and halve the length of the opportunities to 10 time units. The net effect is that, because demand points occur more frequently, there is now significantly less opportunity to fall behind and still catch up before demand must be satisfied.

The resulting critical overtime levels are shown in Figure 3.9. The critical fractiles of machine output are unchanged from the base case. The cost to go increases substantially, as expected, since we have effectively placed additional "constraints" on the system. The critical numbers are also seen to significantly increase.

The above example has shown that it is desirable to have as much time as possible to "catch up" in the event that production falls behind. To demonstrate this principle further, we move the five demand points to the end of the horizon, at times (900, 925, 950, 975, 1000). We place the five overtime opportunities earlier in the horizon, at times (75, 275, 475, 675, 875). No other changes are made other than these timing changes.

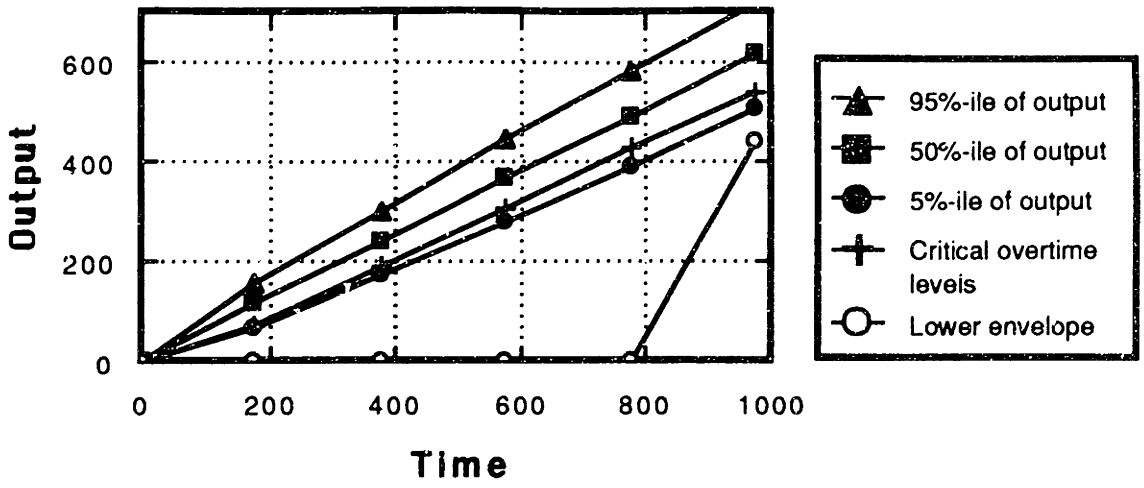


**Figure 3.8** Critical overtime levels when machine is failed. Base case. Cost to go = 12.4

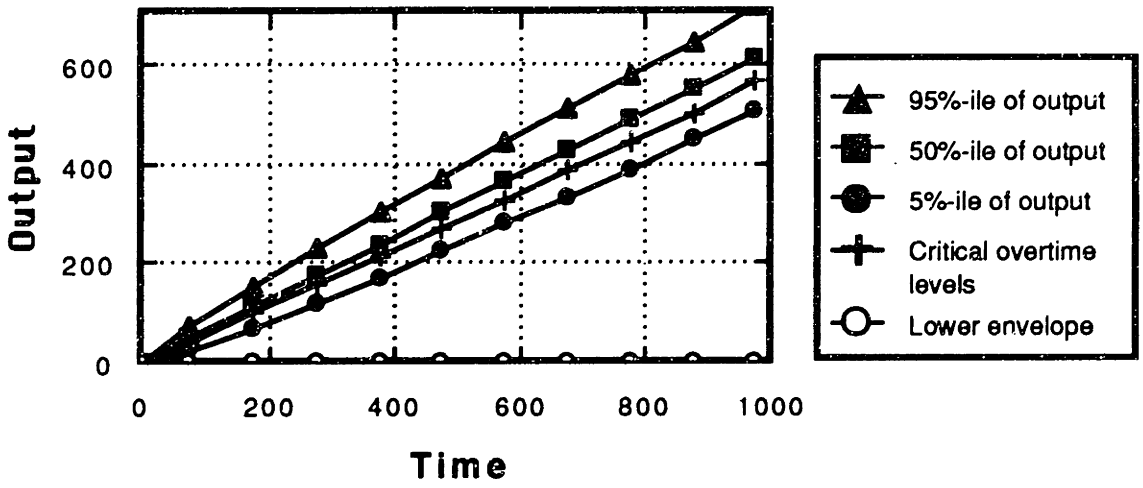


**Figure 3.9** Critical overtime levels with ten demand points. Cost to go = 18.5

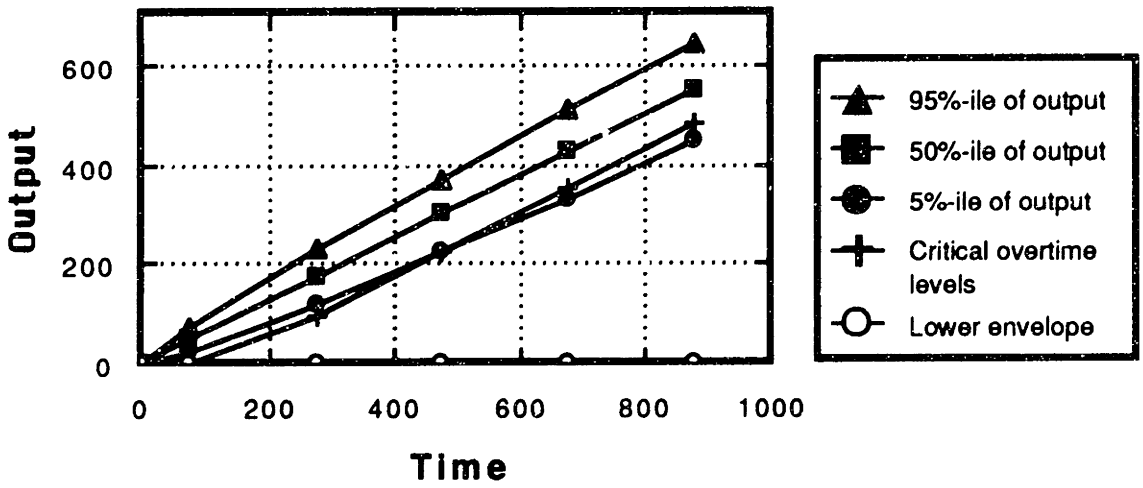


**Figure 3.10** Critical overtime levels with demand at end of horizon. Cost to go = 8.2

The result of these changes is shown in Figure 3.10. First, we note that the cost to go decreases, so the timing changes are indeed beneficial. It is also interesting to note that we are not willing to run overtime at time 75, and that the critical overtime levels are convex increasing. This suggests a form of a wait-and-see strategy.

The final timing change that we consider is shifting the overtime opportunities farther away from demand points, to times (100, 300, 500, 700, 900). As a result, decisions must be made at an earlier time, that is, with less information. The resulting critical overtime levels are shown in Figure 3.11. The magnitude of the change is not large, but the direction of the effect is as we would expect. We note a slight increase in cost, and although it is difficult to see, the critical overtime levels fall substantially early in the horizon (by as much as 63%).

We now consider a variant of the last case, splitting each 20 time unit opportunity into two opportunities of length 10 time units. We place the opportunities at 100 time unit intervals starting at time 75. The net result of these changes is twofold. The first effect is as in the last case, where the decision to run overtime must be made earlier than in the base case. The second effect is the splitting of the opportunity, allowing a smaller sized block of overtime to be purchased. We know that the first effect causes an increase in cost to go and a decrease in the critical overtime levels. The result of these two effects taken together is shown in Figure 3.12. We see a net decrease in cost to go and a slight increase in the critical numbers. This suggests that the decision maker benefits from the added flexibility of smaller, more frequent overtime opportunities. Later in this chapter we will consider extensions to the model where the decision maker can choose the amount of overtime to purchase.

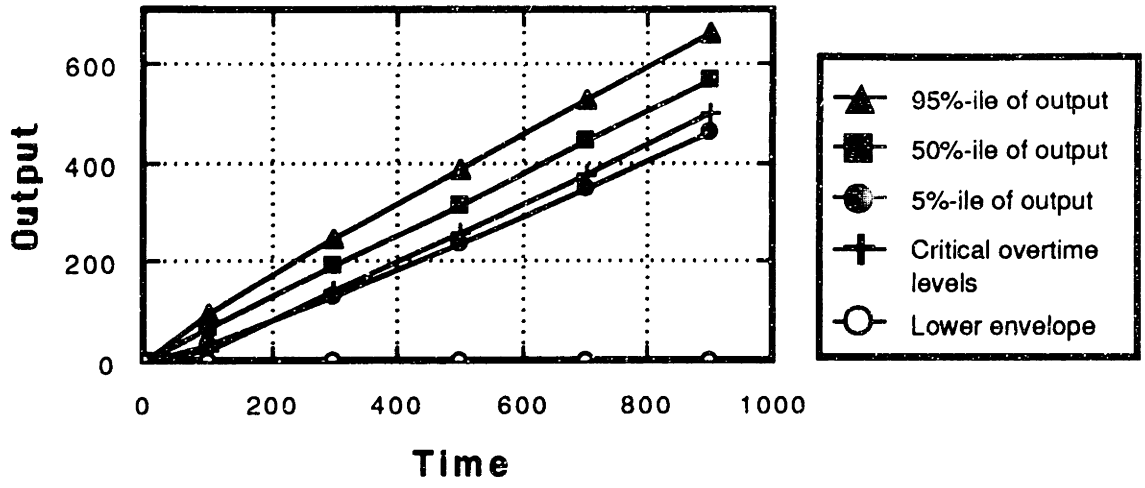


**Figure 3.11** Critical overtime levels with opportunities moved up. Cost to go = 10.0

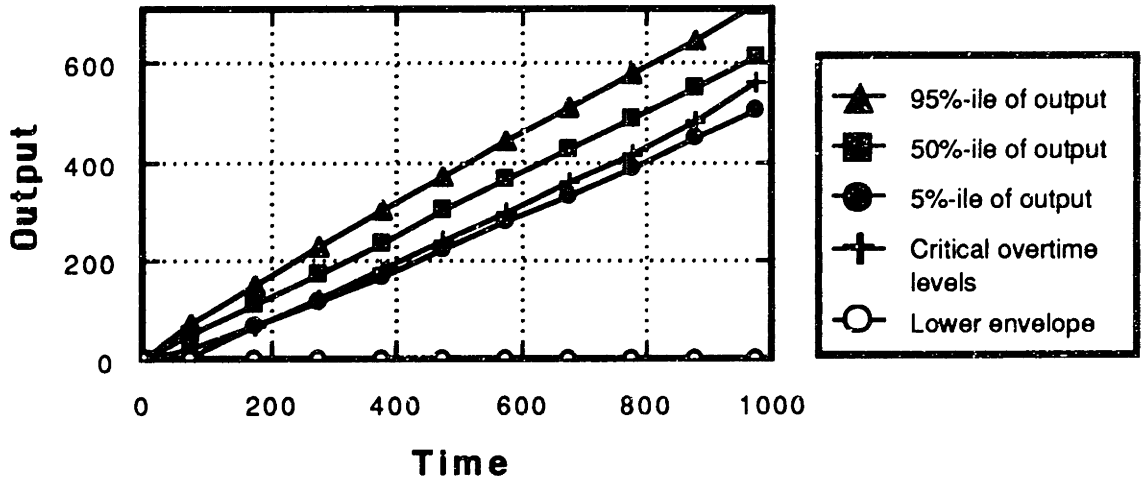


**Figure 3.12** Critical overtime levels with ten opportunities. Cost to go = 9.3

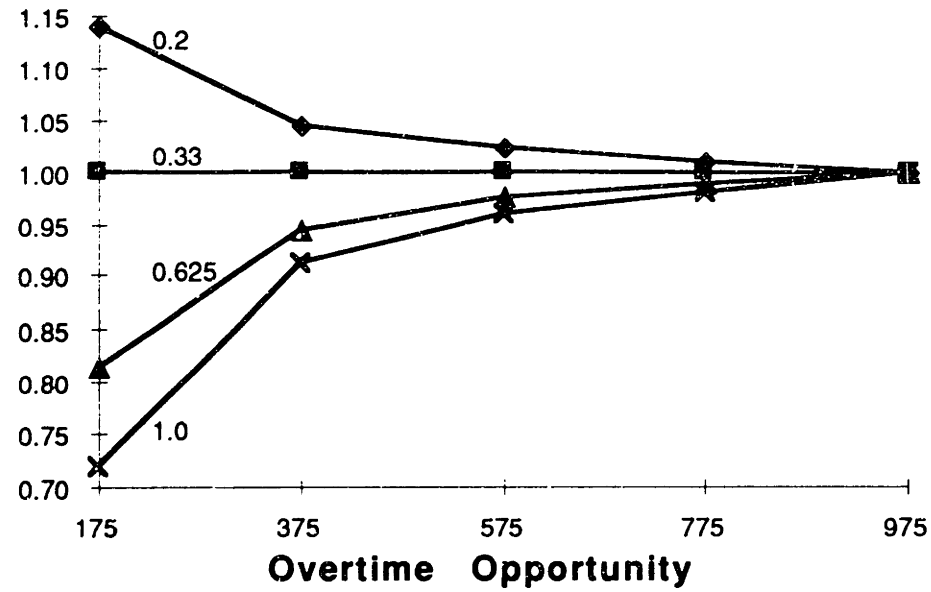


**Figure 3.13** Normalized critical overtime levels with varying per unit overtime cost

We next look at the effect of varying the cost of one time unit of overtime. Four cost levels are considered: 0.33, the base case; 0.625, the cost at which the expected cost to produce a part on overtime is equal to the per unit shortage cost of 1.0; a low (0.2) and a high (1.0) case. Figure 3.13 shows the effect on the critical overtime levels. We see that increasing overtime costs decreases the critical numbers, and that the percent change decreases as we approach the end of the horizon.

We now turn our attention to changes in the machine reliability parameters MTBF and MTTR. We first decrease the MTTR to 7.5 and the MTBF to 12.5. These numbers are chosen such that the stand-alone availability (SAA) of the machine remains constant. The resulting confidence interval of machine output and critical overtime levels are plotted in Figure 3.14. We observe a considerable decrease in the width of the confidence interval, as we would expect since the MTTR has decreased; see the discussion in Chapter 2. Accordingly, there is a dramatic reduction in the cost to go. We also observe a reduction in the critical values,

particularly early in the horizon, due to the reduced variability (less uncertainty in the future).

In Figure 3.15 we show the result of increasing the MTTR to 30 and decreasing the MTBF to 50, again holding the SAA constant. This case is the exact opposite of the previous case. We observe a considerable increase in the width of the confidence interval and an equally substantial change in the cost to go. We also see a general pattern of increase in the critical values, particularly early in the horizon, due to the increased variability (more uncertainty in the future).

The next three charts show the impact of changes in machine utilization. We first decrease the utilization to 80% by decreasing the demand at each demand point to 50 and decreasing the lot sizes to 100. The results are shown in Figure 3.16. We see that this moderate reduction in utilization virtually eliminates the need for overtime: the critical overtime levels are all substantially less than the 5th percentile of the machine output distribution. Furthermore, cost to go has decreased to almost nothing.

In Figure 3.17 we plot the critical overtime levels for various machine utilizations (0.8, 0.9, 0.96, 1.0, 1.04) normalized such that the critical overtime levels for the base case (0.96) are 1.0. The different machine utilizations were achieved by scaling the size of the demand at each demand point, and keeping the ratio of demand to lot size at 1:2. We see that the critical numbers increase as the utilization of the machine increases. We see that the percent increase is greater for the earlier overtime opportunities, although we observe a significant difference between the critical numbers even at the last overtime opportunity in the horizon.

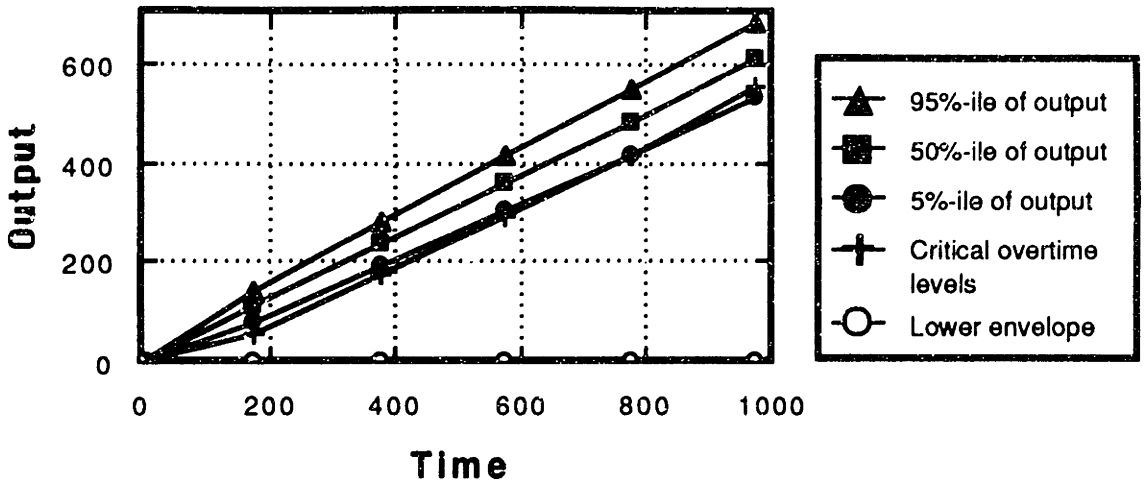


**Figure 3.14** Critical overtime levels with MTTR decreased to 7.5. Cost to go = 4.0

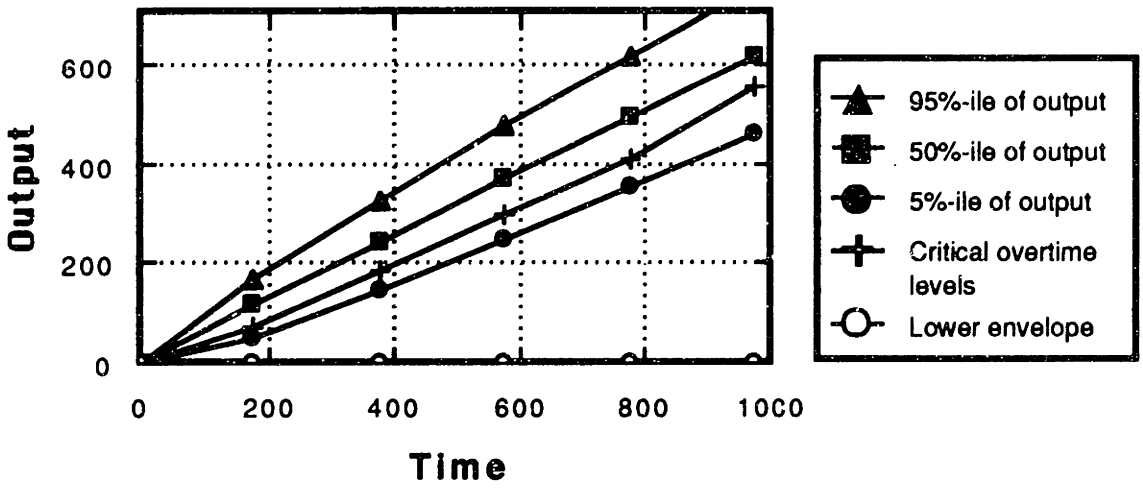


**Figure 3.15** Critical overtime levels with MTTR increased to 30. Cost to go = 21.8

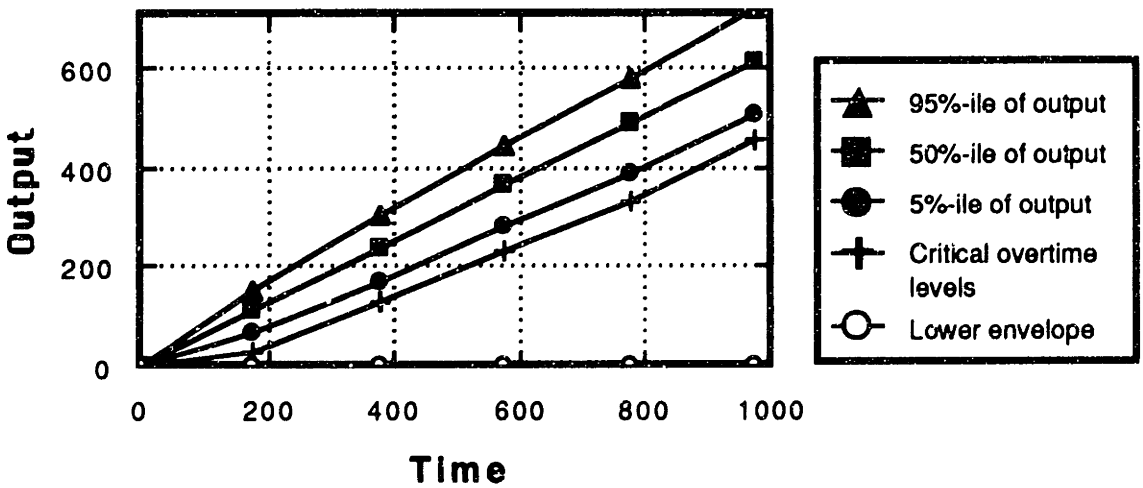


**Figure 3.16** Critical overtime levels with utilization decreased to 80%. Cost to go = 0.6

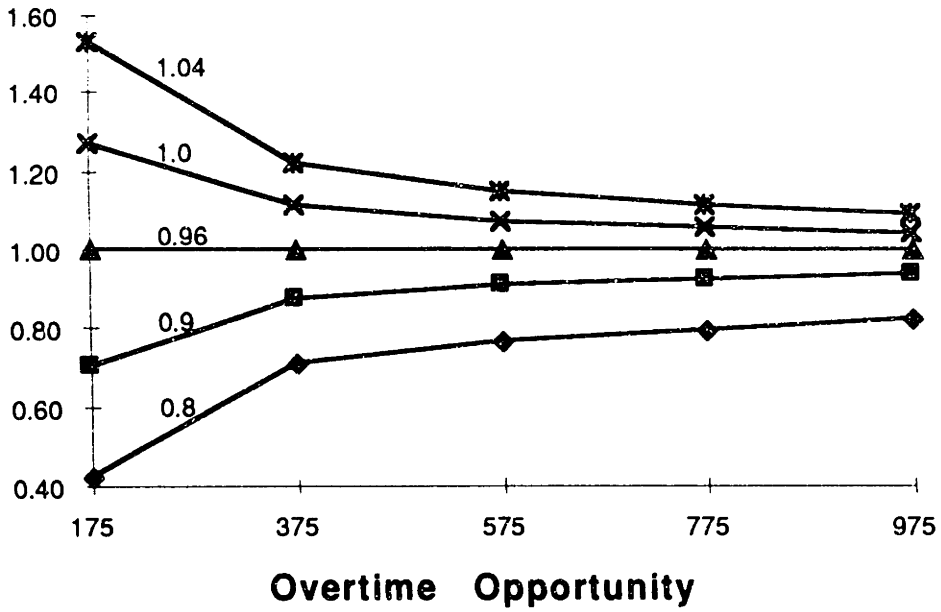


**Figure 3.17** Normalized critical overtime levels with varying machine utilization

Figure 3.18 shows the expected cost to go per unit of total demand for various machine utilizations. We see that the expected cost to go per unit demand increases at a greater than linear rate in the region around full utilization. The point for utilization = 1.04 is not as high as might be initially expected. However, it is easy to see that as utilization is increased beyond 1.0, the probability that an additional unit of work will need to be produced on overtime rapidly approaches one. At utilizations beyond 1.0, the expected cost to go increases linearly (at the overtime cost rate). Thus, the superlinear increase in expected cost to go per part cannot be sustained.

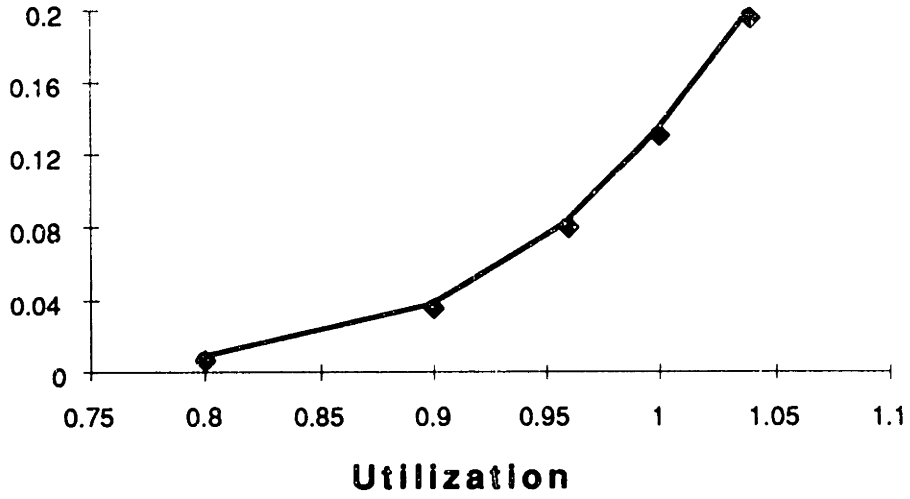


**Figure 3.18** Expected cost per unit demand as a function of machine utilization

## 3.4 Properties of the dynamic programming solution

In this section we will prove certain important properties of the costs and the structure of the optimal policy of the dynamic program formulated in the previous section. This section can be omitted by the reader without loss of continuity.

It will be assumed throughout this section that machine failures are i.i.d. exponential and the same across parts, and repairs are i.i.d. exponential and the same across parts (i.e., the assumptions of Chapter 2 hold). Some additional notation is now introduced for the purposes of this section. First, let $c_n(\tau, \alpha; T)$ be the expected cost-to-go with n stages remaining if the current state is $(\tau, \alpha)$ and there are T time units available for production between stages n and n-1, and let $c^*_{n-1}(\tau, a)$ be the optimal expected cost to go with n-1 stages remaining if the current state is $(\tau, a)$. Then

$$c^*_n(\tau, \alpha) = \min\{c_n(\tau, \alpha; T),\ co_p + c_n(\tau, \alpha; T+OT)\},$$

and

$$c_n(\tau, \alpha; T) = \theta_n(\tau) + \sum_{a=0}^{1} \int_{x=0}^{T} c^*_{n-1}(x+\tau, a)\ \text{trans}(\tau, \tau+x\ ;\ T \mid \alpha a)\ P_{\alpha a}(T)\ dx,$$

where $\text{trans}(\tau_0, \tau_1; T \mid \alpha a)$ is the probability of transitioning from $\tau_0$ to $\tau_1$ in an interval of length T, conditional on transition from $\alpha$ to a in an interval of length T; $P_{\alpha a}(T)$ is the probability that the machine is in state a at time T if it is in state $\alpha$ at time zero; and $\theta_n(\tau)$ is the immediate penalty cost function for stage n.

We will now show that the optimal expected cost-to-go at each stage as function of $\tau$ is non-increasing.

***Theorem 1.*** If $c^*_{n-1}$ and $\theta$ are continuous, non-negative and non-increasing as a function of $\tau$, then $c_n$ is also a continuous, non-negative and non-increasing function of $\tau$.

***Proof.*** Since $\theta$, $c^*_{n-1}$ and the transition probabilities are all non-negative, $c_n$ is non-negative. Further, $c_n$ is continuous because $\theta$ and $c^*_{n-1}$ are continuous. To show that $c_n$ is non-increasing we must show that

$$\frac{\partial}{\partial\tau}c_n(\tau,\ \alpha;\ T) = \frac{\partial}{\partial\tau}\theta_n(\tau) + \sum_{a=0}^{1}\int_{x=0}^{T}\frac{\partial}{\partial\tau}\left\{c^*_{n-1}(x+\tau,\ a)\ \text{trans}(\tau,\ \tau+x\ ;\ T\ |\ \alpha a)\ P_{\alpha a}(T)\right\}\ dx$$

is non-positive. $\partial\theta_n(\tau)/\partial\tau$ is non-positive by assumption, so we need only to show that the integral is non-positive. Since machine failures are i.i.d. exponential and the same across parts, and repairs are i.i.d. exponential and the same across parts, the above integral can be rewritten as

$$\int_{x=0}^{T}\frac{\partial c^*_{n-1}(x+\tau,\ a)}{\partial\tau}\ f(x;\ T\,|\,\alpha a)\ P_{\alpha a}(T)\ dx,$$

where $f(x;\ T\,|\,\alpha a)\ P_{\alpha a}(T)$ is the density of machine uptime as defined in Chapter 2. The integral is non-positive because $\partial c^*_{n-1}/\partial\tau$ is non-positive, and $f(x;\ T\,|\,\alpha a)\ P_{\alpha a}(T)$ is non-negative. ♦

The argument if machine failures and repairs are not i.i.d. exponential or not the same across parts is more complex. However, it is difficult to imagine that this is

not true: if the expected cost to go $c_n$ was not non-increasing then there would exist some value of $\tau$ at which it is optimal to turn the machine off and stop producing. Since we know that there can not be a negative benefit to additional uptime, the expected cost to go can not increase when $\tau$ is increased.

***Theorem 2.*** If stage n represents overtime opportunity p, $co_p \geq 0$ and $c_n(\tau, \alpha; T)$ and $c_n(\tau, \alpha; T+OT)$ are continuous, non-negative and non-increasing as a function of $\tau$, then so is $c_n^*(\tau, \alpha) = \min\{c_n(\tau, \alpha; T),\ co_p + c_n(\tau, \alpha; T+OT)\}$.

***Proof.*** The minimum of two non-negative functions must also be non-negative. Similarly, the derivative of minimum of two functions with non-positive derivatives must also have a non-positive derivative. Note that where the derivative of the minimum does not exist, both one-sided derivatives are non-positive. Lastiy, the minimum of two continuous functions is also continuous. ♦

In Figure 3.19 we show our immediate penalty cost function $\theta_n(\tau)$. The function has this shape for the following reasons. If $\tau$ is so low that we have not produced any parts to satisfy incremental demand at the demand point, then we will incur a penalty equal to the penalty cost rate times the quantity demanded. As $\tau$ increases, this penalty will remain constant until $\tau$ is such that we start producing the part that will satisfy demand at the demand point. $\theta_n(\tau)$ then decreases at a linear rate as parts are produced, until the batch is completed or enough parts have been produced to satisfy all of the demand. As $\tau$ increases beyond this point, the penalty cost does not decrease further. If there are multiple production runs for the part over the horizon, then there can be multiple regions of decrease separated by regions where the penalty cost is constant.

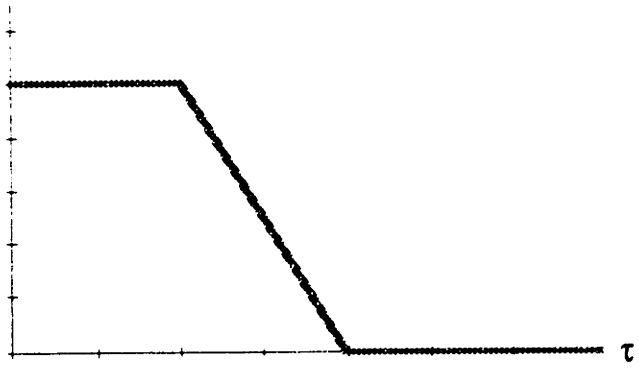


**Figure 3.19** Immediate penalty cost function

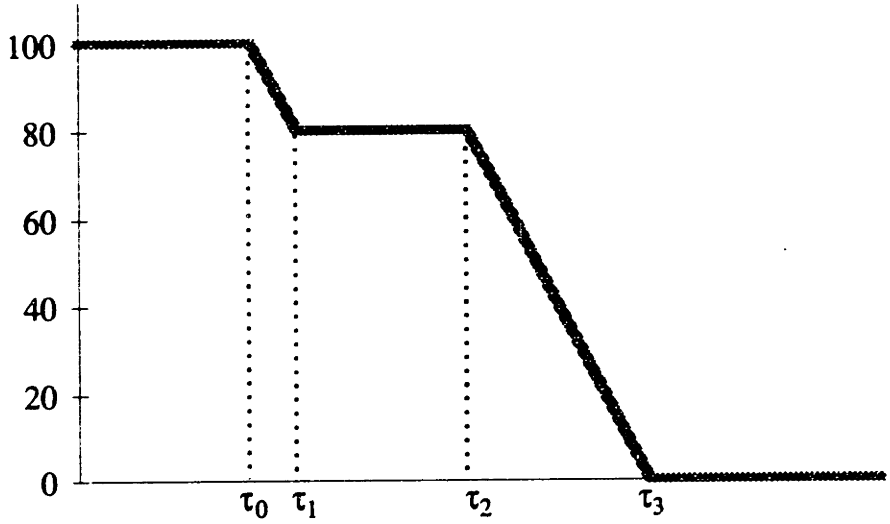


**Figure 3.20** Immediate penalty cost function with multiple production runs

To see this, let us consider a simple numerical example. Suppose there are two demand points of 100 units each for a part, two production runs of 100 units each, and suppose the initial inventory of the part is 20 units. If the per unit shortfall cost is one (for simplicity), then the penalty cost function at the second demand point might appear as shown in Figure 3.20. The value of $\tau$ at which exactly 80 units of the first production run are completed is labeled as $\tau_0$ in Figure 3.20. If by the time of the second demand point, at least 80 units of the first production run are not completed, none of the demand at second demand point will be satisfied, so the shortfall is 100

units. Between $\tau_0$ and the point at which the first production run is completed (labeled as $\tau_1$), the penalty cost function decreases linearly, since this production can be used to satisfy demand at the second demand point. From $\tau_1$ until the second production run begins (labeled as $\tau_2$), the shortfall is 80 units since no additional production occurs. Once the second production run begins, the penalty cost function decreases linearly until 80 units are produced, at which time all demand at the second demand point is satisfied. This point is labeled as $\tau_3$.

Irrespective of the number of production runs, the penalty cost function at any demand point is a continuous, non-negative and non-increasing function of $\tau$. As a result, if the terminal costs are continuous, non-negative and non-increasing, then by induction the optimal expected cost-to-go at each stage is a continuous, non-negative and non-increasing function of $\tau$.

The focus of the development that follows will be to characterize the form of the optimal overtime decisions. Before proceeding, we will require the following

***Definition 1.*** A unidimensional function $f$ is *weakly increasing** if there exists some $x_0$ such that $f(x_0) = 0$, $f(x) \geq 0$ for all $x > x_0$, and $f(x) \leq 0$ for all $x < x_0$.

***Lemma 1.*** If machine failures are i.i.d. exponential and the same across parts, and repairs are i.i.d. exponential and the same across parts, then, ignoring impulses, $f( t; T+OT \mid \alpha a ) P_{\alpha a}(T+OT) - f( t; T \mid \alpha a ) P_{\alpha a}(T)$ is a weakly increasing function of t.

---

* The term weakly increasing and its definition are new.

The proof of this lemma is provided at the end of this section. We are now ready to state

***Theorem 3.*** $c_n(\tau, \alpha; T)$ is a continuous and non-increasing function of T if machine failures are i.i.d. exponential and the same across parts, and repairs are i.i.d. exponential and the same across parts.

***Proof.*** The portion of $c_n(\tau, \alpha; T)$ that is a function of T is the integral

$$\gamma(T) = \int_{x=0}^{T} c^*_{n-1}(x+\tau,\ a)\ f(x;\ T \mid \alpha a)\ P_{\alpha a}(T)\ dx.$$

For any positive value of OT, consider the difference $\gamma(T+OT) - \gamma(T)$. This can be written as

$$\int_{x=0}^{T+OT} c^*_{n-1}(x+\tau,\ a)\left[f(x;\ T+OT \mid \alpha a)\ P_{\alpha a}(T+OT) - f(x;\ T \mid \alpha a)\ P_{\alpha a}(T)\right] dx.$$

This integral is a zero-sum weighted average, since

$$\int_{x=0}^{T+OT} f(x;\ T+OT \mid \alpha a)\ P_{\alpha a}(T+OT) - f(x;\ T \mid \alpha a)\ P_{\alpha a}(T)\ dx = 0.$$

From Theorem 2, $c^*_{n-1}(x+\tau,\ a)$ is a non-increasing function of x. From Lemma 1, we know the expression in brackets is weakly increasing. Thus, our zero-sum weighted average gives negative weight to larger values of $c^*_{n-1}(\cdot)$ and positive weight to smaller values of $c^*_{n-1}(\cdot)$ (since $c^*_{n-1}(\cdot)$ is non-increasing). Therefore, the zero-sum weighted average of $c^*_{n-1}(\cdot)$ must be negative, so $c_n(\tau, \alpha; T)$ is a non-

increasing function of T. The continuity of $c_n(\tau, \alpha; T)$ with respect to T follows from the continuity of $f(x;\ T \mid \alpha a)\ P_{\alpha a}(T)$ with respect to T. ♦

To facilitate a discussion of the optimal overtime decisions, we first require

***Definition 2.*** If stage n represents overtime opportunity p, then the largest value of $\tau$ such that $c_n(\tau, \alpha; T) = co_p + c_n(\tau, \alpha; T+OT)$ is the *critical overtime level* for a given $\alpha$ at stage n. Similarly, the smallest value of $\tau$ such that $c_n(\tau, \alpha; T) = co_p + c_n(\tau, \alpha; T+OT)$ is the *lower envelope* for a given $\alpha$ at stage n.

Note that the critical overtime level and lower envelope need not exist. We can now state

***Theorem 4.*** If both the critical overtime level and lower envelope exist, then the optimal policy will not purchase overtime for any value of $\tau$ above the critical overtime level, or any value of $\tau$ below the lower envelope.

***Proof.*** We know that for sufficiently large $\tau$, $c_n(\tau, \alpha; T)$ and $c_n(\tau, \alpha; T+OT)$ are zero since no stockouts will occur and thus there will be no penalty costs. As a result, for sufficiently large $\tau$, the difference between the cost of purchasing overtime and not purchasing overtime is $co_p + c_n(\tau, \alpha; T+OT) - c_n(\tau, \alpha; T) = co_p$. By definition of the critical overtime level, we know that the costs are equal at that point, and are not equal again. Since the cost of purchasing overtime is eventually greater (by an amount $co_p$), it must be that the cost of not purchasing overtime is less for all $\tau$ greater than the critical overtime level. Using analogous logic, one can prove the result for the lower envelope. ♦

For most cases, the optimal policy is a two critical number policy: run overtime if and only if the value of $\tau$ is between the lower envelope and the critical overtime level. This need not always be true, and extreme cases can be constructed where it is not true. These cases can occur when the lower envelope conditional on a = 0 is larger than the critical overtime level conditional on a = 1. For example, suppose that the failure and repair rates are so low that the probability that the machine fails twice between two stages is very small. In this situation there could be a value of $\tau$ between the two critical numbers such that if the machine fails over the upcoming interval, it is not optimal to run overtime because no demand will be satisfied even with the additional overtime, and if that the machine does not fail over the interval, it is not optimal to run overtime because all demand will most likely be satisfied.

Based on the above reasoning, we suspected such a case might occur with very high MTTR and low SAA, and overtime opportunities that are small relative to the size of the demands. To test this hypothesis, the computer program described in Section 3.3 was used to find the optimal overtime decisions. The parameter settings used were MTTR = MTBF = 100 (SAA = 50%), with demand of 25 parts at intervals of 100 time units, and overtime opportunities of length 10 time units, placed 15 time units before each demand point. The critical overtime levels and lower envelope for these parameters are shown in Figure 3.21. A two critical number policy is not optimal for the ninth overtime opportunity if the machine is failed at the decision point. The critical overtime level is at $\tau$ = 249, and the lower envelope is at $\tau$ = 137. However, it is not optimal to run overtime between $\tau$ = 201 and $\tau$ = 214. As the MTBF and MTTR are increased further, other decision points lose their two critical number structure.

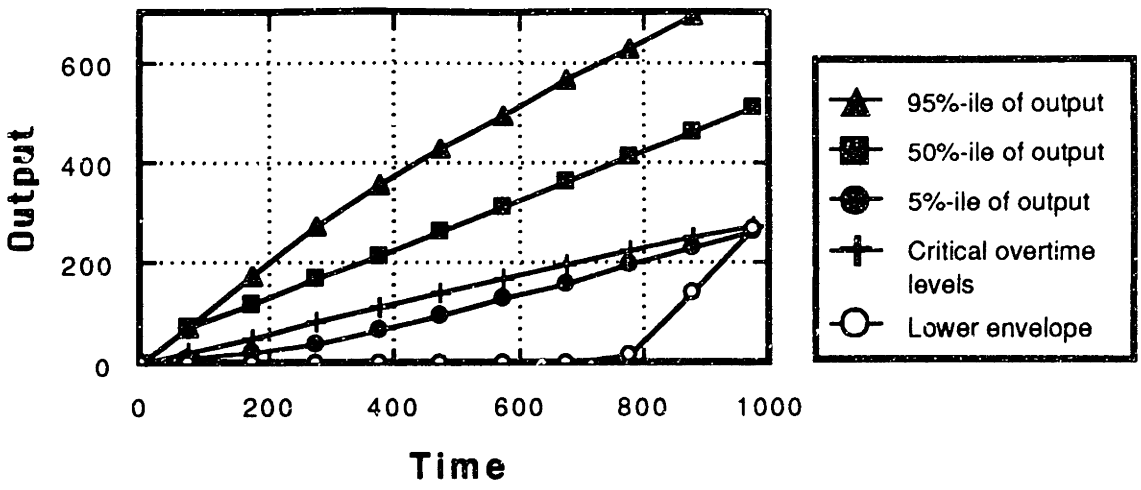


**Figure 3.21** Critical overtime levels when machine is failed. Two critical number policy not optimal.

***Theorem 5.*** If it exists, the critical overtime level at overtime opportunity p is non-increasing as a function of $co_p$, the cost of the overtime block. Similarly, the lower envelope at overtime opportunity p is non-decreasing as a function of $co_p$.

***Proof.*** We have already established that $c_n(\tau, \alpha; T)$ is a non-increasing function of T, so that $c_n(\tau, \alpha; T) \geq c_n(\tau, \alpha; T+OT)$ and thus $\Omega(\tau) = co_p + c_n(\tau, \alpha; T+OT) - c_n(\tau, \alpha; T) \leq co_p$. Furthermore, $\Omega(\tau)$ is equal to $co_p$ for sufficiently large $\tau$, and also for sufficiently small $\tau$. Therefore, $\Omega(\tau)$ is initially decreasing. We expect $\Omega(\tau)$ to appear something like that shown in Figure 3.22. Such a function will intersect the x-axis in an even number of places, or not at all. The points at which the function crosses the axis are the critical numbers. We now show that the rightmost (leftmost) point at which this function first crosses the x-axis is therefore non-increasing (non-decreasing) as a function of $co_p$.

Let $L(y)$ be the set $\{\tau : \Omega(\tau) = y\}$. Let $L^-(y)$ ($L^+(y)$) be the smallest (largest) element in the set $L(y)$. Since $\Omega(\tau)$ is initially decreasing, $L^-(y)$ is initially increasing as y decreases. Since $\Omega(\tau)$ must eventually increase back to $co_p$, at some point $\Omega(\tau)$ will reach a local

minimum; let us call the value of $\tau$ at which this happens $\tau_L$. After $\tau_L$, $\Omega(\tau)$ will start to increase, but these values of $\tau$ cannot be part of the set $L^-(y)$ since these y values were achieved at lower values of $\tau$. At some point $\Omega(\tau)$ may reach a local maximum and then start to decrease again, as shown in Figure 3.22. If $\Omega(\tau)$ decreases lower than $\Omega(\tau_L)$, then while $\Omega(\tau)$ decreases, these values of $\tau$ will be part of the set $L^-(y)$ until another local minimum is reached. Irrespective of the number of local minima and maxima, $L^-(y)$ is increasing in y. We can analogously show that $L^+(y)$ is decreasing in y. Note that an increase (decrease) in $co_p$ effectively shifts the entire function $\Omega(\tau)$ upward (downward) relative to the x-axis. Thus $L^-(y)$ and $L^+(y)$ give the lower envelope and critical overtime levels, and the result is proven. ♦

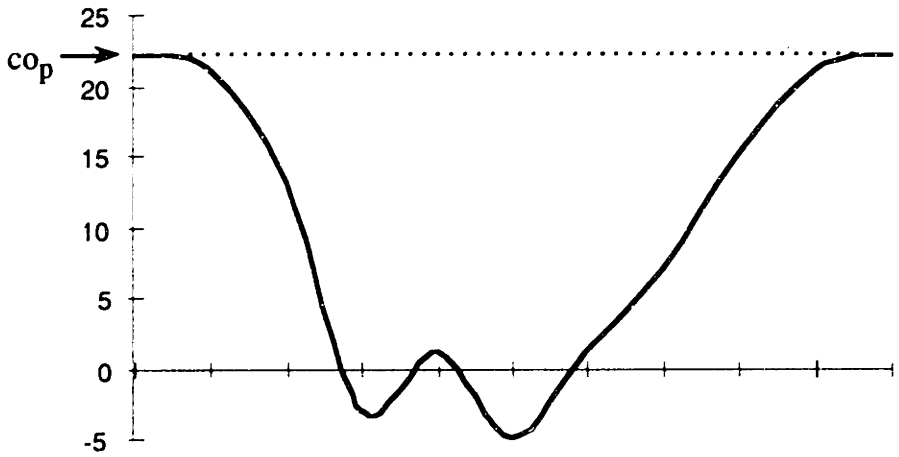


**Figure 3.22** Example of $\Omega(\tau)$, the increased cost as a result of purchasing overtime

As promised, we conclude this section with the following

***Lemma 1.*** If machine failures are i.i.d. exponential and the same across parts, and repairs are i.i.d. exponential and the same across parts, then, ignoring impulses, $f(t; T+OT \mid \alpha a) P_{\alpha a}(T+OT) - f(t; T \mid \alpha a) P_{\alpha a}(T)$ is a weakly increasing function of t.

Our proof will require the following two definitions.

***Definition 2.*** A differentiable uni-dimensional function $f$ is *strictly pseudoconvex* over [a,b] if and only if for $\forall x_1, x_2 \in [a,b]$, $x_1 \neq x_2$, such that $f(x_1) \geq f(x_2)$, the following two conditions are satisfied:

(i) $df(x_1)/dt < 0$ if $x_2 > x_1$,

(ii) $df(x_1)/dt > 0$ if $x_1 > x_2$.

***Definition 3.*** A uni-dimensional function $f$ is *strictly pseudoconcave* over [a,b] if $-f$ is strictly pseudoconvex over [a,b].

***Proof of Lemma 1.*** We will use the notation of the uptime densities derived in Chapter 2. We begin with the case $\alpha = 1$, $a = 0$. After simplification, the difference is

$$f(\, t;\, T+OT \mid 10\,)\, P_{10}(T+OT) - f(\, t;\, T \mid 10\,)\, P_{10}(T) =$$

$$\lambda\ e^{-\lambda t-\mu(T-t)} \left[e^{-\mu OT}\ I_0\left(2\sqrt{x+\lambda\mu t OT}\right) -\ I_0\left(2\sqrt{x}\right)\right],$$

where

$$x = \lambda\, \mu\, t\, (T-t).$$

This expression is valid for $0 \leq t \leq T$. For $T < t \leq T+OT$, $P_{10}(T)\, f(\, t;\, T \mid 10\,)$ is zero and $P_{10}(T+OT)\, f(\, t;\, T+OT \mid 10\,)$ is positive so the difference is positive. For the difference to be weakly increasing we must show that there exists some a, $0 \leq a \leq T$, such that the difference is non-positive for $0 \leq t \leq a$, zero at $t = a$, and non-negative for $a \leq t \leq T$.

Since $\lambda\ e^{-\lambda t-\mu(T-t)}$ is non-negative for all t, it does not affect the sign of $P_{10}(T+OT)\ f(\ t;\ T+OT \mid 10\ ) - P_{10}(T)\ f(\ t;\ T \mid 10\ )$. We will therefore limit our attention to the expression in square brackets since it determines the sign, which we can rewrite as

$$\Delta(t) = k\ I_0\left(2\sqrt{\lambda\mu t(T-t)+\lambda\mu tOT}\right) - I_0\left(2\sqrt{\lambda\mu t(T-t)}\right)$$

where

$$k = e^{-\mu OT}.$$

We now note a number of properties of $\Delta(t)$ that will be of subsequent importance. Let $f_1 = k\ I_0(2\sqrt{\lambda\mu t(T-t)+\lambda\mu tOT})$ and let $f_2 = I_0(2\sqrt{\lambda\mu t(T-t)})$.

i) $0 < k < 1$.

ii) $I_0(t)$ is non-negative, convex and strictly increasing for $t \geq 0$, which follows immediately from its first and second derivatives (Abramowitz and Stegun, 1964).

iii) $f_2$ is strictly increasing as a function of t up to T/2, strictly decreasing after T/2, and symmetric about T/2 over [0, T/2]. $f_1$ is strictly increasing as a function of t up to (T+OT)/2, strictly decreasing after (T+OT)/2, and symmetric about (T+OT)/2 over [0, (T+OT)/2]. These properties follow from (ii).

iv) $f_2$ is strictly pseudoconcave over [0, T] and $f_1$ is strictly pseudoconcave over [0, T+OT]. This follows from (ii) and (iii).

v) At $t = 0$, $\Delta(0) < 0$ since $0 < k < 1$ from (i).

Since $\Delta(0) < 0$, we must show that $\Delta(t)$ crosses the x-axis at most once. Let $\Omega$ be the smallest value of t at which $\Delta(t) = 0$, $0 < \Omega \leq T$. We now show that $\Delta(t) \geq 0$ for $\Omega \leq t \leq T$, i.e., $\Delta(t)$ not cross the axis again. We consider two separate cases.

First suppose $\Omega > T/2$. This implies that $\Delta(T/2) < 0$. An example of this case is shown is Figure 3.23. By definition of $\Omega$, $f_1 < f_2$ for $t < \Omega$. This implies that $f_1$ increases more slowly up to $T/2$. Therefore, due to the symmetry of each function, $f_1$ decreases more slowly. Because of the symmetry and strict pseudoconcavity of each function, and since $f_1$ decreases more slowly it must be greater than $f_2$ for $t > \Omega$.

Now consider the case $\Omega \leq T/2$. An example of this case is shown is Figure 3.24. Since $I_0(t)$ is convex increasing, the difference $k\, I_0(2\sqrt{\lambda\mu t(T-t)+\lambda\mu tOT}) - I_0(2\sqrt{\lambda\mu t(T-t)})$ is increasing. Since it is zero at $t = \Omega$, it must be positive for $\Omega < t \leq T/2$. We can therefore conclude that $\Delta(T/2) > 0$. Since the mode of $f_1$ is greater and to the right of the mode of $f_2$, $f_1$ must remain to the right (and therefore, above) $f_2$ due to the symmetry and strict pseudoconcavity of the two functions. Thus $f_1 > f_2$ for $t > \Omega$.

We have assumed that $f_1 < f_2$ for $t < \Omega$, and shown that $f_1 \geq f_2$ for $\Omega \leq t \leq T$. Therefore $P_{10}(T+OT)\, f(\, t;\, T+OT \mid 10\,) - P_{10}(T)\, f(\, t;\, T \mid 10\,)$ must be weakly increasing. The proof for the case $\alpha = 0$, $a = 1$ follows from the same arguments since $f(\, t;\, T \mid 10\,)$ and $f(\, t;\, T \mid 01\,)$ are nearly identical.

The cases $\alpha = 1$, $a = 1$ and $\alpha = 0$, $a = 0$ are also very similar. We can write the two differences as

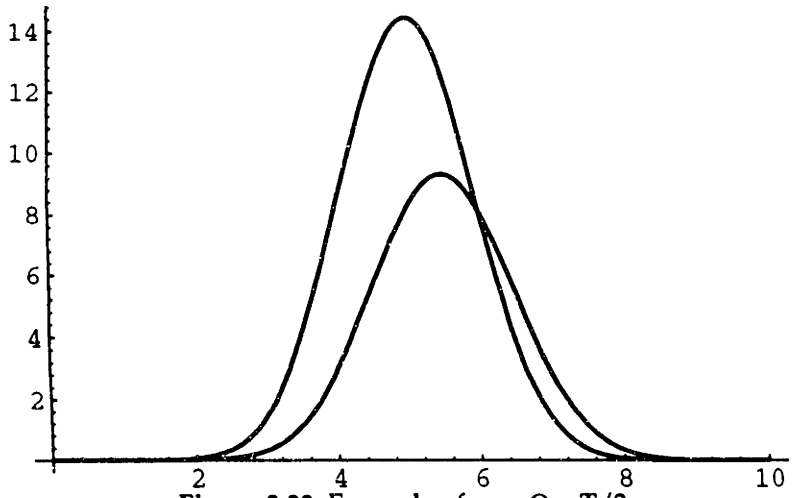

**Figure 3.23** Example of case $\Omega > T/2$

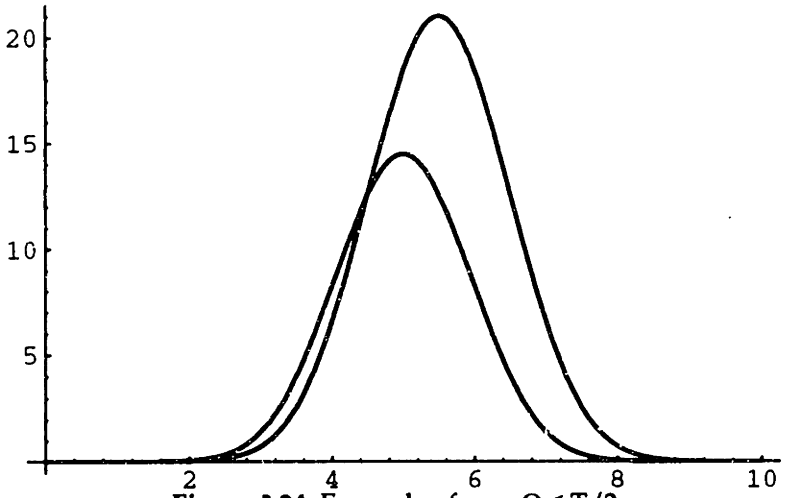

**Figure 3.24** Example of case $\Omega \leq T/2$

$$f(\,t;\,T+OT \mid 11\,)\,P_{11}(T+OT) - f(\,t;\,T \mid 11\,)\,P_{11}(T) =$$

$$\lambda\mu\ e^{-\lambda t-\mu(T-t)}\left[e^{-\mu OT}\ t\ \frac{I_1\left(2\sqrt{x+\lambda\mu tOT}\right)}{\sqrt{x+\lambda\mu tOT}} - t\ \frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}}\right]$$

$$+\ u_0(T+OT-t)e^{-\lambda(T+OT)}\ -\ u_0(T-t)e^{-\lambda T},$$

$$f(\,t;\,T+OT \mid 00\,)\,P_{00}(T+OT) - f(t;\,T \mid 00\,)\,P_{00}(T) =$$

$$\lambda\mu\ e^{-\lambda t-\mu(T-t)}\left[e^{-\mu OT}\ (T+OT-t)\ \frac{I_1\left(2\sqrt{x+\lambda\mu tOT}\right)}{\sqrt{x+\lambda\mu tOT}} - (T-t)\ \frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}}\right]$$

$$+\ u_0(t)e^{-\mu(T+OT)}\ -\ u_0(t)e^{-\mu T}.$$

As before, the quantities that multiply the expressions in square brackets are non-negative and therefore can be ignored. We rewrite the two expressions in square brackets as

$$\Delta_{11}(t) = k_{11}\ t\ \frac{I_1\left(2\sqrt{x+\lambda\mu tOT}\right)}{\sqrt{x+\lambda\mu tOT}} - t\ \frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}}$$

and

$$\Delta_{00}(t) = k_{00}\ (T+OT-t)\ \frac{I_1\left(2\sqrt{x+\lambda\mu tOT}\right)}{\sqrt{x+\lambda\mu tOT}} - (T-t)\ \frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}}$$

where

$$k_{11} = k_{00} = e^{-\mu OT}.$$

We can now see that $\Delta_{11}(t)$ and $\Delta_{00}(t)$ have the same characteristics as $\Delta(t)$. First, $k_{11}$ and $k_{00}$ are both positive and less than one. Further, it is easy to show that $k_{11}\ t\ I_1(2\sqrt{\lambda\mu t(T-t)+\lambda\mu tOT})\ /\ \sqrt{\lambda\mu t(T-t)+\lambda\mu tOT}$ and $t\ I_1(2\sqrt{\lambda\mu t(T-t)})\ /\ \sqrt{\lambda\mu t(T-t)}$ share the same properties as $f_1$ and $f_2$ described above with the exception of property (v). The same is true for $k_{00}\ (T+OT-t)\ I_1(2\sqrt{\lambda\mu t(T-t)+\lambda\mu tOT})\ /\ \sqrt{\lambda\mu t(T-t)+\lambda\mu tOT}$ and $(T-t)\ I_1(2\sqrt{\lambda\mu t(T-t)})\ /\ \sqrt{\lambda\mu t(T-t)}$.

Property (v) deals with the behavior of the difference at t = 0. In this case, the differences are zero at t = 0. However, we know that

$$I_1(t) \sim \tfrac{1}{2}t \qquad \text{(for t small, } t > 0\text{)}$$

from Equation 9.6.7 of Abramowitz and Stegun (1964), so

$$t \frac{I_1\left(2\sqrt{x}\right)}{\sqrt{x}} \sim t \qquad \text{(for t small, } t > 0\text{)}.$$

This means that for t small, $\Delta_{11}(t)$ is approximately $k_{11}\, t - t$ which is negative since $0 < k_{11} < 1$.

The case $\alpha = 0$, $a = 0$ does not exhibit a strict property of this type. In this case $\Delta_{00}(t)$ is approximately $k_{11}\,(T+OT-t) - (T-t)$. If $OT > (1-k_{00})\, T \,/\, k_{00}$ then $\Delta_{00}(t) > 0$ for sufficiently small positive t. However, if $\Delta_{00}(t)$ is positive for all positive t in a neighborhood of zero, then $\Delta_{00}(t) > 0$ for $0 < t \le T/2$. This follows from the fact that $I_1(t)/t$ is increasing and convex in t, so $\Delta_{00}(t)$ can be seen to be increasing. Therefore, this is a special case of $\Omega \le T/2$, where $\Omega = 0$. The argument is therefore unchanged, except that $\Delta_{00}(t) > 0$ for $0 < t < T$. This completes the proof for all four cases. ♦

Although we have proven that $f(\, t;\, T+OT \mid \alpha a\,)\, P_{\alpha a}(T+OT) - f(\, t;\, T \mid \alpha a\,)\, P_{\alpha a}(T)$ is weakly increasing in t only for the case of i.i.d. exponential repairs and i.i.d. exponential failures, we expect this result to hold for a much broader class of failure and repair distributions. For example, we expect that if the uptime distribution is Normal with mean and variance proportional to T, then the result would still hold. This conjecture is based on the fact that $f(\, t;\, T \mid \alpha a\,)$ is asymptotically Normal in T (Takács 1957a, Takács 1957b).

## 3.5 A computational refinement

In this section we describe a method that can be used to reduce the computational effort of the dynamic program when the reliability of the machine (in terms of failure and repair rates) is the same across all parts. This section can be omitted by the reader without loss of continuity.

Recall from Section 3.2 that the dynamic programming algorithm we have described requires $O(s^2 M)$ multiplications and additions, and the computation of $O(s^2 M)$ transition probabilities. Since the determination of the transition probabilities requires significantly more computational effort than vector multiplication (see the Appendix to Chapter 2), the time to compute the transition probabilities will dictate the running time of the algorithm.

The key observation is to recognize that when the reliability of the machine (in terms of failure and repair rates) is the same across all parts, a single transition probability can be reused several times. We now describe this in detail.

Suppose we wish to compute the expected cost of some state $(t_1, \tau_1, \alpha_1)$ and wish to compute the expected cost

$$c_n(\tau, \alpha; T) = \theta_n(\tau) + \sum_{a=0}^{1} \int_{x=0}^{T} c^*_{n-1}(x+\tau, a)\ \text{trans}(\tau, \tau+x\ ;\ T \mid \alpha a)\ P_{\alpha a}(T)\ dx$$

where, as in the previous section, $c_n(\tau, \alpha; T)$ is the expected cost-to-go with n stages remaining if the current state is $(\tau, \alpha)$ and there are T time units available for production between stages n and n-1; $c^*_{n-1}(\tau, a)$ is the optimal cost to go with n-1

stages remaining if the current state is $(\tau, a)$; $trans(\tau_0, \tau_1; T \mid \alpha a)$ is the probability of transitioning from $\tau_0$ to $\tau_1$ in an interval of length T, conditional on transition from $\alpha$ to a in an interval of length T; $P_{\alpha a}(T)$ is the probability that the machine is in state a at time T if it is in state $\alpha$ at time zero; and $\theta_n(\tau)$ is the immediate penalty cost function for stage n.

If the machine reliability is the same across parts, we can rewrite $trans(\tau, \tau+x; T \mid \alpha a)$ as $f(x; T \mid \alpha a)$ since a transition from $\tau$ to $\tau+x$ units implies the same amount of uptime under the same failure process irrespective of the value of $\tau$, assuming that there is not a machine changeover between $\tau$ and $\tau + x$. If a transition from $\tau$ to $\tau + x$ implies s units of setup time (and therefore x-s units of machine uptime), then we can rewrite $trans(\tau, \tau+x; T \mid \alpha a)$ as $f(x-s; T-s \mid \alpha a)$.

Let us consider a simple numerical example to illustrate how transition probabilities can be reused. Suppose the machine is currently set up to produce part 1, and we plan to produce 100 units of part 1, incur a 10 minute changeover, then produce 100 units of part 2. For simplicity, suppose the production rate is one part per minute. Lastly, suppose T, the time between the current stage and the next, is 60 minutes. Note that $trans(\tau, \tau+x; 60 \mid \alpha a) = trans(0, x; 60 \mid \alpha a)$ for $0 \le \tau \le 40$. Note also that $trans(\tau, \tau+x; 60 \mid \alpha a) = trans(110, 110+x; 60 \mid \alpha a)$ for $110 \le \tau < 210$, and further that $trans(110, 110+x; 60 \mid \alpha a) = trans(0, x; 60 \mid \alpha a)$ for $110 \le \tau < 210$.

For $40 < \tau < 110$, it is possible that in the time between the two stages, production of the first part is completed and a changeover occurs. If $\tau + x \le 100$ then no changeover occurs, so $trans(\tau, \tau+x; 60 \mid \alpha a) = trans(0, x; 60 \mid \alpha a)$. If $\tau \le 100$ and $\tau + x \ge 110$ then the changeover is started and completed, so $trans(\tau, \tau+x; 60\text{-}10 \mid \alpha a) = trans(110\text{-}x, 110; 60\text{-}10 \mid \alpha a)$. If $\tau \le 100$ and $100 < \tau + x < 110$ then the changeover is

started but not completed, and in this case each transition probability will be unique. However, these transition probabilities can be reused when $100 < \tau < 110$ by noting that $110-\tau$ is the number of minutes of the changeover that are completed, so that $\text{trans}(\tau, \tau+x; 60-(110-\tau) \mid \alpha a) = \text{trans}(100+(110-\tau)-x, 100+(110-\tau); 60-(110-\tau) \mid \alpha a)$.

In total, we must compute $\text{trans}(0, x; 60 \mid \alpha a)$ at $0 \le x \le 60$ (61 values), $\text{trans}(110-x, 110; 60-10 \mid \alpha a)$ at $0 \le x \le 50$ (51 values), and $\text{trans}(\tau, \tau+x; 60 \mid \alpha a)$ at $40 < \tau \le 100$ and $100 < \tau + x < 110$ ($59 + 58 + \cdots + 50 = 545$ values), for a total of 657 transition probabilities. This is a vast reduction from the $60 \times 210 = 12{,}600$ transition probabilities that would be computed if the algorithm were implemented without reuse.

Although such reuse does not reduce the computational complexity of the algorithm, for most problems this technique will greatly reduce the computational time required to run the algorithm.

Lastly, we note that if the time between two stages is equal to the time between two other stages, the transition probabilities can also be reused.

## 3.6 Static optimal solutions

In Section 3.3 we described an algorithm to determine when it is optimal to run overtime. The solution obtained by this algorithm is *dynamic* since the optimal policy is a function of the state space when the decisions must be made. In contrast, a *static* solution is one in which all decisions are made at a given point in time, and are not a function of the state of the system at future points in time. In this section we will show how to determine the static optimal policy, in which all decisions must be made at the beginning of the horizon and can not be changed over the course of the horizon. We will briefly examine whether or not such static solutions are competitive with the dynamic solutions discussed elsewhere in this chapter.

### Determining static optimal solutions

In Section 3.2 we described a calculus-based approach for evaluating the cost of a given production plan. The evaluation involved the computation of

$$\sum_{j=1}^{M} cs_j \sum_{a=1}^{|A_j|} \left[ L_{aj} + \left(D_j - I_{JK_j}(0) - Q_{a-1,j}\right)^+ \times \left( G_{A_j(a)}\left(0;\ T_{A_j(a),j},\ \frac{Q_1}{P_1},\cdots,\frac{Q_{A_j(a-1)}}{P_{A_j(a-1)}}\right) - G_{A_j(a-1)}\left(\frac{Q_{A_j(a-1)}}{P_{A_j(a-1)}};\ T_{A_j(a-1),j},\ \frac{Q_1}{P_1},\cdots,\frac{Q_{A_j(a-2)}}{P_{A_j(a-2)}}\right)\right)\right],$$

where

$$L_{aj} = \int_0^{Q_{A_j(a)}} \left(D_j - I_{JK_j}(0) - Q_{a-1,j} - x\right)^+ g_{A_j(a)}\left(\frac{x}{P_{A_j(a)}};\ T_{A_j(a),j},\ \frac{Q_1}{P_1},\ \cdots,\ \frac{Q_{A_j(a-1)}}{P_{A_j(a-1)}}\right) dx,$$

and $T_{ij} = TD_j - S_1 - \cdots - S_i$. We refer the reader to Section 3.2 for an explanation of the notation and an interpretation of these expressions.

The above evaluation procedure assumes that no overtime is purchased over the horizon. If instead we wish to evaluate the expected cost of a production plan under the assumption that the $p^{th}$ overtime opportunity is purchased, we simply replace $T_{ij}$ in the above expression with $T_{ij} + OT_p$ for all j such that $TD_j > TO_p$, and re-evaluate the expected cost of the production plan, adding $co_p$ to the cost.

In general, to find the optimal static policy when there are $N_{OT}$ different overtime opportunities, we could evaluate each of the $2^{N_{OT}}$ different possible combinations of running or not running overtime at each opportunity. Each evaluation can be performed by numerical integration or by Laplace transform inversion using the results of Chapter 2.

**An improved algorithm**

In this section we describe approaches to simplify computation of the optimal static policy. These approaches only work if the marginal benefit of additional overtime is decreasing. Although this is not true in general, it is true for a variety of realistic numerical examples that we have explored.

We have seen that purchasing overtime opportunity p replaces $T_{ij}$ with $T_{ij} + OT_p$ for all j such that $TD_j > TO_p$. One can visualize this as "shifting" each demand point after the overtime opportunity to the right by $OT_p$ time units. Before the results of this section can be utilized, the total expected cost function must be shown to be decreasing and convex as each demand point is shifted to the right. This can be accomplished with $O(M)$ evaluations of the total expected cost function.

The algorithm we propose begins by evaluating the expected cost if no overtime is purchased. We then restrict attention to the $N_{OT}$ different plans in which we choose only one overtime opportunity.

If the expected cost rises as a result of choosing one of the $N_{OT}$ opportunities, then we can safely ignore that opportunity since it will have an even smaller impact on reducing expected cost if combined with other opportunities. By ignoring such an opportunity, we can eliminate $2^{N_{OT}-1}$ of the possible combinations in which that opportunity is chosen.

The next step of the algorithm is to consider the different plans in which we choose any two overtime opportunities. Before we evaluate any additional plans, we can first compute lower bounds on the expected cost of any plan in which we choose two overtime opportunities as follows. First we calculate the benefit of choosing one of the opportunities by subtracting the expected cost of purchasing that opportunity from the expected cost if no overtime is purchased. We then do the same for the other opportunity under consideration. We then add the sum of the benefits to the expected cost if no overtime is purchased. When the marginal benefit of additional overtime is decreasing, this total will be a lower bound on the expected cost of purchasing both overtime opportunities. An example of this is given in the next subsection. If this lower bound is higher than any of the expected cost of any of the already computed plans, then the plan can not be optimal. For the combinations of two opportunities that produce a lower bound that is below the lowest expected cost of any already computed opportunity, we should evaluate their expected cost. Whenever we can eliminate a combination of two opportunities (since they will

result in an increase in expected cost), we can eliminate the $2^{N_{OT}-2}$ possible combinations in which those two opportunities are chosen together.

In general, at the $i^{th}$ step of the algorithm we evaluate the expected costs of the plans in which we choose i of the $N_{OT}$ overtime opportunities. The algorithm terminates when we have eliminated all possibilities or have reached the $N_{OT}^{th}$ step of the algorithm. In the worst case, we must evaluate every possible combination. The number of evaluations is then

$$\binom{N_{OT}}{0} + \binom{N_{OT}}{1} + \binom{N_{OT}}{2} + \cdots + \binom{N_{OT}}{N_{OT}}$$

which is equal to $2^{N_{OT}}$. As a result, this procedure can not be worse than the fully enumerative procedure described earlier, except that we will do some additional work in computing the lower bounds. These bounds are extremely simple to compute, however, and will not affect the total running time of the algorithm in any substantive way.

We now briefly examine how the above problem can be viewed as a combinatorial optimization problem. Let S and T be index sets of the overtime opportunities such that $S \subset T$, let a be the index of an opportunity such that $a \notin T$, and let $v(\cdot)$ be the expected total cost of any subset of overtime opportunities. Then the fact that the marginal benefit of additional overtime is decreasing implies that $v(T \cup \{a\}) - v(T) \leq v(S \cup \{a\}) - v(S)$. Accordingly, the function v is submodular. This is a useful observation because a submodular function can be minimized in polynomial time. See Nemhauser and Wolsey (1988). Polynomial submodular function

minimization algorithms are quite complicated, and thus if $N_{OT}$ is not large the above procedure, although not polynomial, may be preferred.

Lastly, we consider a simple special case, where the opportunities are each of the same length, and the opportunities earlier in the horizon are no more expensive than those that occur later in the horizon. In this case, we can see that the earliest opportunities are the most preferable since they afford the greatest protection against stockout. As a result, the best combination when purchasing n opportunities will be to purchase the first n opportunities. Therefore, we need only to evaluate $N_{OT} + 1$ different combinations, where each combination considers purchasing the first n opportunities, $n = 0, \ldots, N_{OT}$.

**Comparison of static and dynamic optimum**

We now briefly examine whether a static solution is competitive with the dynamic solution. To address this question, we consider the following simple example. The production plan involves three parts built one time each over a horizon of 300 time units. We consider a short time horizon since we expect this to be favorable to a static solution. For simplicity we consider a symmetric problem, that is, where all the parts have the same parameters. The parts have demands of 30 units each at 100 time unit intervals and lot sizes of 60 units. The MTBF = 20 and MTTR = 15 time units, for an SAA of 57%. The production speeds are assumed to be one, and we ignore setup times (i.e., assume that they are zero), so that the expected utilization of the machine is 105%. There are three overtime opportunities of length 10 time units located 15 time units before each demand point. The overtime cost is 3.5 per time unit and the stockout cost is 10 per unit. This data is summarized in Table 3.2. As in the base case experiment of Section 3.3, we set the terminal costs to the expected cost of the amount of overtime required to complete any unfinished

portion of the production plan. For the base case, the terminal costs are set to 3.5 x (3 x 30 - $\tau$) / 0.57 + 3.5 x (1-$\alpha$) / $\mu$. This experiment is very similar to the base case experiment of Section 3.3, except with a shorter horizon and slightly higher failure rate (and thus slightly higher utilization).

| Demand points | | | Overtime Opportunities | | |
|---|---|---|---|---|---|
| Part | Time | Quantity | # | Time | Length |
| 1 | 100 | 30 | 1 | 75 | 10 |
| 2 | 200 | 30 | 2 | 175 | 10 |
| 3 | 300 | 30 | 3 | 275 | 10 |

Horizon length = 300

| | |
|---|---|
| Production batch size = 60 | MTBF = 20 |
| Production rate = 1 | MTTR = 15 |
| Utilization = 105% | SAA = 57% |
| Backorder cost = 10 | OT Cost = 3.5 |

**Table 3.2** Data for experiment

The confidence intervals of machine output, critical overtime levels, and lower envelope are shown in Figure 3.25 for the case where the machine is working, and Figure 3.26 for the case where the machine is failed. We note that although the lower envelope is positive at the last decision point (at time 275) when the machine is failed, we see that the envelope is below the 5-%ile of the machine output distribution.

The optimal dynamic solution has an expected cost of 130 if the machine is initially working, and 183 if the machine is initially failed. We expect this large difference in expected costs between the working and failed cases due to the high utilization of the machine and the size of the MTTR relative to the length of the horizon.

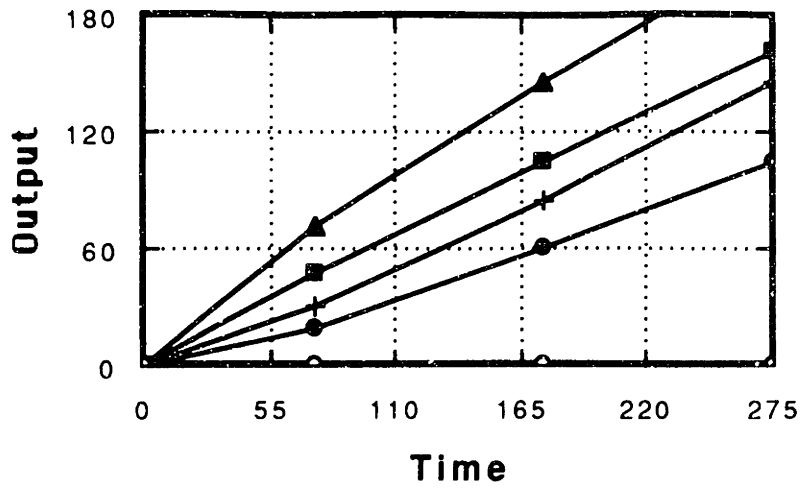


**Figure 3.25** Critical overtime levels and confidence interval of output (machine working).

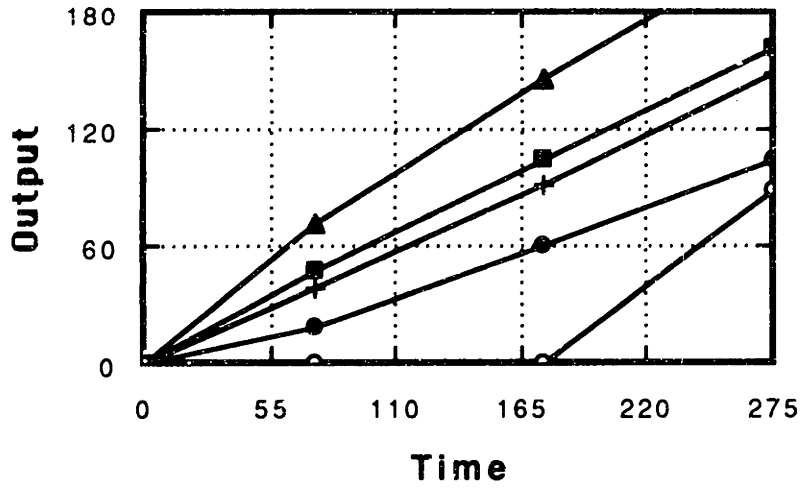


**Figure 3.26** Critical overtime levels and confidence interval of output (machine failed).

| Overtime Opportunities | | | Expected cost | |
|---|---|---|---|---|
| 1 | 2 | 3 | Machine up | Machine down |
| 0 | 0 | 0 | 198 | 280 |
| 0 | 0 | 1 | 203 | 278 |
| 0 | 1 | 0 | 197 | 269 |
| 0 | 1 | 1 | 206 | 271 |
| 1 | 0 | 0 | 196 | 265 |
| 1 | 0 | 1 | 205 | 267 |
| 1 | 1 | 0 | 201 | 260 |
| 1 | 1 | 1 | 214 | 267 |

**Table 3.3** Expected cost of static policies.

In Table 3.3 we show the expected cost of the $2^3 = 8$ different static policies. The ones and zeroes in the first three columns indicate whether or not that overtime opportunity was purchased (1 = yes, 0 = no). We see that the optimal static policy is to purchase overtime opportunity #1 only if the machine is working at time zero, and purchase the first two if the machine is failed at time zero.

We note that as expected, the best policy if only one overtime opportunity is chosen is (1,0,0) and the best policy if two overtime opportunities are chosen is (1,1,0). This is consistent with the principle that overtime earlier in the horizon has greater value.

Let us ignore the fact that this problem has the special structure where the opportunities are of the same length and cost, and examine the lower bounding procedure that we described in the previous subsection for the more general problem. In general, one must examine the total cost function to ensure that the benefit of additional overtime is decreasing before applying the lower bounds. Although we have not done this, it will be evident that the benefit of additional overtime is decreasing for this problem because we have enumerated all the possible solutions.

The bounds are reported in Table 3.4. For the case (0,1,1) for example, the lower bound is computed from the cost of (0,0,1) over (0,0,0) [203 – 198 = 5], plus the cost of (0,1,0) over (0,0,0) [197 – 198 = -1], plus the cost of (0,0,0) [198], for a total of 202. Since the actual expected cost was 206, we report a gap of size 4. Table 3.4 lists four different bounds for the case (1,1,1) since it can be computed in four different ways. The first and weakest is to sum the benefits of (0,0,1), (0,1,0) and (1,0,0). The other

three ways are to sum (0,1,1) and (1,0,0); (1,0,1) and (0,1,0); and (1,1,0) and (0,0,1). Of course, we would have concluded immediately that no policy that chooses overtime opportunity 3 can be optimal, so we would have eliminated policies (0,1,1), (1,0,1) and (1,1,1) without needing to evaluate them.

| Overtime Opportunities | | | Expected cost | |
|---|---|---|---|---|
| 1 | 2 | 3 | Lower bound | Gap |
| 0 | 1 | 1 | 202 | 4 |
| 1 | 0 | 1 | 201 | 4 |
| 1 | 1 | 0 | 195 | 6 |
| 1 | 1 | 1 | 200 | 14 |
| 1 | 1 | 1 | 204 | 10 |
| 1 | 1 | 1 | 204 | 10 |
| 1 | 1 | 1 | 206 | 8 |

**Table 3.4** Lower bounds on expected cost of static policies.

We conclude with the observation that the expected cost of the best static policy is over 50% higher than the cost of the dynamic optimal policy in the case where the machine is working at time zero, and 70% higher in the case where the machine is failed at time zero. Based on this limited evidence, it should be clear that there are benefits to the dynamic optimization that we propose even over short time intervals and moderate variability. Over longer time horizons or under greater production variability (e.g., higher MTTR for a fixed SAA – see Section 2.8), the superiority of dynamic optimization will be even more pronounced.

We do not mean to imply that static optimization can not perform well in certain circumstances. For example, if the machine utilization is very low, the expected amount of overtime purchased may be very low. In an extreme case, the static optimal policy might not purchase overtime, which could be quite competitive (in terms of expected cost) with dynamic optimization. It is important to realize, however, that such cases are not very interesting.

## 3.7 Extensions

In this section we describe a variety of different extensions to the basic model of Section 3.3. These extensions do not change our basic methodology; each involves the solution of a dynamic program by a backward recursion scheme. However, we will see that the structure of these dynamic programs and algorithms to solve them will differ substantially from our basic model.

**Early overtime authorization**

In Section 3.3 we described a model for determining whether or not to purchase overtime opportunities. This model assumes that the overtime opportunities are fixed in length and occur at certain fixed points in time. In some real-world contexts, the decision as to whether or not to run overtime must be made in advance of the point in time that the overtime actually begins. For example, a certain union agreement might require that the decision regarding whether or not overtime is run at the end of the day must be made by 10:00 AM on that same day. We now describe how to incorporate such an extension into our model.

In this section we will show several diagrams such as the one in Figure 3.27, which is intended to represent the dynamic programming logic described in Section 3.3. The circles represent the possible discretized states. This diagram does not distinguish between the two possible machine states, working or failed. The columns of circles represent the stages of the dynamic program (numbered in reverse chronological order). In this diagram time flows from left to right, and the transitions are made from left to right.

The lines between stages represent the possible transitions, each with an associated probability. Two types of lines are shown: solid black and gray. The gray lines that leave a state are intended to represent the transitions if an overtime opportunity is purchased at that state. No gray lines should leave a state if overtime opportunities are not permissible in that state.

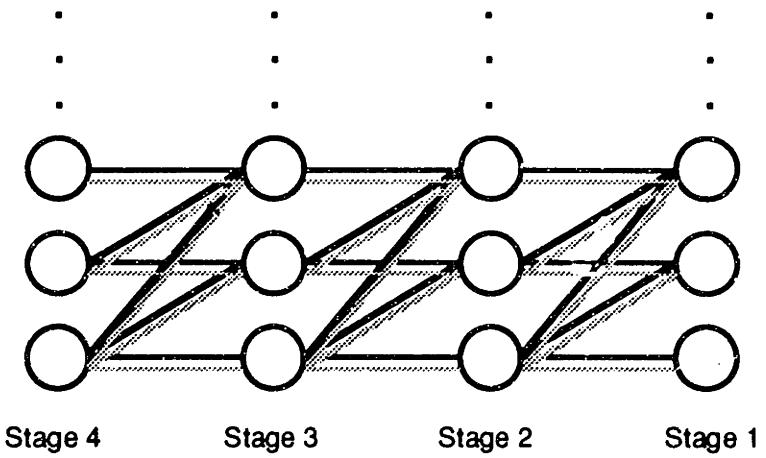


**Figure 3.27** Stages and transitions in dynamic programming algorithm

Now consider the effect of requiring that a decision regarding overtime at the $p^{th}$ opportunity must be made at time $TO_p - x$ $(x > 0)$, instead of $TO_p$. Suppose that the stage that corresponds to the $p^{th}$ overtime opportunity is stage n. We must first change the "time" of stage n to be $TO_p - x$ and then reindex the stages so that they are again in reverse chronological order. If no reindexing is required, then the dynamic program can be solved as before. The decision regarding overtime will be made x time units earlier, so that the time between stages n+1 and n will be x time units shorter and the time between stages n and n-1 will be x time units longer. This will affect the transition probabilities, and accordingly, the expected cost to go that is computed at these stages.

If changing the time of stage n disrupts the chronological order of the stages, then further modification to the dynamic program is required. Suppose that after the stages are reindexed in reverse chronological order, the stage corresponding to the $p^{th}$ overtime opportunity is m, where m > n. Then for each stage m-1, m-2, ..., n, we create a duplicate set of states that we will denote by stage (m-1)', (m-2)', ..., n'. These duplicate stages are incorporated into the model as follows. If we decide to purchase overtime at a state in stage m, then we transition to the duplicate stage (m-1)' instead of stage m-1. This is depicted in Figure 3.28 for the case n = 2 and m = 3.

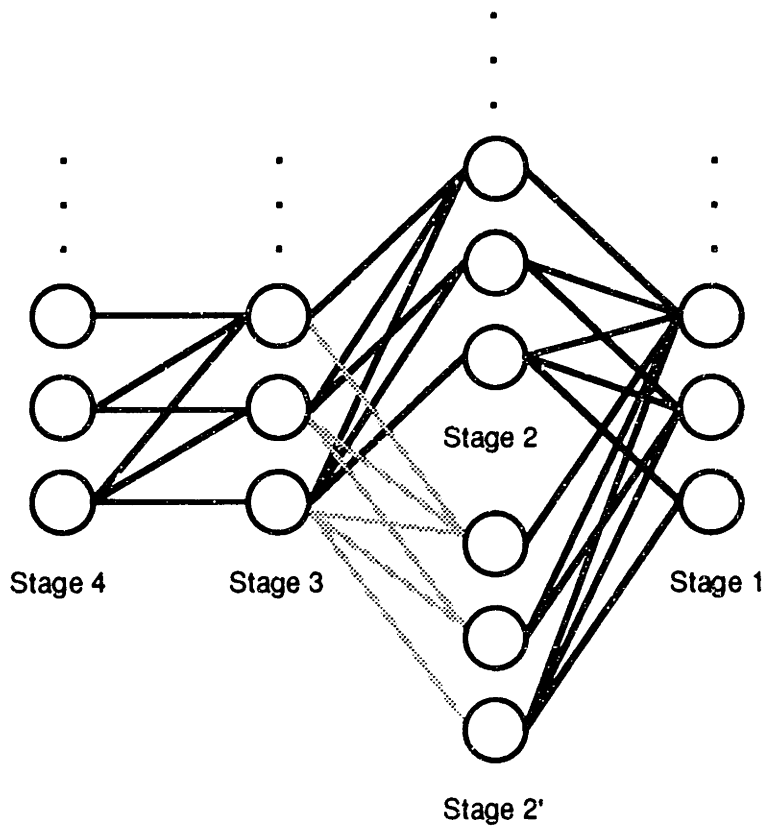


**Figure 3.28** Modified stages and transitions for early overtime authorization

The transition probabilities from stage m to (m-1)' are the same as those from stage m to m-1. Similarly, the transitions between stages (m-1)' and (m-2)', (m-2)' and

(m-3)', ..., (n-1)' and n' are the same as the transitions between stages m-1 and m-2, m-2 and m-3, ..., n-1 and n. The transition probabilities differ only in the transition from stage n' to n-1 (versus the transition from stage n to n-1). When transitioning from stage n', we add $OT_p$ to the time available for production. The net result of these changes is simple: a transition to the duplicate set of states (those whose stage we have denoted with a prime) represents a commitment to purchase the $p^{th}$ overtime opportunity. We do not see the benefit of this purchase until the transition into the $n\text{-}1^{st}$ stage.

Since decisions must be made earlier (i.e., with less information), the total expected cost will increase. We have seen this effect in Theorem 3 of Section 3.4. Further, we expect the critical overtime levels to decrease, as seen in Section 3.3 and Figure 3.11 in an experiment where the overtime opportunities were moved earlier in the horizon.

**Overtime opportunities of variable size**

The model of Section 3.3 assumed that the overtime opportunities are fixed in length. In this and in the following subsection we show two ways to extend our model to incorporate overtime opportunities of variable size.

In this subsection we describe an extension to the model of Section 3.3 in which overtime can be dynamically purchased in a series of discrete blocks. We refer to this extension as dynamic purchasing since after a block of overtime is purchased, the state of the system is observed before a decision must be made whether or not to purchase additional overtime. In contrast, the extension of the next subsection might be called static purchasing, since the quantity of overtime to be purchased is chosen and all the overtime is performed without an opportunity for recourse.

We now assume that the cost of overtime at the $p^{th}$ overtime opportunity $c_p(t)$ is increasing and convex in t, the amount of overtime purchased. With this assumption, we can modify our dynamic program to permit the decision maker to purchase overtime in a series of discrete blocks, where the size of the blocks are determined by the places where the "steps" occur. Based on the current system state, the decision maker can stop running overtime at any of the discrete points. Thus, we are incorporating a continuous choice of overtime quantity into the dynamic program by approximating the cost function $c_p(t)$ as a step function. The discretization can have any number of steps, of any length and size. Figure 3.29 shows a discretization with equal size cost increments. This particular choice of discretization results in a very large minimum purchase. Some care should be taken to choose an appropriate discretization, although real-world circumstances, such as union contracts or other agreements with workers may dictate the appropriate discretization.

We now describe the required modifications to the dynamic program. Previously we used a single stage of the dynamic program to represent the decision of whether or not to purchase a fixed size block of overtime at a particular opportunity. We now model the overtime opportunity as a series of stages, where each stage represents a discrete block. At a stage, the decision maker has the opportunity to purchase the discrete block of overtime. If the block of overtime is purchased, the overtime is performed and the decision maker observes the output of the machine and the state of the machine at the end of the block of overtime before deciding whether or not to purchase the next block. If the decision maker does not purchase the next block of overtime, the overtime opportunity is over. This is because the overtime blocks will be increasing in marginal cost, so that if it is not optimal to

purchase a block, it will not be optimal to purchase an even larger quantity of overtime at a higher per time unit cost. This follows from Theorem 5 of Section 3.4, where we showed that the critical overtime levels are non-increasing as a function of the cost of the overtime block.

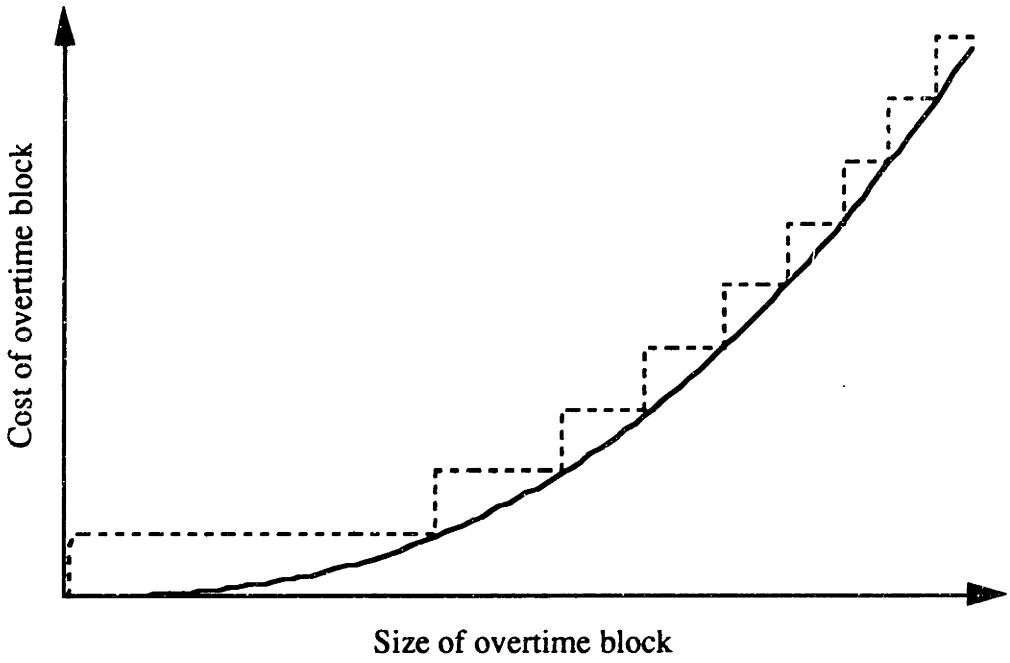


**Figure 3.29** Discrete approximation of convex cost function

Figure 3.30 shows an example of how this modification to the dynamic program is performed. In the example shown, the overtime decision corresponding to Stage 2 is broken into three discrete choices. We have labeled the corresponding stages Stage 2.0, 2.1 and 2.2. At Stage 2.0, the first block of overtime can be purchased, or not. If the block is not purchased, then a transition occurs to Stage 1. If the block is purchased (represented by a gray line), we transition to Stage 2.1 and the cost of the block is incurred immediately. The transition probabilities are determined by the random output of the machine, where the available time for production is equal to the size of the overtime block. At Stage 2.1, we can purchase the second block and

transition to Stage 2.2, or not purchase the block and transition to Stage 1. Lastly, at Stage 2.2 we transition to Stage 1 whether or not we purchase the last overtime block. If we do purchase the block, the cost is immediately incurred and the additional time for production is taken into account in the transition probabilities.

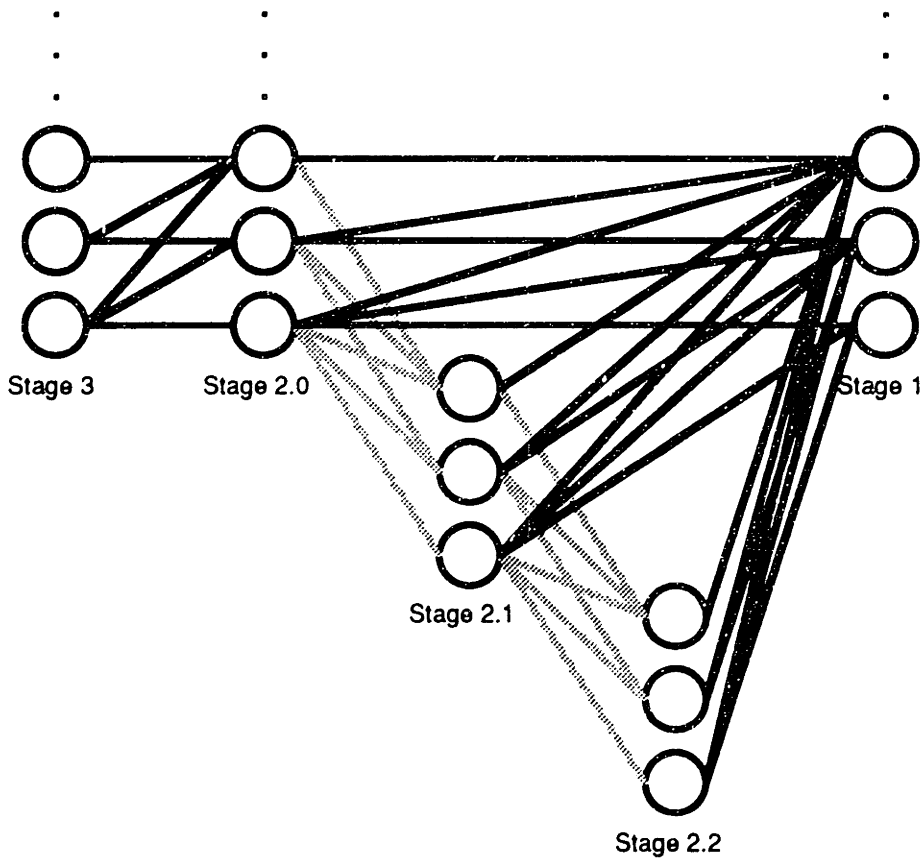


**Figure 3.30** Modified stages and transitions for variable size overtime opportunities

In this instance, to solve the dynamic program we would again start at the end of the horizon and work backwards computing the optimal cost to go at each stage. When we reach a stage where there is a variable sized overtime opportunity (such as Stage 2.0-2.2 in Figure 3.30), the dynamic program can still be solved by backward recursion. In the case of the example in Figure 3.30, once the cost to go has been computed for Stage 1, the cost to go is computed for Stage 2.2 in the usual way. Once

this is finished, the cost to go for Stage 2.1 can be computed for either action (purchase the second overtime block or not), and the optimal decision determined for each state. Once this is finished, the same can be accomplished at Stage 2.0, and then the dynamic programming recursion proceeds as before to Stage 3 and continues to the beginning of the horizon in this fashion.

Replacing a fixed sized opportunity with a variable sized opportunity composed of smaller overtime increments can not result in an increase in total expected cost, since the set of actions available to the decision maker has been expanded at no additional cost. Recall that the computational complexity of the dynamic programming algorithm is linear in the number of stages, so that the computational effort will increase linearly with the number of steps in the discretization of the variable sized opportunity.

**Choosing among a set of overtime opportunities**

In the previous subsection we looked at an extension to the model of Section 3.3 in which overtime could be dynamically purchased in a series of discrete blocks. In the extension of this subsection, the quantity of overtime to be purchased is chosen and all the overtime is performed without an opportunity for recourse.

The extension presented in this subsection is of interest for two reasons. First, we no longer require the restriction of the previous subsection that the cost of overtime at the $p^{th}$ overtime opportunity $c_p(t)$ be increasing and convex in t. Second, the static purchasing scenario may be an accurate representation of reality. We have seen real-world environments in which the quantity of overtime purchased must be decided in advance of the point at which overtime begins, although there is flexibility in terms of how much overtime is purchased.

Let us first consider a variable-sized overtime opportunity at time zero. For example, this would correspond to an opportunity over the weekend before any production begins Monday AM. The dynamic programming algorithm can easily facilitate evaluation of such an opportunity with minimal modification.

Consider Figure 3.31, shown below, a modification of Figure 3.1 from Section 3.2. In this diagram we plot time on the horizontal axis and $\tau$ on the vertical axis. The white circles represent the possible values of $\tau$ that can be reached, while the shaded circles represent values of $\tau$ that are not achievable even if the machine does not fail. The lines between the circles represent the possible transitions. We have not shown every possible transition, only those from $\tau = 0$ (shown as solid lines) and the transitions that would result if the machine did not fail at all (shown as a dashed line).

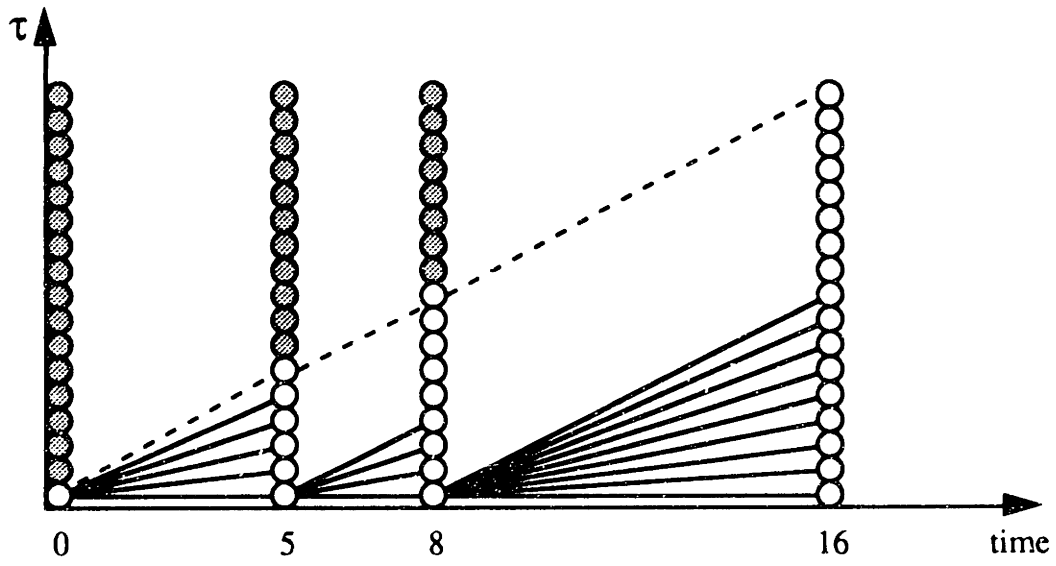


**Figure 3.31** Additional states needed to evaluate variable sized overtime opportunity at time zero

The first step to evaluate an overtime opportunity of variable size at time zero is to solve the dynamic program as before, but instead of limiting attention to those states shown in Figure 3.31 as white circles, compute the expected cost to go at the shaded circles as well. Once the dynamic program has been solved, the expected cost to go vector for time zero tells us the benefit that would result if, instead of starting at $\tau = 0$, we could start at some other value of $\tau$. Denote this vector for a given value of $\alpha$ by $c_N(\tau, \alpha)$.

Denote the cost of purchasing t time units of overtime by c(t). Then the optimal amount of overtime to purchase at time zero is found by solving

$$\underset{0 \le t \le t_{max}}{\textbf{minimize}}\ c(t) + \sum_{a=0}^{1} c_N\left( \textbf{min}(t,\ S_1) + \int_0^{(t-S_1)^+} x\ f\left(x;\ (t-S_1)^+ \mid \alpha\right)\ dx\ ,\ a \right) P_{\alpha a}(t),$$

where t is the amount of overtime purchased, $\alpha$ is the initial state of the machine, a is the state of the machine when the overtime is completed, $S_1$ is the setup time required before production can begin, $\textbf{min}(t, S_1)$ is the amount of the setup that is completed on overtime, the integral is the expected uptime of the machine over an interval of length $(t - S_1)^+$ with initial machine state $\alpha$, so $\textbf{min}(t, S_1)$ plus the integral is the expected value of $\tau$ at the end of the overtime period, and $P_{\alpha,a}(t)$ is the probability that the machine is in state a, given that t time units earlier it is in state $\alpha$.

$t_{max}$ will typically be a constraint on the amount of overtime that is available before time zero, although if such a constraint does not exist, then it should be set to the largest value of $\tau$ that is achievable over the horizon. Since a simple closed form expression has been found for the above integral (given by equations (18) and (19) of

Chapter 2), the optimal value of t can be found with very little effort by simple enumeration.

The development above has assumed that $t_{max}$ is such that production of the first batch can not be completed even if all $t_{max}$ time units are purchased. If this is not the case then the above minimization must also take into account the additional setup times, and machine reliabilities across different parts, if they differ.

We now turn our attention to choosing among a set of overtime opportunities in the middle of the horizon. These overtime opportunities could be of any length and any cost. In Section 3.3 we used a single stage of the dynamic program to represent the decision of whether or not to purchase a fixed size block of overtime at a particular opportunity. To incorporate several different overtime opportunities that are available at a single point in time, we create a stage for each opportunity and place these stages in parallel.

An example of this is shown in Figure 3.32. For simplicity of the diagram we have not drawn the additional gray lines that represent transitions when overtime is purchased. In this example we are replacing the overtime opportunity at Stage 3 with three different overtime alternatives, where each alternative has a different length and cost of overtime (where typically one of the alternatives is to not run overtime and not to incur any overtime cost). We take the subsequent stage, Stage 2 in this example, and replace it with three stages in parallel, which we have labeled 2A, 2B, and 2C. A transition from stage 3 to Stage 2A represents a choice of the overtime alternative "A". Accordingly, the available machine time that is available between Stages 3 and 2A reflects the amount of overtime purchased, and the

immediate cost at Stage 2A is set to reflect the purchase of overtime alternative "A". The transitions and costs from Stages 2A-2C to Stage 1 are unchanged.

The addition of alternatives broadens the set of actions that are available to the decision maker at no additional cost, so the total expected cost can not increase. Each additional alternative adds one stage to the dynamic program. Recall that the computational complexity of the dynamic programming algorithm is linear in the number of stages, so that the computational effort will increase linearly with the number of alternatives presented.

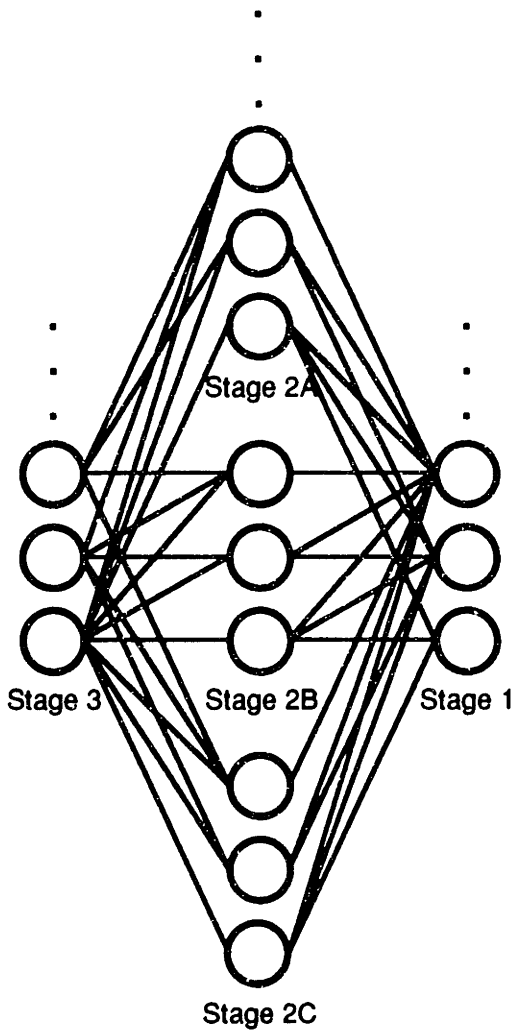


**Figure 3.32** Modified stages and transitions for choosing among a set of overtime opportunities

**Constraining the number of overtime opportunities used**

In the development thus far, we have assumed that there is no restriction on the number of overtime opportunities that can be purchased. In some real-world situations however, there may be such constraints, for example, where only two out of any three consecutive weekends can be used for overtime. We now show how to accommodate such a constraint into our model.

The basic idea is to make copies of all the states and stages in the dynamic program. We will call such a copy a *layer*. As before, the initial stage corresponds to the beginning of the horizon. We start out in Layer 0. The dynamic program is structured as before, where transitions occur to the next stage (within the same layer). The difference is that if we choose to run overtime, we transition to the next stage, but in one layer higher. This is depicted in Figure 3.33. The number of layers is equal to the maximum number of times that we are permitted to run overtime over the horizon, plus one. The layer number indicates how many times we have run overtime thus far. In the topmost layer, we do not permit overtime to be run, thereby enforcing the constraint.

The dynamic programming algorithm proceeds very much like before. It starts with the topmost layer and performs the backward recursion the first stage is reached. This can be done because the cost to go at any stage in Layer 2 is not a function of the other layers. We then move one layer downward, and perform the backward recursion starting with the last stage. The cost to go in this layer is only a function of the cost to go at the topmost layer. This continues one layer at a time until the first stage of Layer 0 is reached. Note that we do not need every stage in every layer, since we can only transition up one layer per stage. Therefore, on the $n^{th}$ layer the stages

up to and including the stage that represents the $n^{th}$ overtime opportunity can be omitted.

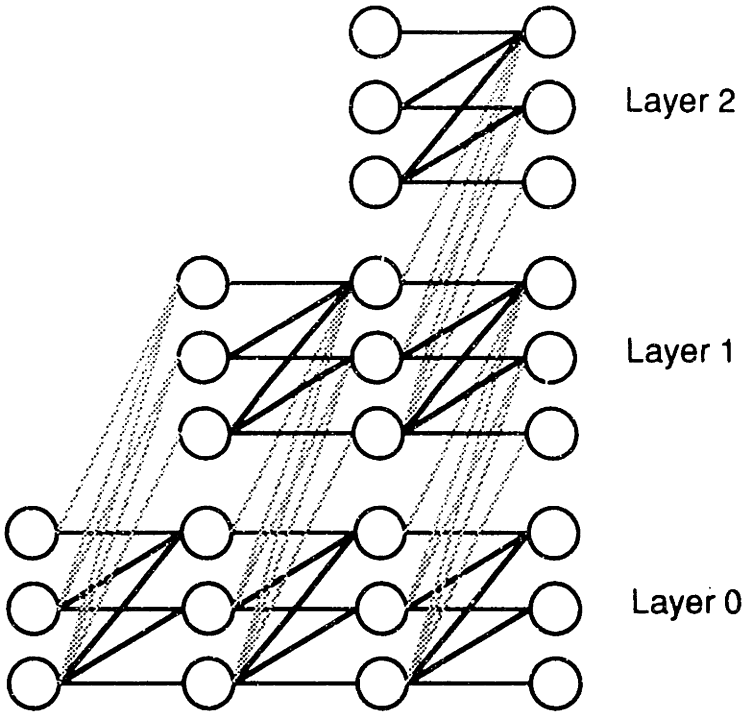


**Figure 3.33** Modified stages and transitions for choosing among a set of overtime opportunities

Although the above procedure at first appears to entail a significant amount of additional computation, this is in fact not the case. Since the transition probabilities are the same between any two successive stages irrespective of which layer we are evaluating, the number of transition probabilities that need to be computed does not increase as a result of the addition of layers. Recall that the computation of the transition probabilities will dictate the running time of the algorithm, so this extension requires very limited additional computational effort.

As a result, we observe that we are able to obtain sensitivity information on the number of overtime opportunities that are permitted. In particular, by computing the expected cost to go at a few additional stages we can evaluate the impact of

reducing the number of overtime opportunities that are available. In Figure 3.34 the additional stages that need to be evaluated are shaded in gray. In this example there are two layers so we permit two overtime opportunities. By computing the expected cost to go at the first stage on Layer 1, we will have determined the increase in total cost that would result if only one overtime opportunity were available. Similarly, the expected cost to go at the first stage on Layer 2 tells us the increase in total cost that would result no overtime opportunity were available.

Since the addition of layers is not computationally expensive, it is therefore quite practical to add Layers -1, -2, ... below Layer 0, that tell us the benefit that results if we had one, two, ... extra overtime opportunities.

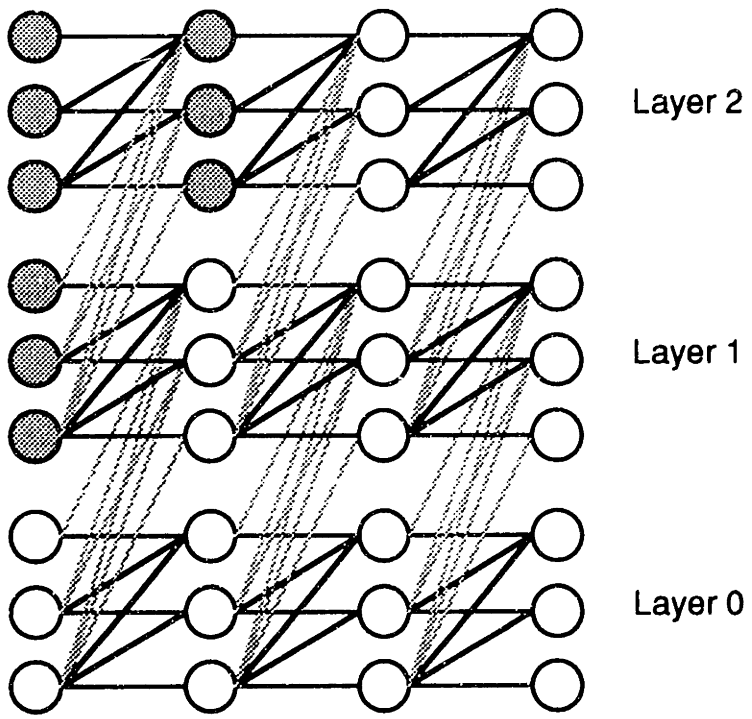


**Figure 3.34** Evaluation of a decrease in the number of overtime opportunities permitted

Of course, we have only considered the most simple type of constraint that can be accommodated. For example, if the length of the horizon were two weeks, one

could place one constraint on the number of overtime opportunities permitted in the first week, and a second constraint on the number of overtime opportunities permitted in the second week. Another possibility is only to constrain some of the overtime opportunities, e.g., to place a constraint on the number of times that overtime can be worked on the weekend. Limitless other possibilities exist. The ones that we have mentioned are reasonably straightforward.

An entirely different type of constraint that can be accommodated is a constraint on the amount of *time* that can be consumed, e.g., no more than eight hours of overtime per week. Such a constraint might arise from human resource issues, or might be a result of necessary machine downtime for activities such as preventative maintenance.

The general methodology for incorporating such a constraint is very similar to the one that we have just described. The layers now represent the amount of overtime (in terms of time) that has been consumed, instead of the number of times that overtime has been worked. An appropriate discretization must be chosen; suppose this is 15 minutes. If an overtime opportunity that is 60 minutes in length is undertaken, then the transition moves not one layer higher as before, but now four layers higher. Otherwise the algorithm is unchanged.

## 3.8 Steady-state Analysis

In this section we examine the impact of the finite horizon assumption that we have made in the preceding sections. We show empirically how the critical overtime levels are affected by increasing the length of the horizon, and briefly examine the factors that influence the rate at which the steady state is attained.

We use the base case described in Table 3.1 of Section 3.3. We first suppose that the horizon were twice as long, where the demand points and overtime opportunities in the second half of the horizon are identical to those in the first half. This doubles the number of demand points to 10. Figure 3.35 shows the impact of gradually increasing the length of the horizon (while adding demand points and overtime opportunities) on the critical overtime level at the first overtime opportunity. We see that the length of the horizon initially has a noticeable effect, although this effect rapidly diminishes and a steady-state is achieved. This indicates that the horizon length of 1000 minutes (16.66 hours) in the base case was too short for accurate decision making. However, we see that by the time there are 15 demand points (which corresponds to a horizon length of 50 hours), the steady-state critical overtime level is essentially achieved.

In Figure 3.36 we show the results of the same experiment, except now we consider the critical overtime level at the second demand point. The convergence to a steady-state value is slightly slower and the percent difference between the critical overtime level with five demand points and the steady-state value is slightly larger. Not surprisingly, we observe in general that the critical overtime levels further out in the horizon are more affected by the length of the horizon.

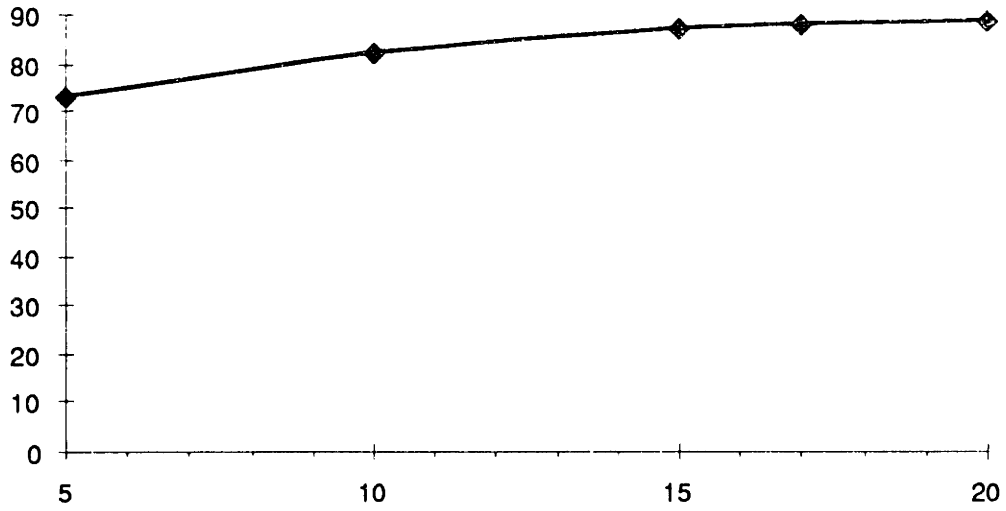


**Figure 3.35** Critical overtime level at the first decision point with varied number of demand points

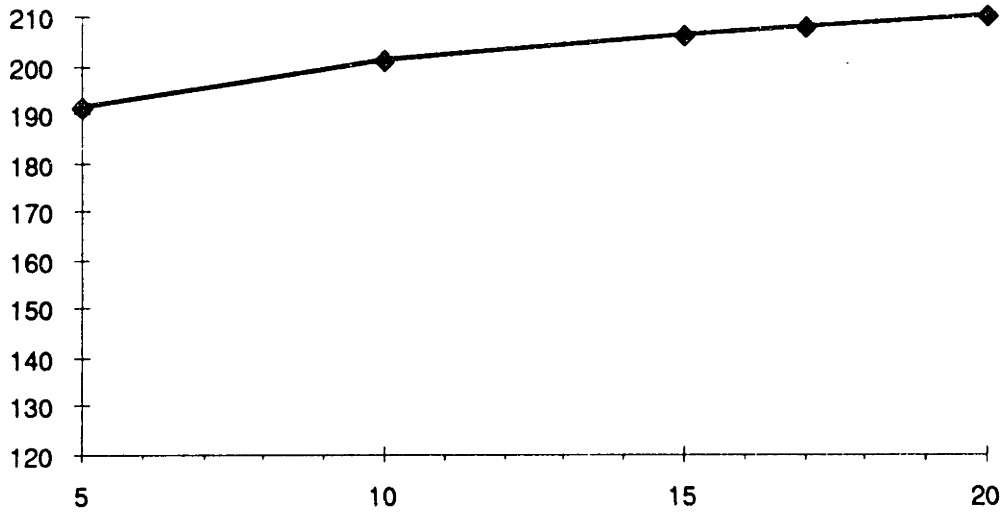


**Figure 3.36** Critical overtime level at the second decision point with varied number of demand points

We now wish to demonstrate that it is the length of the horizon and not the number of demand points that determines the deviation of the critical overtime levels from their steady-state values. This is done by taking the base case data and doubling the production batch size to 240, doubling the demand quantities to 120, placing the demand points 400 units apart, and placing the overtime opportunities 50 units before each demand point. Figure 3.37 shows the impact of varying the

number of demand points on the first critical overtime level. We see that steady state is essentially achieved with half of the number of demand points.

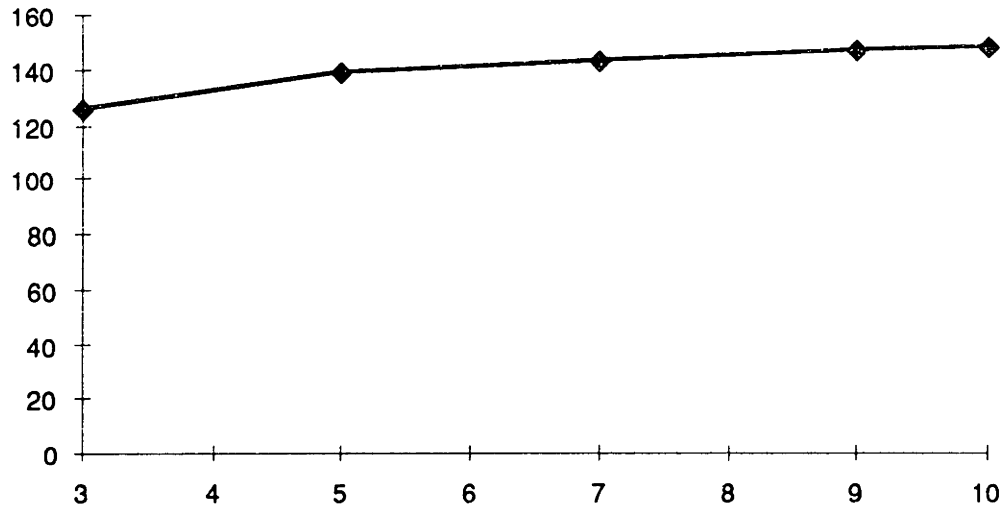


**Figure 3.37** Critical overtime level at the first decision point all data doubled

We conclude by mentioning the results of two experiments we have not included here. The first experiment was the opposite of the previous experiment, which showed that by halving the data values, the number of demand points required to reach steady state is twice as large. We have also observed that the rate at which convergence is achieved is accelerated when the cost of overtime is increased.

## 3.9 Rescheduling and sensitivity analysis

In this section we describe how to modify the dynamic programming algorithm discussed earlier in this chapter to obtain information regarding the sensitivity of the model to changes in the inputs.

The first and possibly most important question we address is how to use the algorithm to determine when and where rescheduling is beneficial. We consider a special type of rescheduling that we will call "cutting short". Cutting short means shifting a portion of a batch that was intended to be built now to the next batch that was planned for that same part. The rationale for this type of rescheduling is that many facilities will produce batches in excess of the immediate requirements, motivated by economic lot sizing concerns. As a result, the size of the batch can often be reduced without risk of stockout at the next demand point. Cutting short is therefore desirable when there is an imbalance between the intended production quantities and demand, either due to the non-stationarity of the demand, or because of above-average levels of downtime in the immediate past. One could therefore think of cutting short as rebalancing. In one particular plant, we have observed that real-world schedulers frequently employed cutting short as a method of alleviating short term capacity problems.

Figure 3.38 is a modification of Figure 3.5 that depicts what a cutting short strategy might look like in terms of its impact on the state space. Here we are cutting short by two units at time 8, resulting in the "removal" of the two shaded states, and shifting this additional work to time 16 (shown as dotted circles). Note that since the state variable $\tau$ measures cumulative output, the cumulative requirements for

each stage between times 8 and 16 (which is only time 10 in this example) decrease by two units.

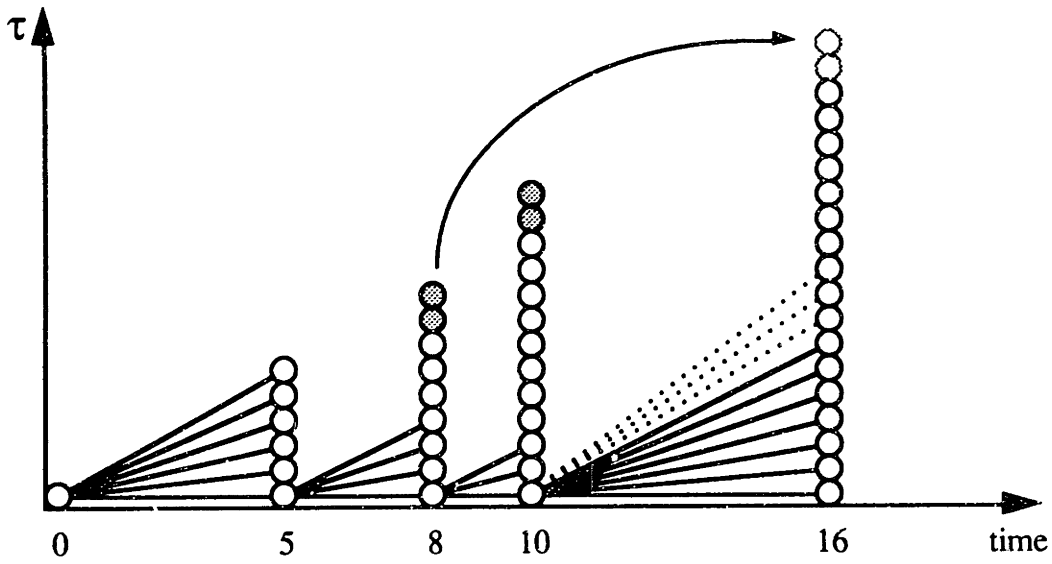


**Figure 3.38** State space representation with rescheduling

Computing this information is computationally inexpensive. The dynamic program must be re-solved using the modified production plan, but the transition probabilities do not need to be recomputed if the machine reliability is the same across all parts*. Recall that the computational effort is largely determined by the computation of the transition probabilities, so this extension requires very little additional effort. Determining the marginal benefits of shifting production between production runs requires one problem to be solved for each pair of production runs of the same part. Thus, if each part were produced twice over the horizon, there would be N different problems to be solved. Note that when these N different problems are solved by backward recursion, there is much replication of effort

* If machine reliability is part depdendent and we are only shifting one part (in order to evaluate marginal benefits), then although the transition probabilities will change as a result of cutting short, the effect will be very minor, and using the old transition probabilities will be an excellent approximation.

which can be avoided if an algorithm is coded that reuses the computational results from previous problems.

An alternative to computing marginal benefits is to restrict attention to cutting short in quantities equal to the lot size minus the sum of the demands before the next production run. Perhaps the ideal is to have an interactive tool that could be placed in the hands of a human scheduler, who could then explore a richer set of alternatives.

Although we have framed the above discussion in terms of "cutting short", all of the same ideas can be applied to the rescheduling strategy of "getting ahead", in which the target lot size for the next run of a part is increased to alleivate the load on the machine for the next production run. This may not be possible in some environments because of unavailability of excess raw materials. It may also be undesirable if raw materials are in limited supply and shared across parts. Lastly, it may be unappealing in certain factories where the culture is such that excess inventories are viewed as wasteful. Indeed, such a strategy is counter to quality concerns and the just-in-time philosophy. Nevertheless, overbuilding and carrying excess inventory during times in which there is excess capacity may be an alternative worth considering.

The information that is obtained from a cut short or get ahead sensitivity analysis can also be used to estimate the shadow prices of the lengths of the overtime opportunities. One less unit that must be produced (e.g., by cutting short) is nearly equivalent to the benefit of one additional time unit of overtime, multiplied by the machine's stand-alone availability. In this way, one can estimate the marginal

benefit of an additional unit of overtime. The marginal net benefit is obtained by subtracting the marginal cost of the additional time unit of overtime.

We now turn our attention to other sensitivity analyses. First, note that any change in the cost parameters (per time unit cost of overtime or per unit backorder cost) affects only the immediate costs at each stage, and therefore does not affect the transition probabilities. Once again, this means that the dynamic program can be re-solved very quickly as the cost parameters are varied.

The second observation that we make is that since we do not assume that there is any relationship between the demand quantities and production schedule, the demand quantities can be varied and affect only the immediate costs at each stage. Once again, this means that the dynamic program can be re-solved very quickly for different demand quantities. One potential benefit of this in addition to sensitivity analysis is to analyze different demand scenarios if the demand quantities are uncertain. This is discussed briefly in the next section.

## 3.10 Models with stochastic demand

The purpose of this section is to make some progress toward understanding the impact of the addition of demand variability to our models. Up to this point in the chapter we have assumed that the timing and quantity of demand is known with certainty over some short horizon. While this may be an acceptable assumption for some environments, for others it will be highly unrealistic. The addition of demand variability causes the model to become much more difficult. We therefore restrict our attention to two special cases. The first incorporates stochastic demands in the special case where only one part is produced on the machine. The second special case we consider assumes that the demand for all parts occurs at the same point in time, there is only one such point over the horizon, and the precise demand quantity is not known until the last moment.

### 3.10.1 Single part

In the case where there is only a single part built on the machine, the state variable $\tau$ can be replaced with a new state variable that represents the inventory level of the part. We will effectively abandon the notion of a schedule, assuming that the machine produces at full capacity, so that the only control available is the quantity of overtime to purchase. We retain the same stages as the previous formulation, one stage for each demand point and one stage for each overtime opportunity, except now we insert an additional stage after each stage that represents a demand point. Like the stages that represent the demand points, there are no decisions made at these additional stages. The transitions between a stage that represents a demand point and the new stage that we have inserted after it are governed by the demand distribution. Since the state variable $\tau$ has been replaced with a state variable

representing inventory level, the transitions between these two stages represent the fulfillment of demand.

The cost of stockouts are assessed after the demand is filled. Within this structure, we can capture either lost sales or backorders as a result of stockout. In either case we require the state space to include negative values for the inventory level. In the lost sales case, we treat the negative inventory valued states as though they were state zero (no inventory on hand) when computing the transition probabilities to the next stage. The negative states are necessary to help assess the appropriate lost sales costs. In the backorder case, the negative states have a physical interpretation that affects the transition probabilities to the next stage. Note that we can also assess a per unit inventory holding cost in the positive states if desired.

The only remaining detail is how to value inventory (or how to penalize a backlog) at the end of the horizon. As before, we can enter an arbitrary terminal costs that are a function of the state.

The model is then solved with a backwards dynamic programming recursion similar to the one described earlier in this chapter. The algorithm for computing expected cost to go and optimal overtime decisions is unchanged.

If inventory holding costs are assessed, it may be desirable to insert additional stages where the decision maker has the option to turn the machine off and stop producing. The state of the machine (on or off) could be carried as an additional state variable, and would approximately double the computational effort required.

The problem is much more difficult when there are multiple parts produced on the same machine, each with stochastic demand. An extension of the above model to the case of multiple parts would require an additional state variable for each additional part, plus the state variable $\tau$ to keep track of progress relative to the schedule. The addition of state variables increases the complexity of the dynamic programming algorithm exponentially, and therefore rapidly becomes unrealistic. We do not explore the well developed theory of state space reduction; the interested reader is referred to Larson (1968). We do note, however, that the model discussed earlier in this chapter could be used to evaluate a number of different demand scenarios. Although this is no substitute for a model that can accommodate stochastic demands, it may be of great assistance in helping a decision maker to understand the impact of demand uncertainty on the optimal overtime decisions.

### 3.10.2 Single demand point

The model of this subsection will differ from the model discussed at the beginning of the chapter in three fundamental ways. First, we assume that there is only a single demand point for all parts, and that it occurs at the end of the horizon. The second major difference is that the demand for each part is now a random variable with a known distribution function, where the uncertainty in the demand quantity is not resolved until the demand point. Lastly, we assume a fixed production sequence as before, but we now find optimal production quantities, and later, overtime levels as well.

Initially we restrict ourselves to finding optimal set of production quantities. An alternative approach might be to find an optimal set of run times, where the available machine time is partitioned among the different parts. Although each of these policy types has its own merit, the best policy is a mixture of the two: a

dynamic policy in which the decision to stop production is based on both the realized output of the machine *and* the remaining time for production. We briefly explore an approximate dynamic policy of this type at the end of this subsection.

Brief literature review

The classic single demand point model is the "newsboy" model (Lee and Nahmias, 1993), which has been and continues to be extensively studied. Some extensions include multiple items (Evans, 1967; Smith et al., 1980), uncertain replenishment (Rose, 1992), and multiple time periods for production (Bitran et. al, 1986; Matsuo, 1990). However, each of these models is fundamentally different from the one that we consider here. For example, Rose assumes that demand is deterministic, none of the authors except Rose address machine unreliability, and there does not appear to be any paper that addresses the option to purchase additional capacity (overtime).

Formulation

The mathematical structure of our model will closely parallel that of the classic newsboy model, which we now briefly describe. Let x denote the current inventory level, c the unit purchase price, h the cost per unit of inventory remaining at the end of the period, p the unit shortage cost and g(·) the PDF of demand. The problem is then to choose an order-up-to quantity y to minimize the expected purchase, holding and shortage costs. Mathematically, we can state the problem as

$$C^*(x) = \min_{y \geq x} c\,(y-x) + p \int_y^\infty (t-y)\, g(t)\, dt + h \int_0^y (y-t)\, g(t)\, dt.$$

The problem is solved by finding the value of y such that $\partial C(x)/\partial y$ is zero. To find this partial derivative, we need to employ Leibnitz's rule

$$\frac{\partial}{\partial y}\int_{p(y)}^{q(y)} f(x,y)\,dx = \int_{p(y)}^{q(y)} \frac{\partial f(x,y)}{\partial y}\,dx + \frac{\partial q(y)}{\partial y} f(q(y),y) - \frac{\partial p(y)}{\partial y} f(p(y),y)$$

(Beyer, 1987). We will use this extensively in our analysis. From this rule it is easy to see that the optimal solution $y^*$ to the newsboy model occurs at the point where $G(y^*) = (p - c) / (p + h)$, unless this implies $y^* < x$, in which case it is optimal not to order.

We now extend this basic single part model to our multiple part, unreliable production process model, for now ignoring overtime opportunities. The problem is to find the optimal order-up-to levels to minimize the sum of purchasing, holding and shortage costs over all parts. Let y, x, c, p, h and $g(\cdot)$ retain the same meanings as above, except now we add a subscript i, for each part $i = 1, \ldots, N$. We assume without loss of generality that the parts are indexed in the order in which they will be produced. Denote the setup time for each part as $S_i$. If we are already setup to produce part 1, then we set $S_1 = 0$. We assume for simplicity that each part is produced at the same rate when the machine is working ($P_i = 1 \; \forall \; i$).

Let T denote the amount of time available for production, and the time available after setups as $T_i = T - S_1 - \cdots - S_i$. As before, the CDF F(t; T) is the probability that in T units of time, the cumulative output of the machine is at most t parts. This distribution was discussed in detail in Chapter 2, although our results will not depend on the form of this distribution.

We can now write the problem as

$$C^*(x) = \min_{y_1 \geq x_1, \ldots, y_N \geq x_N} C(y,x)$$

where $C(y,x) = \sum_{i=1}^{N} C_i(y,x)$,

$$\begin{aligned}
\text{and } C_i(y,x) = {} & c_i \int_{x_i}^{y_i} (t - x_i)\, f\left(\sum_{j=1}^{i-1} y_j - x_j + t - x_i;\ T_i\right) dt \\
& + c_i(y_i - x_i)\, \bar{F}\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right) \\
& + p_i \int_{x_i}^{y_i} \int_{u}^{\infty} (t - u)\, g_i(t)\, dt\, f\left(\sum_{j=1}^{i-1} y_j - x_j + u - x_i;\ T_i\right) du \\
& + p_i\, \bar{F}\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right) \int_{y_i}^{\infty} (t - y_i)\, g_i(t)\, dt \\
& + p_i\, F\left(\sum_{j=1}^{i-1} y_j - x_j;\ T_i\right) \int_{x_i}^{\infty} (t - x_i)\, g_i(t)\, dt \\
& + h_i \int_{x_i}^{y_i} \int_{0}^{u} (u - t)\, g_i(t)\, dt\, f\left(\sum_{j=1}^{i-1} y_j - x_j + u - x_i;\ T_i\right) du \\
& + h_i\, \bar{F}\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right) \int_{0}^{y_i} (y_i - t)\, g_i(t)\, dt \\
& + h_i\, F\left(\sum_{j=1}^{i-1} y_j - x_j;\ T_i\right) \int_{0}^{x_i} (x_i - t)\, g_i(t)\, dt
\end{aligned}$$

where the summations from 1 to i-1 are taken to be null at i = 1.

Each $C_i(y, x)$ represents the expected purchasing, holding and shortage costs incurred for part i given a set of order-up-to levels $y_i$. We have written $C_i(y, x)$ as the sum of eight terms. The first two terms express the expected purchasing cost, where the first term is the expected purchasing cost if the realized uptime of the machine is such that the available supply of the $i^{th}$ part is between the values of 0 and $y_i$-$x_i$ and the second term is the expected purchasing cost if the realized uptime of the machine is

such that the available supply of the $i^{th}$ part is the desired value $y_i$-$x_i$. There is no purchasing cost if the available supply of the $i^{th}$ part is not greater than zero. The next three terms represent the expected shortage costs. The first of these terms is the expected shortage cost if the available supply is between 0 and $y_i$-$x_i$, the second term is the expected shortage cost if the available supply is $y_i$-$x_i$, and the third is the expected shortage cost if the available supply is 0. Similarly, the last three terms represent the expected holding costs, where the first of these terms is the expected holding cost if the available supply of the $i^{th}$ part is between 0 and $y_i$-$x_i$, the second term is the expected holding cost if the available supply is $y_i$-$x_i$, and the third is the expected holding cost if the available supply is 0.

Properties of the objective function

To obtain the optimal order quantities we wish to show that the total cost function is convex with respect to the order quantities. If this is so, we can find minimizing order quantities by finding where the partial derivative of the total cost function is zero. We now discuss each of these properties in turn.

We begin with the first order optimality condition for $y_N$, using Leibnitz's rule to obtain

$$\frac{\partial}{\partial y_N} C(y,x) = \left(c_N - p_N \overline{G}_N(y_N) + h_N \overline{G}_N(y_N)\right) \overline{F}\left(\sum_{j=1}^{N} y_j - x_j;\ T_N\right).$$

When written as the product of two terms as we have done, this derivative has a nice interpretation. The first term is the derivative of the cost function for the classical newsboy problem. This term is multiplied by the probability that we can complete our production plan in the time available.

Because of this structure, the first order optimality condition is reduced to $G_N(y_N) = (p_N - c_N) / (p_N + h_N)$, the solution to the classical newsboy problem. As before, it is easy to show that if this implies $y_N < x_N$, then the optimal $y_N$ is $x_N$. The optimal $y_N$ should not be dependent on the other $y_i$, because once we have produced parts 1, ..., N-1, all we can do is try to minimize the costs for part N. The optimal $y_N$ should not be dependent on the machine's reliability, because the best thing to do is attempt to achieve the optimal order-up-to quantity exactly.

We now turn to the more difficult task of taking the partial derivative of the total cost function C(y, x) with respect to $y_i$ for $i < N$. After simplification, the result is

$$(1)\quad \frac{\partial}{\partial y_i} C(y,x) = \overline{F}\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right)\left(c_i - p_i\overline{G}_i(y_i) + h_i G_i(y_i)\right) + \sum_{k=i+1}^{N} (p_k - c_k)\left[F\left(\sum_{j=1}^{k} y_j - x_j;\ T_k\right) - F\left(\sum_{j=1}^{k-1} y_j - x_j;\ T_k\right)\right] - \sum_{k=i+1}^{N} (p_k + h_k) \int_{x_k}^{y_k} G_k(u)\ f\left(\sum_{j=1}^{k-1} y_j - x_j + u - x_k;\ T_k\right) du$$

where the summations from i+1 to N are taken to be null at i = N. This expression is easier to interpret if we rewrite it as

$$\frac{\partial}{\partial y_i} C(y,x) = \overline{F}\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right)\left(c_i - p_i\overline{G}_i(y_i) + h_i G_i(y_i)\right) + \sum_{k=i+1}^{N} (p_k - c_k)\left[F\left(\sum_{j=1}^{k} y_j - x_j;\ T_k\right) - F\left(\sum_{j=1}^{k-1} y_j - x_j;\ T_k\right)\right] + \sum_{k=i+1}^{N} (p_k + h_k) \int_{x_k}^{y_k} \overline{G}_k(u)\ f\left(\sum_{j=1}^{k-1} y_j - x_j + u - x_k;\ T_k\right) du$$

$$- \sum_{k=i+1}^{N} (p_k + h_k) \int_{x_k}^{y_k} f\left(\sum_{j=1}^{k-1} y_j - x_j + u - x_k;\ T_k\right) du,$$

and then simplify to obtain

$$\begin{aligned}\frac{\partial}{\partial y_i} C(y,x) \;=\; & \bar{F}\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right)\left(c_i - p_i \bar{G}_i(y_i) + h_i G_i(y_i)\right) \\ & - \sum_{k=i+1}^{N} (c_k + h_k)\left[F\left(\sum_{j=1}^{k} y_j - x_j;\ T_k\right) - F\left(\sum_{j=1}^{k-1} y_j - x_j;\ T_k\right)\right] \\ & + \sum_{k=i+1}^{N} (p_k + h_k) \int_{x_k}^{y_k} \bar{G}_k(u)\, f\left(\sum_{j=1}^{k-1} y_j - x_j + u - x_k;\ T_k\right) du.\end{aligned}$$

The first term is analogous to $\partial C(y, x)/\partial y_N$ discussed above. The second two terms give the impact of the choice of $y_i$ on the parts k = i+1, ..., N. The first of these terms represents the marginal cost of machine time. The expression in square brackets is the probability that machine output is insufficient to produce up to $y_k$ but sufficient to start production of part k. As this probability increases, total cost decreases at rate $c_k + h_k$, assuming that the units built are not sold. The final term is the marginal cost of lost sales. The integral represents the expected sales given that machine output is greater than zero but less than $y_k$. As this increases, shortage costs are accrued at a rate $p_k$ and holding costs, which have already been charged in the second term, are avoided at a rate $h_k$.

It can be seen from this first order condition that as T tends to infinity, the optimal $y_i$ each approach their "newsboy point" $y_i^N$, that is, the point where $G(y_i) = (p_i - c_i) / (p_i + h_i)$. It should also be evident that the optimal $y_i$ are never greater than $y_i^N$, their respective newsboy points. We now argue this formally by induction. We have already shown that the optimal $y_N$ is $y_N^N$, the newsboy point for part N. Suppose that

we have shown that the optimal $y_k$ are not greater than $y_k^N$ for k = i+1, ..., N. We will now show that the optimal $y_i$ is also less than or equal to $y_i^N$. We first require the following result:

$$\frac{\partial}{\partial y_i} C(y, x) = \bar{F}\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right)\left(c_i - p_i\bar{G}_i(y_i) + h_iG_i(y_i)\right)$$
$$+ \sum_{k=i+1}^{N} (p_k - c_k)\left[F\left(\sum_{j=1}^{k} y_j - x_j;\ T_k\right) - F\left(\sum_{j=1}^{k-1} y_j - x_j;\ T_k\right)\right]$$
$$- \sum_{k=i+1}^{N} (p_k + h_k) \int_{x_k}^{y_k} G_k(u)\ f\left(\sum_{j=1}^{k-1} y_j - x_j + u - x_k;\ T_k\right) du$$

(from equation (1))

$$\geq \bar{F}\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right)\left(c_i - p_i\bar{G}_i(y_i) + h_iG_i(y_i)\right)$$
$$+ \sum_{k=i+1}^{N} (p_k - c_k)\left[F\left(\sum_{j=1}^{k} y_j - x_j;\ T_k\right) - F\left(\sum_{j=1}^{k-1} y_j - x_j;\ T_k\right)\right]$$
$$- \sum_{k=i+1}^{N} (p_k + h_k)\ G_k(y_k) \int_{x_k}^{y_k} f\left(\sum_{j=1}^{k-1} y_j - x_j + u - x_k;\ T_k\right) du$$

(because $G_k(\cdot)$ is non-decreasing)

$$= \bar{F}\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right)\left(c_i - p_i\bar{G}_i(y_i) + h_iG_i(y_i)\right)$$
$$- \sum_{k=i+1}^{N} \left(c_k - p_k\bar{G}_k(y_k) + h_kG_k(y_k)\right)\left[F\left(\sum_{j=1}^{k} y_j - x_j;\ T_k\right) - F\left(\sum_{j=1}^{k-1} y_j - x_j;\ T_k\right)\right]$$
$$\geq \bar{F}\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right)\left(c_i - p_i\bar{G}_i(y_i) + h_iG_i(y_i)\right)$$

(because $G_k(y_k) \leq (p_k - c_k) / (p_k + h_k)$ for k = i+1, ..., N, by the induction hypothesis).

Using this result, it immediately follows that for any $y_i > y_i^N$, $\partial C(y, x)/\partial y_i$ is positive. Therefore, if $x_i < y_i^N$, the optimal $y_i$ lies between $x_i$ and $y_i^N$. If $x_i \geq y_i^N$, then it is optimal not to produce (the optimal $y_i$ equals $x_i$).

We now show that if $x_k \le y_k \le y_k^N$ for k = i, i+1, ..., N, then $\partial^2 C(y, x)/\partial y_i^2$ is non-negative. To show this, we once again use Leibnitz's rule to take the second partial derivative with respect to $y_i$ to obtain

$$\begin{aligned}\frac{\partial^2}{\partial y_i^2}C(y,x) &= \left[\overline{F}\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right)\left(p_i g_i(y_i) + h_i g_i(y_i)\right)\right] \\ &+ \left[-f\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right)\left(c_i - p_i \overline{G}_i(y_i) + h_i G_i(y_i)\right)\right] \\ &+ \left[\sum_{k=i+1}^{N}\left\{(p_k - c_k)\left(f\left(\sum_{j=1}^{k} y_j - x_j;\ T_k\right) - f\left(\sum_{j=1}^{k-1} y_j - x_j;\ T_k\right)\right)\right.\right. \\ &\left.\left. - (p_k + h_k)\int_{x_k}^{y_k} G_k(u)\,\frac{\partial}{\partial y_i}\left\{f\left(\sum_{j=1}^{k-1} y_j - x_j + u - x_k;\ T_k\right)\right\}\,du\right\}\right].\end{aligned}$$

We now show that this second partial derivative is non-negative. We have written the second partial derivative as the sum of three (square bracketed) terms. The first term can be seen to be non-negative by inspection. The second square bracketed term is non-negative if $-c_i + p_i\overline{G}_i(y_i) - h_i G_i(y_i)$ is non-negative, which is true if $G_i(y_i) \le (p_i - c_i) / (p_i + h_i)$, which is always true for $y_i < y_i^N$. Showing that the third bracketed term is non-negative is slightly more difficult. We note that for each k,

$$\begin{aligned}&(p_k - c_k)\left(f\left(\sum_{j=1}^{k} y_j - x_j;\ T_k\right) - f\left(\sum_{j=1}^{k-1} y_j - x_j;\ T_k\right)\right) \\ &\quad - (p_k + h_k)\int_{x_k}^{y_k} G_k(u)\,\frac{\partial}{\partial y_i}\left\{f\left(\sum_{j=1}^{k-1} y_j - x_j + u - x_k;\ T_k\right)\right\}\,du \\ &\ge (p_k - c_k)\left(f\left(\sum_{j=1}^{k} y_j - x_j;\ T_k\right) - f\left(\sum_{j=1}^{k-1} y_j - x_j;\ T_k\right)\right) \\ &\quad - (p_k + h_k)\,G_k(y_k)\int_{x_k}^{y_k} \frac{\partial}{\partial y_i}\left\{f\left(\sum_{j=1}^{k-1} y_j - x_j + u - x_k;\ T_k\right)\right\}\,du\end{aligned}$$

(because $G_k(\cdot)$ is non-decreasing)

$$\geq (p_k - c_k)\left(f\left(\sum_{j=1}^{k} y_j - x_j;\ T_k\right) - f\left(\sum_{j=1}^{k-1} y_j - x_j;\ T_k\right)\right)$$

$$- (p_k + h_k)\frac{(p_k - c_k)}{(p_k + h_k)}\int_{x_k}^{y_k}\frac{\partial}{\partial y_i}\left\{f\left(\sum_{j=1}^{k-1} y_j - x_j + u - x_k;\ T_k\right)\right\}du$$

(because $G_k(y_k) \leq (p_k - c_k) / (p_k + h_k)$)

$$= (p_k - c_k)\left(f\left(\sum_{j=1}^{k} y_j - x_j;\ T_k\right) - f\left(\sum_{j=1}^{k-1} y_j - x_j;\ T_k\right)\right)$$

$$- (p_k - c_k)\left(f\left(\sum_{j=1}^{k} y_j - x_j;\ T_k\right) - f\left(\sum_{j=1}^{k-1} y_j - x_j;\ T_k\right)\right)$$

$= 0$. Q.E.D.

Given the other $y_j$, $j \neq i$, this result allows us to find the optimal $y_i$ by determining if $\exists\, y_i \in [x_i, y_i^N]$ such that $\partial C(y, x)/\partial y_i = 0$. If such a $y_i$ exists then it is optimal, otherwise, the optimal policy is not to order. Since $\partial C(y, x)/\partial y_i$ is a non-decreasing function of $y_i$ over the range $[x_i, y_i^N]$ when $y_k \leq y_k^N$ for $k = i+1, \ldots, N$, the optimal $y_i$ can be found by simple binary search.

Given the above results, after we have found $y_N$ we can find the other $y_i$ by solving the above problem as a N-1 dimensional unconstrained minimization problem on the interval $x_i \leq y_i \leq y_i^N$, $i = 1, \ldots, N-1$. For an excellent discussion of algorithms to solve such problems, see Bazaraa et al. (1993). An alternative approach is presented on the next few pages.

Solution algorithm

The difficulty in finding the optimal production quantities is that the first order condition tells us that N-1 of the $y_i$ are mutually dependent. We now describe a solution procedure that exploits the special structure of these dependencies. In particular, consider the difference

$$
\begin{aligned}
\hat{C}_{i+1} &= \frac{\partial C(y,x)}{\partial y_{i+1}} - \frac{\partial C(y,x)}{\partial y_i} \\
&= \overline{F}\left(\sum_{j=1}^{i+1} y_j - x_j;\ T_{i+1}\right)\left(c_{i+1} - p_{i+1}\overline{G}_{i+1}(y_{i+1}) + h_{i+1}G_{i+1}(y_{i+1})\right) \\
&\quad - \overline{F}\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right)\left(c_i - p_i\overline{G}_i(y_i) + h_iG_i(y_i)\right) \\
&\quad - (p_{i+1} - c_{i+1})\left[F\left(\sum_{j=1}^{i+1} y_j - x_j;\ T_{i+1}\right) - F\left(\sum_{j=1}^{i} y_j - x_j;\ T_{i+1}\right)\right] \\
&\quad + (p_{i+1} + h_{i+1})\int_{x_{i+1}}^{y_{i+1}} G_{i+1}(u)\ f\left(\sum_{j=1}^{i} y_j - x_j + u - x_{i+1};\ T_{i+1}\right) du.
\end{aligned}
$$

Note that if $y_i$ is optimal, $\partial C(y, x)/\partial y_i$ is zero, so that $\hat{C}_{i+1} = \partial C(y, x)/\partial y_{i+1}$. The reason that this is significant is because $\hat{C}_{i+1}$ is a function only of $y_1, \ldots, y_i$. Therefore if the optimal $y_1$ is known then $\hat{C}_2$ can be used to find the optimal $y_2$, and then $\hat{C}_3$ can be used to find the optimal $y_3$, and so forth.

Since the optimal $y_1$ is not known, we must use a search technique to find it. We now prove three important properties that will be helpful in this regard.

Let the production quantities that result from the above procedure be denoted by $\hat{y}_i$. We first show that $\hat{y}_N = y_N^N$ iff $\partial C(y, x)/\partial y_1 = 0$. Observe that $\hat{C}_N$ is exactly equal to $\partial C(y, x)/\partial y_N - \partial C(y, x)/\partial y_{N-1}$, and thus $\hat{y}_N = y_N^N$ iff $\partial C(y, x)/\partial y_{N-1} = 0$. Further, for any i, $\hat{C}_{i+1} = \partial C(y, x)/\partial y_{i+1}$ iff $\partial C(y, x)/\partial y_i = 0$. Therefore, $\hat{y}_N = y_N^N$ iff $\partial C(y, x)/\partial y_1 = 0$.

The second property is that if the guess for the optimal value of $y_1$ is too large, $\hat{y}_N > y_N^N$. We have shown above that if $x_k \le y_k \le y_i^N$ for $k = i, i+1, \ldots, N$, then $\partial^2 C(y, x)/\partial y_i^2 \ge 0$. Accordingly, if the guess for the optimal value of $y_1$ is too large, $\partial C(y, x)/\partial y_1 > 0$,

so that in order for $\hat{C}_2 = 0$, $\hat{y}_2$ must be chosen such that $\partial C(y, x)/\partial y_2 > 0$, so that $\hat{y}_2$ will be greater than the optimal $y_2$. Repeating this argument, we see that each $\hat{y}_i$ will be greater than the optimal $y_i$, and thus $\hat{y}_N > y_N^N$. By analogous reasoning we can conclude that if the guess for the optimal value of $y_1$ is too small, $\hat{y}_N < y_N^N$.

The third and final property that we wish to show is that $\hat{C}_{i+1}$ is an increasing function of $y_{i+1}$. This property is particularly important, as it allows us to find $\hat{y}_{i+1}$ by simple binary search. To prove this, we take the partial derivative of $\hat{C}_{i+1}$ with respect to $y_{i+1}$ and simplify to obain

$$\frac{\partial}{\partial y_{i+1}}\hat{C}_{i+1} = \overline{F}\left(\sum_{j=1}^{i+1} y_j - x_j;\ T_{i+1}\right)\left(p_{i+1}g_{i+1}(y_{i+1}) + h_{i+1}g_{i+1}(y_{i+1})\right)$$

which is clearly non-negative since each term is non-negative, and thus the result is proven.

Using these properties, we are now ready to state the following

***Algorithm:***

1. Pre-processing. Compute the $y_i^N$. If any $x_i \geq y_i^N$ then the optimal $\hat{y}_i = x_i$ and it is optimal not to produce this part. Remove all such parts from the list of parts to be produced over the horizon.
2. Initialization. Set $\hat{y}_1 = y_1^N$. Set $U = y_1^N$ and $L = x_1$.
3. Main loop. For each $i = 2, \ldots, N$, find the $\hat{y}_i$ such that $\hat{C}_i = 0$. If any $\hat{y}_i > y_i^N$ then $\hat{y}_1$ is too large. Set $U = \hat{y}_1$, $\hat{y}_1 = (U + L) / 2$, and repeat Step 3.
4. Optimality test. If $| \hat{y}_N - y_N^N | < \varepsilon$ then the $\hat{y}_i$ are optimal. Stop.

5. Adjustment step. If $\hat{y}_N > y_N^N$ then $\hat{y}_1$ is too large. Set $U = \hat{y}_1$, $\hat{y}_1 = (U + L) / 2$, and go to Step 3. If $\hat{y}_N < y_N^N$ then $\hat{y}_1$ is too small. Set $L = \hat{y}_1$, $\hat{y}_1 = (U + L) / 2$, and go to Step 3.

The algorithm essentially performs a binary search on the guess for the optimal $y_1$ by maintaining an upper and lower bound (U and L) on the optimal value. The algorithm terminates when the current value of $\hat{y}_N$ is within some small positive $\varepsilon$ of $y_N^N$.

Because the properties that we have proven above are valid only if $x_i \le y_i \le y_i^N$ for i = 1, ..., N, we must take care to ensure that this remains true throughout the algorithm. We perform the test in Step 2 to ensure that we do not proceed if any $y_i > y_i^N$. We set $L = x_1$ so that $\hat{y}_1 \ge x_1$. Lastly, in a pre-processing step we remove a part i from consideration if $x_i > y_i^N$. We can do this because, for any such part, the optimal $\hat{y}_i$ is $x_i$, and it is thus optimal not to produce that part. Since the part would not be produced, it has no effect on the other parts.

Dynamic rescheduling

In the development above we have discussed how to determine a set of production quantities to minimize expected total cost. Of course, as the plan is implemented, the reliability of the machine may be much higher or much lower than expected. As a result, if we were given the opportunity to do so, we might adjust the production plan based on what actually happens as time rolls forward.

We now consider how to dynamically update the optimal policy based on the realized output of the machine. One approach would be to repeatedly solve the model as fast as possible with constantly updated information from the factory floor.

Such an approach places a great demand on both computational resources and information systems. We instead propose a simpler method that would allow someone on the shop floor to determine when to stop production of the current part based on the current inventory level. We now describe the optimal dynamic policy for the current part, assuming that the decision maker will follow a static optimal policy for all subsequent parts. In this way, the dynamic solution obtained is only an approximation to the true dynamic optimal policy.

Suppose it is currently time zero, and for any particular future point in time we would like to determine the amount of completed production at or above which it is optimal to stop producing the current part and switch to the next part. We find these critical inventory levels as follows. We feed inputs into the model as if it is now some future point in time. The model is then used to find the optimal production plan as we vary the inventory level of part 1. We have found the critical inventory level when we have found the lowest inventory level such that the optimal decision is not to produce. We then know that at this future point in time if we are at or above this level then we should stop producing part 1. In terms of the mathematical model, this equates to finding the smallest $x_1$ such that the optimal $y_1$ is equal to $x_1$. If we can do this, then we can trace out a curve that shows this critical inventory level over time. The optimal dynamic operating policy is therefore to produce until the inventory level crosses the curve. Once this happens and production is switched to the next part, the model should be solved again to find the critical inventory level as a function of time for the next part.

Two important details have been omitted from the above discussion. The first involves the existence of such a critical inventory level. Recall that we are only interested in the *lowest* $x_1$ such that at the optimum $y_1 - x_1 = 0$, so the only question

we must answer is whether or not such an $x_1$ exists. But this is clearly so, since if we set $x_1 = G_1^{-1}((p_1 - c_1) / (p_1 + h_1))$, we know $y_1 \leq G_1^{-1}((p_1 - c_1) / (p_1 + h_1))$ and since we must constrain $y_1$ to be at least $x_1$, $y_1 = x_1$.

The second detail that needs to be resolved is how to compute the critical inventory level for a future point in time. We now describe a method based on the solution procedure outlined above. Suppose the point in time is $t_1$ and the current time is $t_0$. Then the first step is to update the horizon length by replacing T with $T - (t_1 - t_0)$, set $x_1 = 0$, and then solve for the optimal production quantities. We then search over $x_1$, at each iteration finding the optimal $y_i$, until we identify the *lowest* $x_1$ such that at the optimum, $y_1 = x_1$.

Impact of overtime opportunities

In the development above we purposely omitted any discussion of how to make optimal overtime decisions. Suppose now that there are $p = 1, \ldots, N_{OT}$ opportunities over the horizon to run overtime, and for simplicity assume that they are each of duration OT at cost $c_p$. In the development above we computed optimal production quantities ignoring overtime opportunities. This is equivalent to assuming that we choose not to run overtime, and the resulting expected cost is the expected cost of this strategy.

Suppose instead that we decide that we are going to run overtime once. To evaluate the expected cost of this strategy we simply replace T by T + OT and find the optimal production quantities to compute the minimum expected cost, and then add $c_p$. Note that unlike the previous models in this chapter, it does not matter *when* we run overtime, since all overtime opportunities occur before the demand point. Because of this simple fact, we can find the optimal policy by finding the optimal

production quantities $N_{OT}$ + 1 times, with T taking on the values T, T + OT, T + 2 OT, ..., T + $N_{OT}$ OT.

Of course, we expect that the total cost function will be convex in OT if overtime costs are convex in OT. If this is true, then the optimal overtime level can be found by a more efficient search procedure. We leave this as a conjecture for now.

Extension to different machine speeds

For notational convenience, up to this point we have ignored the possibility that the machine operates at different speeds when producing different parts. If the speeds are different, then the requirements on the machine need to be expressed in common units, such as time, instead of parts. This can be accommodated easily, replacing all expressions such as

$$F\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right) \text{ and } f\left(\sum_{j=1}^{i} y_j - x_j;\ T_i\right)$$

with

$$F\left(\sum_{j=1}^{i} \frac{y_j - x_j}{P_j};\ T_i\right) \text{ and } f\left(\sum_{j=1}^{i} \frac{y_j - x_j}{P_j};\ T_i\right),$$

where $P_j$ is the speed at which the machine produces part j when it is working. Our solution procedure for finding the optimal $y_i$ is also unchanged.

## References for Chapter 3


Adshead, N. S. and D. H. R. Price. "Overtime Decision Rule Experiments With a Model of a Real Shop". European Journal of Operational Research, 39(3), pp. 274-283, 1989.

Bazaraa, Mokhtar S., Hanif D. Sherali and C. M. Shetty. Nonlinear Programming: Theory and Algorithms, 2nd edition. New York: John Wiley and Sons, Inc., 1993.

Bellman, Richard E. Dynamic Programming. Princeton, NJ: Princeton University Press, 1957.

Bellman, Richard E. and Stuart E. Dreyfus. Applied Dynamic Programming. Princeton, NJ: Princeton University Press, 1962.

Bertsekas, Dimitri P. Dynamic Programming: Deterministic and Stochastic Models. Englewood Cliffs, NJ: Prentice-Hall, 1987.

Beyer, William H., ed. CRC Handbook of Mathematical Sciences, 6th edition. Boca Raton, Florida: CRC Press, 1987.

Birge, J., J. B. G. Frenk, J. Mittenthal and A. H. G. Rinnooy Kan. "Single-Machine Scheduling Subject to Stochastic Breakdowns". Naval Research Logistics, 37, pp. 661-677, 1990.

Birge, John R. and Kevin D. Glazebrook. "Assessing the Effects of Machine Breakdowns in Stochastic Scheduling". Operations Research Letters, 7(6), pp. 267-271, 1988.

Bitran, Gabriel R., Elizabeth A. Haas and Hirofumi Matsuo. "Production Planning of Style Goods with High Setup Costs and Forecast Revisions". Operations Research, 34(2), pp. 226-236, 1986.

Bitran, Gabriel R. and Devanath Tirupati. "Approximations for Networks of Queues with Overtime". Management Science, 37(3), pp. 282-300, 1991.

Brooke, Lindsay. "Stamping the Ram". Automotive Industries, September 1993, pp. 77-78.

Denardo, Eric V. Dynamic Programming: Models and Applications. Englewood Cliffs, NJ: Prentice-Hall, 1982.

Evans, Richard V. "Inventory Control of a Multiproduct System with a Limited Production Resource". Naval Research Logistics Quarterly, 14(2), pp. 173-184, 1967.

Federgruen, Awi and Linda Green. "Queueing Systems with Service Interruptions." Operations Research, 34(5), pp. 752-768, 1986.

Federgruen, Awi and Linda Green. "Queueing Systems with Service Interruptions II". Naval Research Logistics, 35(3), pp. 345-358, 1989.

Gelders, L. and P. R. Kleindorfer. "Coordinating Aggregate and Detailed Scheduling Decisions in the One-Machine Job Shop: Part I. Theory". Operations Research, 22(1), pp. 46-60, 1974.

Gelders, L. and P. R. Kleindorfer. "Coordinating Aggregate and Detailed Scheduling Decisions in the One-Machine Job Shop: II – Computation and Structure". Operations Research, 23(2), pp. 312-324, 1975.

Gittins, J. C. "Bandit Processes and Dynamic Allocation Indices". Journal of the Royal Statistical Society, Series B, 41, pp. 148-177, 1979.

Glazebrook, K. D. "Scheduling Stochastic Jobs on a Single Machine Subject to Breakdowns". Naval Research Logistics Quarterly, 31, pp. 251-264, 1984.

Groenevelt, Harry, Liliane Pintelon and Abraham Seidmann. "Production Batching with Machine Breakdowns and Safety Stocks". Operations Research, 40(5), pp. 959-971, 1992 (a).

Groenevelt, Harry, Liliane Pintelon and Abraham Seidmann. "Production Lot Sizing with Machine Breakdowns". Management Science, 38(1), pp. 104-123, 1992 (b).

Kletter, David B. Determining Production Lot Sizes and Safety Stocks for an Automobile Stamping Plant. S. M. Thesis, MIT, June 1994.

Larson, Robert E. State Increment Dynamic Programming. New York: American Elsevier Pub. Co., 1968.

Lee, Hau L. and Stephen Nahmias. "Single-Product, Single-Location Models", Chapter 1 in Graves, S. C., A. H. G. Rinnooy Kan and P. H. Zipkin, eds., Logistics of Production and Inventory, Amsterdam: North-Holland, 1993.

Matsuo, Hirofumi. "The Weighted Total Tardiness Problem with Fixed Shipping Times and Overtime Utilization". Operations Research, 36(2), pp. 293-307, 1988.

Matsuo, Hirofumi. "A Stochastic Sequencing Problem for Style Goods with Forecast Revisions and Hierarchical Structure". Management Science, 36(3), pp. 332-347, 1990.

Nemhauser, George L. and Laurence A. Wolsey. Integer and Combinatorial Optimization. New York: John Wiley & Sons, Inc., 1988.

Pinedo, Michael. Scheduling: Theory, Algorithms, and Systems. Englewood Cliffs, NJ: Prentice-Hall, 1995.

Pinedo, Michael and Elias Rammouz. "A Note on Stochastic Scheduling on a Single Machine Subject to Breakdown and Repair". Probability in the Engineering and Informational Sciences, 2, pp. 41-49, 1988.

Reiman, Martin I. and Lawrence M. Wein. "Dynamic Scheduling of a Two-Class Queue with Setups". Working Paper No. 3692-94-MSA. Cambridge, MA: Alfred P. Sloan School of Management, MIT, 1994.

Rose, John S. "The Newsboy with Known Demand and Uncertain Replenishment: Applications to Quality Control and Container Fill". Operations Research Letters, 11(2), pp. 111-117, 1992.

Sengupta, Bhaskar. "A Queue with Service Interruptions in an Alternating Random Environment". Operations Research, 38(2), pp. 308-318, 1990.

Sethi, Suresh P. and Qing Zhang. "Hierarchical Production and Setup Scheduling in a Single Machine System", Chapter 8 in Hierarchical Decision Making in Stochastic Manufacturing Systems. Boston, MA: Birkhäuser, 1994.

Smith, Stephen A., John C. Chambers and Eli Shlifer. "Optimal Inventories Based on Job Completion Rate for Repairs Requiring Multiple Items". Management Science, 26(8), pp. 849-852.

# 4. Comparison of operating policies for a single unreliable machine

## Introduction and motivation

In this chapter we will study several different policies that could be used to control a single, unreliable machine producing multiple products to stock (see Figure 4.1). We will see that the selection of an operating policy can have a significant impact on the performance of a production/inventory system. Our goals are to obtain a better understanding of the strengths and weaknesses of different policies, and insight into how the policies that we consider compare against one another in different environments. We hope that our findings will assist decision makers in the selection of an operating policy that is best suited for a particular environment, and stimulate further research in this area of considerable practical importance.

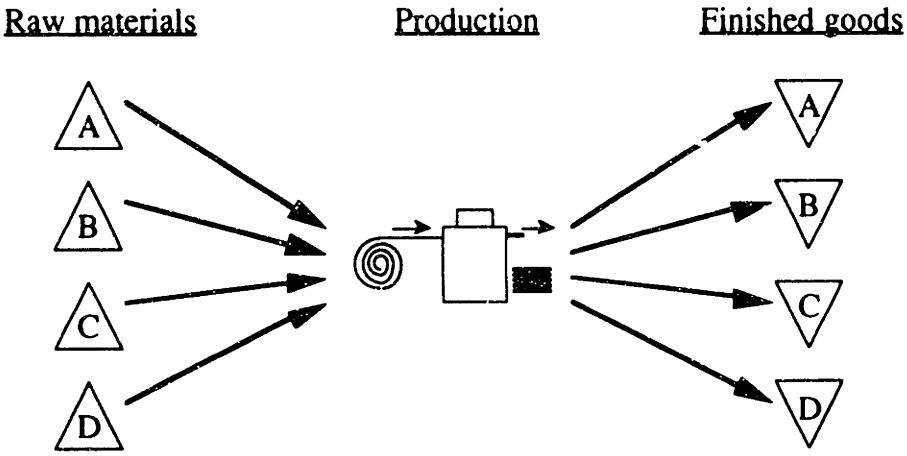


**Figure 4.1** Single machine, multiple product production/inventory system

Before proceeding, we wish to note several important assumptions. First, we will only consider situations in which setup times and/or setup costs are such that batching of production is necessary. Further, we will ignore decisions regarding the

procurement of raw materials, so that the production stage can be considered in isolation. We will only consider replenishment (or *pull*) policies that base production decisions on the quantity of inventory that has been depleted, rather than policies such as MRP (Orlicky, 1975; Baker, 1993) that base production decisions on forecasts of future demand (*push* policies). We will require that all demand must be met, i.e., all stockouts will be backordered.

Our model of the production process will include three types of variability: demand, production, and setup. We will assume that the demand processes are stationary and uncorrelated over time and across parts, and that the demand distribution for each part is known. Our model of the production process will utilize the results from Chapter 2, where we assume that the times between failures are i.i.d. exponential and the times to repair are also i.i.d. exponential, both with known parameters, and where the machine cannot fail while it is being setup, under repair, or idle. Lastly, we will assume that setup times are stochastic. Rather than assume some distribution for setup time, we will explicitly model the setup crew(s) as a shared resource. In this model, two or more machines each periodically require setup from one or more setup crews. In this way, requests for setup may not be served immediately because of queueing effects. We will use this model of the setup process to explore how factors such as setup crew utilization affect the choice of operating policy.

To operate such a production/inventory system, three types of decisions must be made:

- Whether or not to produce anything
- Given that we are going to produce, what part to produce next

- When to stop producing the current part

The policies we consider will differ on these three dimensions. We will see that the manner in which these decisions are made will impact the performance of the system, as well as dictate the requirements for the information systems necessary to support the operation of the system. The policies and their mapping on these three dimensions are discussed in Section 4.1.

**Literature review**

Although many papers that have appeared in the literature compare different operating policies for a single machine, we are unaware of any comparison of different replenishment policies for a machine with setups. We briefly review the comparative analyses that have been performed, even if they do differ fundamentally from the assumptions we make here. We also review some of the literature on determining optimal parameters for a single policy.

Systems without setups

Zipkin (1995) considers a single machine producing multiple items with identical parameters but with no setups. Under the assumption of Poisson demand, the production system is modeled as a M/G/1 queue. The author then contrasts several different base-stock policies by comparing their average inventory, average number of backorders and average total cost.

Under heavy traffic conditions, Wein (1992) solves a Brownian control problem to minimize the sum of holding and backordering costs in a multi-item make-to-stock system. Many other authors have studied the real time control of manufacturing systems when setup costs are setup times are negligible. Much of this work studies

the control of flexible manufacturing systems (FMSs), see for example, Kimemia and Gershwin (1983), Maimon and Gershwin (1988), and Lin and Cochran (1990). These models make the assumption that setup times are negligible and that the objective is to track production as close as possible with demand (that is, to keep the inventory position near zero); see Gershwin (1994). This view is formalized by Bielecki and Kumar (1988) who show that for a single unreliable machine producing a single product with a variable production rate and no setups, there are parameters such that the optimal target inventory level is zero. It should be clear that these models differ considerably from the batch production process that we study in this paper.

Analysis of individual policies

Several authors have studied the determination of optimal or near optimal parameters for a single policy.

A paper by Graves (1980) considers a single machine, multi-product production system with setups. The model is periodic review, deciding at the start of each period whether to continue producing the current part, stop and changeover to the next part, or idle the machine. A major difference from the present study is that production times are assumed to be deterministic. A heuristic is developed and shown to be effective when compared to several others.

Zipkin (1986), Karmarkar (1987, 1993) and Kletter (1994) discuss the determination of optimal parameters for a fixed-lot, reorder point system when the arrival of pull signals to the machine can be approximated as a Poisson process.

Federgruen and Katalan (1994a, 1994b) determine approximately optimal base stock levels for a simple cyclic production sequence with stochastic production times,

setup times and Poisson or Compound Poisson demand. A simple cyclic production sequence is one in which the sequence of parts that will be produced is fixed, each part is produced exactly once in the sequence, and when a part is produced, it is produced until the inventory reaches some predetermined ("base stock") level. Markowitz et al. (1995) also study simple cyclic production sequences, approximating the problem of determining a minimum cost policy by a diffusion control problem. When setup costs (but not setup times) are present, the optimal dynamic lot sizing policy is found for the approximate problem. The authors also present results when setup times (but not setup costs) are present. The policies suggested by this heavy traffic approach differ considerably from a simple base stock strategy.

Chapter 4 of Buzacott and Shanthikumar (1993) discusses a variety of different queueing models of a single machine producing multiple items to stock. The authors also describe and analyze a particular operating policy known as the generalized PA system that they developed; see also Buzacott and Shanthikumar (1992).

Comparison of Push and Pull policies

Spearman et al. (1990), Spearman and Zazanis (1992), and Hopp and Spearman (1996) contrast MRP, CONWIP and Kanban methods of production control. Their broad conclusion is that pull methods are superior to push methods of control. We will not discuss this conclusion, since in the present study we do not consider push systems such as MRP. The authors identify situations in which the CONWIP methodology that they developed appears to be superior to Kanban control. They also acknowledge situations where the reverse is true. However, the authors do not describe a methodology for determining optimal parameters for a CONWIP system.

For another discussion that compares Kanban and MRP systems, see Krajewski et al. (1987).

**Overview of this chapter**

There are six more sections to this chapter. In the next section we describe the policies that we will study, and place these policies within a framework. In Section 4.2 we discuss the metrics that we will use to evaluate and compare these policies, and begin comparing the policies using these metrics. Section 4.3 describes the structure and assumptions of a simulation that we will use to enhance our understanding of the relative performance of the policies. Section 4.4 gives a brief overview of how the simulations of the different policies were validated. In Section 4.5 we describe the simulation experiments performed, and a final section summarizes our observations and conclusions.

## 4.1 Policies for comparison

The purpose of this section is to describe the policies that we will study in this chapter. We first describe a framework that will be used to classify different policies, and then we will place the policies of interest in this framework.

### A framework

As described in the introduction, we will classify our policies according to three dimensions:

A. Whether or not to produce anything
B. Given that we are going to produce, what to produce next
C. When to stop producing the current part

It is important to note that these three dimensions are not independent. We will see below that a choice along one of these dimensions may impose restrictions on what can be chosen for one or both of the other dimensions.

Dimension A specifies what authorizes production. We consider three possible alternatives:

A1. Inventory: At least one part is below its reorder point
A2. Sequence: Based on the last part produced, the sequence tells us whether to produce or idle
A3. Schedule: The schedule tells us at what points in time to start producing

The choices made in dimension A determine the aggregate inventory levels, which impact both holding cost and floor space usage. We have selected the three most common production authorization mechanisms found in the literature and in practice.

Note that these mechanisms differ in terms of how each decides when to produce nothing. A1 produces nothing when the inventory positions of all parts are above their reorder points; A2 will idle the machine based on the last part that was produced; and A3 will produce a part until the inventory position reaches an order-up-to level, and then produce nothing until the next part is scheduled to be produced.

Once production has been authorized, dimension B dictates what will be produced next. We consider three alternatives:

B1. Queue: Parts are served first-in-first-out in a reorder point "pull" system
B2. Sequence: The production sequence is fixed
B3. Inventory: The "most critical" part is chosen

B1 and B2 represent the two most common mechanisms found in the literature. We have not adequately described B3, since "most critical" could be interpreted in a number of different ways. We will more fully describe the variant of this policy that we have chosen to implement below. Although B3 will be difficult to analyze, we include it because it has been used in practice and because it is closely related to the optimal policy in the zero setup time, zero setup cost Brownian control problem described and solved by Wein (1992).

Once production has been authorized and the part to be produced is selected, a changeover begins, followed by production of the part. Dimension C specifies what determines the production quantity, or when production is stopped. We consider three alternatives:

C1. EOQ: Production is stopped after a fixed number of parts have been produced

C2. Inventory level with continuous review: When production begins, the inventory position is observed and the production quantity is chosen to bring the inventory position back up to some fixed level

C3. Schedule: Setups occur at predetermined points in time; the target production quantity is determined by an order-up-to level, but production must stop when next setup is scheduled to occur

C4. Setup crew preemption: Production continues at least until a minimum number of parts have been produced, then stops as soon as a setup crew is available, or when some maximum number of parts have been produced; there are two variants, depending on how min and max are set:

C4a. min and max set to achieve an average lot size
C4b. min and max set based on number of parts to be produced so that the order-up-to level is achieved on average

C1-C3 are frequently encountered in the literature. C4 is motivated by discussions with colleagues at GM, where waiting for setup crews has been observed to significantly impact not only setup times, but the overall variability of the system.

The framework we have proposed highlights that the decision of *when* to produce might be made independently of *what* to produce. B3 provides one example of this. An interesting alternative that we have not included above is authorizing production based in whole or part on an aggregate rather than individual item inventory levels. One can imagine instances in a reorder point system where no item is below its reorder point, yet producing one of the items is advised because the

total system inventory is low. Graves (1980) studies a policy of this type and shows it to be effective when compared to several others.

Lastly, note that since the demand process is assumed to be stationary, we do not consider policies that build anticipatory stocks.

**Policies of interest**

We will consider seven different policies in this chapter. We now describe each in turn.

**P1**. This policy is a typical continuous review lot-size/reorder-point policy that has been extensively studied in the literature (Hadley and Whitin, 1963; Lee and Nahmias, 1993). The *inventory position* (stock on-hand, plus stock on-order, minus backorders) is monitored continuously. When the inventory position reaches the reorder point R, an order for Q units is placed. In terms of the framework above, this policy can be classified as [A1, B1, C1].

**P2**. This policy is similar to P1 in that an order is placed when the inventory reaches a reorder point R. However, unlike P1, once production starts the inventory position is observed, and the production quantity is set to bring the inventory position up to S. Let $\pi$ be the number of parts that must be produced to increase the inventory position up to S. Note that $\pi$ may be different each time the part is about to be set up, depending on how long the pull signal for the part was in queue, and how much demand occurred over that interval. This policy can be classified as [A1, B1, C2].

**P3.** This policy is analogous to P1, except that instead of producing a fixed lot every time, the availability of setup crews are taken into consideration. If a part is waiting in queue at the machine, then once some threshold number of parts $Q^-$ has been produced, the production of the current part will be interrupted as soon as a setup crew arrives. If no setup crew arrives by the time $Q^+$ parts have been produced, production of the current part is stopped and the machine waits until a setup crew is available. $Q^-$ and $Q^+$ should be set so that on average, Q parts are produced. This policy can be classified as [A1, B1, C4a].

**P4.** This policy is analogous to P2, except that instead of exactly ordering-up-to S, the availability of setup crews are taken into consideration, as in P3. Unlike P3, however, the minimum and maximum must vary from one production cycle to the next, since $\pi$, the number of parts that must be produced to increase the inventory position up to S, will typically differ from one cycle to the next. Some rule must therefore be specified that dictates how the minimum and maximum number of parts are set. Given this rule, the policy then observes $\pi$ and determines some $\pi^-$ and $\pi^+$. Once $\pi^-$ parts have been produced, production will be interrupted if a setup crew arrives. If no setup crew arrives by the time $\pi^+$ parts have been produced, production of the current part is stopped and the machine waits until a setup crew is available. Once $\pi^-$ has been chosen, we determine $\pi^+$ so that the expected number of parts produced is $\pi$. This determination can be accomplished by binary search since the expected number of parts produced is non-decreasing in $\pi^+$. We suggest setting $\pi^- = \pi - k$; the single static parameter $k > 0$ then dictates the values of both $\pi^-$ and $\pi^+$ for any value of $\pi$. This policy can be classified as [A1, B1, C4b].

**P5.** This policy is a general cyclic production sequence. The sequence in which the parts will be produced is fixed, each part may be produced more than once in the

sequence. Idle times may be inserted anywhere in the sequence. When the sequence is completed, it is repeated starting at the beginning. When it is time to produce a part, the quantity of parts $\pi$ to be produced is determined by the difference between the current inventory position and an order-up-to point S. This policy can be classified as [A2, B2, C2].

**P6**. This policy is a general cyclic production schedule. Like P5, the sequence in which the parts will be produced is fixed. However, we now also fix the times at which changeovers will occur. As in P5, when it is time to produce a part the inventory position is observed and subtracted from an order-up-to point S to determine a target production quantity $\pi$. However, because the times at which changeovers occur has been pre-determined and the machine is unreliable, the number of parts actually produced may be less than $\pi$. On the other hand, if $\pi$ parts are produced before the next changeover is scheduled to occur, production stops and the machine remains idle until the changeover. This policy can be classified as [A3, B2, C3].

**P7**. This policy has machine changeovers at regular intervals, but the part to be produced at each interval is not pre-determined. Rather, the part that is determined to be "most critical" is produced over that interval. Otherwise, this policy is identical to P6. It can be classified as [A3, B3, C3].

Grouping C4a and C4b together, a full enumeration of all of the possible combinations of the different dimensions gives 36 different policies, of which we have considered only six (seven considering C4a and C4b separately). Twenty six others make no physical sense; for example, A1 and B2 are incompatible, as are A2

and B1, A2 and B3 and A3 and B1. Some of the 26 excluded combinations do not form a replenishment policy, such as [A2, B2, C1].

This leaves only six policies that we do not consider (seven if we consider C4a and C4b separately). Although we will not consider them in this chapter, we discuss them briefly. They are P8 [A1, B3, C1], P9 [A1, B3, C2], P10 [A1, B3, C4a], P11 [A1, B3, C4b], P12 [A2, B2, C4b], P13 [A1, B1, C3] and P14 [A1, B3, C3]. P8-P11 are an interesting variant on P1-P4, in which a reorder point authorizes production, but instead of serving parts in a first-come-first-served fashion, the "most critical" part is selected from those parts for which production has been authorized. P12 is a variant of the general cyclic schedule P5, in which the availability of setup crews are taken into consideration. P13 and P14 describe a rather unusual pull system. Production is authorized by a reorder point, the parts are served first-in-first-out (or the "most critical" part is selected in the case of P14), but production continues until either an order-up-to level is reached (like P2), or until it is time for the next setup.

Although we have only chosen seven of the fourteen possible policies that are suggested by our framework, we feel that this is a reasonable set to consider, since it is representative of the policies that we have observed in practice, as well as those that have appeared in the literature, and is fairly representative subset of the fourteen possible policies.

## 4.2 Performance metrics

We now discuss the metrics that we will use to evaluate our different policies, and begin to explore how the different policies behave according to these metrics.

There are seven metrics that we believe will highlight the differences between the policies to be considered. In no particular order, they are:

1. Inventory costs: Average cycle stock
2. Inventory costs: Safety stock requirements
3. Setup costs
4. Variability of the time between production starts
5. Variability of raw material requirements
6. Idleness of machine due to waiting for setup
7. Floor space requirements, or maximum inventory level

Metrics 1-3 are analogous to the costs minimized in traditional inventory models. If the policies are parameterized such that each has the same average time between setups (equalizing metric 3), then the average cycle stock will also be the same for all policies. In a reorder point system, the required safety stock is determined by the distribution of demand over the leadtime. For a cyclic sequence or cyclic schedule, the required safety stock is determined by the distribution of demand over the interval between production starts. In this way, Metrics 2 and 4 are related for such systems.

In a multi-stage production system, the requirements for an upstream stage are generated by the production that occurs at the downstream stage(s). Although we

consider only a single production stage, metrics 4 and 5 measure the variability of the requirements placed on the upstream stage(s), in terms of the variability in the time between orders, and the variability in the size of the orders. The more variability is present, the more difficult it is to coordinate or synchronize the stages. More variability may imply that higher safety stock levels must be held upstream.

Metric 6 is not terribly important in itself, but does indicate the impact of the setup crew(s) on the utilization of the machine. Since we have chosen to explore policies that are designed to specifically reduce the waiting time for crews, this metric will be an important indication of the degree to which such reduction can be (or has been) successful.

To some readers, metric 7 may at first seem unimportant, or might be dismissed as a factor that is typically included within inventory holding cost. We include it here as an important metric because floor space usage can be a factor of substantial importance in industry, and can drive decisions such as optimal lot sizes; see Kletter (1994) for further discussion of this point in the context of an automobile stamping plant. It is important to note that the amount of floor space that must be allocated to storing a part can be related to the maximum inventory and not the average inventory on-hand. For example, this is true in some areas of a metal stamping plant where finished metal parts (such as hoods or doors) are stored in containers that consume as much floor space when they are empty as when they are full.

Note that there are no production costs in the model, with the exception of setup costs.

**Discussion of the metrics**

Using principles from inventory theory and queueing theory, we now discuss how the different policies will impact the different metrics. In some cases it will not be possible to say very much, if anything, about certain metrics for some of the policies. Some policies are more difficult to analyze than others, and different metrics will be harder to analyze depending on which policy we are considering. For this reason, we will use simulation later in this chapter to enhance our understanding and facilitate richer comparisons.

First, we will assume as above that the policies are parameterized such that for a given part, each policy has (approximately) the same average time between setups, equalizing metrics 1 and 3. This normalization frees us from the complex trade-offs between lot size, lead time and inventory holding cost (Karmarkar, 1987; Karmarkar, 1993).

Metric 2, the safety stock required to achieve a target service level, has been given much attention in the classic literature on inventory theory. For P1-P4, the required safety stock is determined by the distribution of demand over the leadtime. For these policies, the leadtime is a random variable, and the moments of the leadtime distribution are non-trivial to estimate. The arrival of pull signals in queue for production forms a type of finite source queue. Since neither the interarrival times nor the service times in this queueing system are of a simple form (e.g., Poisson), this system is difficult to analyze. We defer direct comparisons between P1-P4 until later. The safety stock requirements for P5-P7 are determined by the distribution of demand between production starts. Suppose for simplicity that all parts have the same demand distribution. If the production sequence and setup frequencies are the

same for all three of the policies, then we expect the safety stock requirements to be higher for P5 than P6, because the time between production starts is variable for P5 but not for P6, whereas the mean time between production starts is the same. It is difficult to say how P7 will perform, although one would expect that it would require less safety stock than P6.

Metric 4 may be difficult to determine analytically, but we can make several important observations. First, there will be no variability in the time between production starts for P6. Second, if P1-P4 are parameterized to achieve the same average lot size, then the variability in the time between production starts for P3 will be greater than P1, and the variability for P4 will be greater than P2, because P3 and P4 induce additional variability in the production quantity, and therefore have greater variability in the time until the reorder point is reached. Lastly, we note that there will be variability in the time between production starts for P5 and P7. The extent of this variability is a function of several factors. The variability in both demand and production time will have great impact on both policies.

Metric 5 is the variability in the quantity of raw materials consumed during a production run. We begin with the obvious observations that the variability will be zero for P1, and the variability of P4 will be greater than P2 and the variability of P4 will be greater than P3. Since all the policies we consider are replenishment policies, the variability in the quantity of raw materials consumed is determined by the variability in the time between production starts and the variability in demand. Since the variability in the time between production starts is zero for P6, the variability in the quantity of raw materials consumed will be greater for P5 than P6, and greater for P7 than P6, assuming the same variability in demand across policies.

Since setups occur at known, fixed points in time for P6 and P7, we will assume that metric 6, the waiting time for setups, is zero for these policies. This assumption is essentially equivalent to assuming that the setup crew(s) can be scheduled so that the machine never has to wait for the crew, and that the setup times are deterministic so that there is no deviation from the schedule. For P1, P2 and P5, metric 6 will be the same. P3 and P4 will of course incur less waiting time for setup by their design.

Lastly, we discuss metric 7, the required floor space. Since we permit more parts to be built under policies P3 and P4 than P1 and P2, the required floor space will be greater for P3 and P4. It is difficult to say whether P1 or P2 requires more floor space without knowing which requires more safety stock. Similarly, P5, P6 and P7 will each have maximum inventory equal to their order-up-to level. If the setup frequencies are equalized across the policies, the required safety stock dictates any differences in the order-up-to level, which in turn determines how the maximum inventory levels of the three policies compare.

**Measuring the policies**

At this point we can begin to draw some inferences about the different policies that we consider. We now describe the extent of our understanding of the various policies and highlight areas that need further exploration.

P1 induces equal production quantities but variable inter-order times for raw materials. P3 induces some variability in the production quantities, however, the extent of this variability is an endogenous parameter that, at least in principle, can be chosen to optimize the tradeoff between this variability and the benefits

associated with less waiting for setup. Indeed, P3 is a superset of P1 since one can set the minimum and maximum lot sizes to be equal.

P2 induces both variable production quantities and variable times between demand. P4 is similar, except it further increases the variability in production quantity. We note that P2 and P4 have more complex information systems requirements than either P1 or P3, since the inventory position needs to be measured periodically. In contrast, reorder point triggering can often be accomplished with simple, non-technical methods such as a line of paint on the shop floor, or by placing a card representing a pull signal on a container of parts. It is difficult to say under what circumstances P2 will enjoy lower safety stock requirements and floor space requirements than P1. If this does not occur, then there is no advantage to P2 over P1. We hope that our simulations will provide an answer to this question. We will also rely on simulation to help us understand the extent to which P3 and P4 improve and degrade when compared to P1 and P2 in a variety of different types of production environments.

P5 does not appear to be competitive with P6 based on the metrics we have chosen. When the sequence chosen is the same and setup frequencies are equalized, we expect both the safety stock and floor space required to be greater than P6; P5 induces variability in the time between production starts, whereas for P6 there is none; the variability in the quantity of raw materials consumed will be greater for P5 than P6; and P5 will incur waiting for crews, while P6 fixes setup times, which allows the setups to be scheduled, which should have the effect of reducing or eliminating delay.

P6 induces equal inter-order times for variable quantities of raw materials. We contrast this to P1, which does the opposite. Since demand is variable, there must be variability in either the quantity or the timing. Which is preferable is difficult to say in general. Variability in timing but not quantity makes the sizing of decoupling inventories much easier. Variability in quantity but not timing might allow a complex multi-stage production process to operate according to a rather simple schedule, like those advocated by Muckstadt and Roundy (1993).

Federgruen and Katalan (1994a), who describe an approximate method to optimize a special case of P5, note that "no acceptable analytical method appears to prevail" to evaluate a given parameterization of P6, let alone determine an optimal parameterization in general. However, based on the above observations, we should not be surprised if it is fairly easy to determine parameters for P6 that will outperform P5 according to our metrics.

P7 is similar to P6, except the inter-order times for raw materials are now variable. It remains to be seen to what extent, if any, this extra variability permits required safety stocks (and thus required floor space) to be reduced. From a system-wide perspective, P7 will be preferred to P6 when the reduction in finished goods inventory holding cost that results is greater than the increase in holding cost for raw materials that must be held upstream to account for the variable inter-order times that are induced by P7.

Lastly, we note that a question of particular interest to us is understanding under what circumstances P6 or P7 will outperform P1-P4, and vice versa. We will attempt to answer this important question with a variety of simulation experiments later in this chapter.

## 4.3 Simulation of operating policies

In this section we describe the three critical submodels that we have implemented for use in our simulations. These are models of the demand process, the production process, and the setup crews. We begin this section with a description of how the random processes of interest are simulated. We then proceed to describe each of the three submodels. At the conclusion of this section we describe a few small details regarding how the policies described earlier in the chapter have been implemented.

### Generation of random variables

Central to the simulation of random phenomena is the ability to sample from one or more random variables that describe the randomness inherent in the system of interest. There are a variety of sophisticated and specialized techniques that have been developed to accomplish this, and are now integrated into simulation languages and described in introductory simulation textbooks (Law and Kelton, 1991; Bratley et al. 1987; Pritsker, 1995).

For our purposes, we will use the most basic and general of all techniques: the inverse-transform method. To sample from a known distribution function F, the method first draws a random number that is uniformly distributed on [0, 1]. Denoting this random number by x, the sample from the distribution F is $F^{-1}(x)$. If we cannot invert the distribution function F, we can perform a binary search to find t such that $F(t) = x$. Therefore, in the discussions below we will be satisfied when we have identified a distribution function that characterizes the stochastic behavior of interest.

Sampling from the uniform [0, 1] distribution has been exhaustively studied. See Bratley et al. (1987) or Knuth (1981) for enlightening discussions. In our simulations we have used a linear congruential generator built into the Standard Apple Numerical Environment (SANE) on the Macintosh, as described in Apple Numerics Manual (1988).

**Demand submodel**

To model the demand process in our simulations, we need to accomplish two particular tasks. First, we need to be able to randomize the demand that occurs over an interval of any length. Second, when simulating the reorder point policies, we will need to randomize the time until the reorder point is reached. This is equivalent to saying that we need to determine the random time until some number of parts has been demanded. The difficulty of this second task will depend on the assumptions about the demand distribution.

In all of our simulations we will assume that demand over an interval is normally distributed with mean and standard deviation proportional to the length of the interval. As a result, the demand process can be modeled as a Brownian motion. This observation is helpful, since we will rely on the following key result from the theory of Brownian motion. If demand over an interval of length t is distributed normally with mean $\mu t$ and standard deviation $\sigma\sqrt{t}$, then the distribution function $G_x(t)$ of the time until x parts are demanded is

$$G_x(t) = 1 - \Phi\left(\frac{x-\mu t}{\sigma\sqrt{t}}\right) + e^{2x\mu/\sigma^2}\Phi\left(\frac{-x-\mu t}{\sigma\sqrt{t}}\right) \qquad t \geq 0,$$

where $\Phi(t)$ is the standard Normal distribution function; see Heyman and Sobel (1982). $G_x(t)$ is sometimes called the distribution function of the *first passage time* to x. Simple numerical approximations for the standard Normal distribution function exist; for our simulations, we have implemented the one described in Abramowitz and Stegun (1964) and Press et al. (1989).

It can also be shown (Cox and Miller, 1965) that*

$$E[G_x] = \frac{x}{\mu}, \quad Var[G_x] = \frac{x\,\sigma^2}{\mu^3}.$$

Note that these expressions are asymptotically true for any demand process that can be modeled as a renewal process; this follows from the central limit theorem for renewal processes (Ross, 1983).

In our simulations of reorder point systems we will require the distribution of demand over an interval of length t, conditioned on the event that demand is equal to B over an interval of length s, $s > t$. We can show that this distribution is Normal with mean $Bt/s$ and variance $t(s-t)/s$. See Ross (1983) for a proof using Bayes' rule in the case of a Brownian motion without drift. The proof for the case with drift follows by a similar argument.

Although these simple results are encouraging and easy to implement, caution should be taken to ensure that this is a realistic model of the demand process. Although this model is not likely to accurately reflect the demand process over very

---

* Cox and Miller obtain the first two moments by differentiation of the moment generating transform. However, their expression on p. 222 for the variance is incorrect.

short intervals, our simulations will not be sensitive to this shortcoming. For example, in the reorder point models we are concerned with the demand that occurs between reorders, and over no smaller time interval. The primary concern is that the coefficient of variation not be much larger than 0.4. For example, if the coefficient of variation were 0.5, then the left tail of the demand distribution below $2\sigma$ represents negative demand, which has no physical interpretation. If the coefficient of variation remains small, then there will be a negligible chance that demand over the interval is negative.

**Production submodel**

To model the production process in our simulations, we need to accomplish two tasks. First, we need to determine the (random) number of parts produced over an interval of a given length. This is essential for simulating P6 and P7, where the time available for production is fixed. Second, we need to determine the (random) time required to produce a given number of parts. This is important for P1-P5, where the number of parts to be produced is fixed*.

In our simulations we will not explicitly model the failure and repair of the machine. Rather, we have implemented the results of Chapter 2, which provide us with the distributions we need, so that we do not need to simulate each failure and each repair. By using the results from Chapter 2, we are assuming that the times between failures are i.i.d. exponential, and the times to repair are also i.i.d. exponential, both with known parameters, where the machine cannot fail while it is being setup, under repair, or idle.

---

* Of course, P3 and P4 will be somewhat more complicated. We will describe how these policies are simulated at the end of this section.

Equation (3) of Chapter 2 gives us the density of uptime over an interval of given length. Scaling this by the production rate gives us the number of parts produced over a given interval. Equations (18) and (25) of Chapter 2 give the mean and variance of this distribution. Similarly, equation (33) of Chapter 2, when scaled by the production rate, gives the density of time to produce a given number of parts. It was also shown that if we denote the failure rate by $\lambda$, the repair rate by $\mu$, the production rate by p, and the number of parts to be produced by q, then the mean and variance of **R**, the time to produce a given number of parts, are

$$E[\mathbf{R}] = \frac{q\ \lambda}{p\ \mu}, \quad Var[\mathbf{R}] = \frac{2\ q\ \lambda}{p\ \mu^2}.$$

The expressions for the distributions involve modified Bessel functions of orders zero and one. Codes for evaluation of these functions are provided in most commercial numerical libraries, although many excellent codes are in the public domain and are available via *netlib* (Dongarra and Grosse, 1987). See Chapter 2 for further citations. For our simulations, we used the algorithms in Press et al. (1989). To numerically integrate the densities to obtain cumulative distribution functions, we used the `qromb` routine that implements Romberg's method from Press et al. (1989).

**Setup crew submodel**

For our simulations that incur random waiting time for a setup crew to arrive, we require the distribution of waiting time for setup. Although we could incorporate any given distribution into a simulation, we would like to explicitly model the queueing effects that arise when one or more setup crews are shared among several

machines. In a stamping plant, for example, setups involve a series of tasks that are performed by a number of different workers with a variety of different skills. In this case, a setup crew is a specialized team that travels from machine to machine performing setups. A model of this system should therefore include not one machine but several machines that compete for use of the setup crew(s). Rather than simulate several parallel machines, we have chosen to adopt a simple model of a multi-machine system from which a waiting time distribution is easily obtained.

Our model of the setup process is a closed queueing network, in which the customers circulating through the network are the machines requiring setup. In this network there are two stages. The server at the first stage represents the setup crew. Machines that arrive at this stage queue for service. The machines are served from this queue first-in-first-out, and service times at this stage are the setup times, i.e., the time required to changeover the machine once the crew arrives. After service, the customer (machine) enters the second stage. Here there are as many servers as machines. The interservice times at this stage represent the times between completion of a setup and the request for the next setup.

This queueing model is sometimes called the "machine interference" or "machine repairman" model. It was first studied by Benson and Cox (1951), who studied this system as a M/M/c finite source queue and found the steady-state distribution of queue length. See Gross and Harris (1983) for a complete discussion. Bunday and Scraton (1980) have proven that the steady-state distribution for number in queue for the G/M/c finite source queue is the same as that of the M/M/c finite source queue. This exciting result means that the steady-state distribution of queue length

depends only on the first moment of the distribution of time between setups. We must, however, assume that setup times are exponentially distributed.

We comment briefly on the known results for the case when the service times are not exponentially distributed; not surprisingly, the problem is more difficult. Two classic papers on the M/D/1 finite source queue are Ashcroft (1950) and Benson and Cox (1951). Benson and Cox discuss a variety of other types of systems as well. Saaty (1961) provides an excellent summary of the results known at the time; see also Cox and Smith (1961). These results, however, are largely subsumed by Takács' (1962) study of the M/G/1 finite source queue. He presents closed-form expressions for the Laplace transform of both the transient and steady-state distribution of queue length. He provides many other results as well, including an expression for the steady-state distribution of waiting time as a sum of convolutions of the distribution of service time. See Jaiswal (1968), Stecke and Aronson (1985), and Suri et al. (1993) for additional discussion, results and references.

We have chosen to accept the assumption of exponentially distributed setup times and use this model to generate our distribution of waiting time for setup. We now show how to use the steady-state distribution of queue length to obtain the steady state distribution of waiting time for setup. Suppose the arrival rate of machines for setup is $\lambda$ and the service (setup) rate is $\mu$. From the references cited above, the steady-state queue length distribution $p_i$, the probability that there are i machines in queue at the first stage, or being serviced at the first stage, is

$$p_i = \begin{cases} \binom{M}{i} \left(\frac{\lambda}{\mu}\right)^i p_0 & 0 \le i < c \\ \binom{M}{i} \frac{i!}{c^{i-c}c!} \left(\frac{\lambda}{\mu}\right)^i p_0 & c \le i \le M \end{cases}$$

where there are a total of M machines served by c crews. $p_0$ is found from the fact that the sum of the $p_i$'s must equal one. Of course, the waiting time is zero with probability $p_0 + p_1 + \ldots + p_{c-1}$. If we condition on the queue length being equal to i ($i > c$), then the time until an arriving customer can be serviced is an $(i-c+1)^{th}$-order Erlang distribution with rate $c\mu$, since service times are exponentially distributed, the servers work in parallel, and service can begin after i-c+1 customers have been served. Denoting the density of an $n^{th}$-order Erlang by $E^n(t)$, the density of waiting time for setup is

$$w(t) = \sum_{i=c}^{M} p_i \; E^{i-c+1}(t), \qquad t > 0.$$

It follows that the first two moments are

$$E[w] = \sum_{i=c}^{M} p_i \; \frac{i-c+1}{c\mu},$$

$$Var[w] = \sum_{i=c}^{M} p_i \; \frac{(i-c+1)+(i-c+1)^2}{(c\mu)^2} - E[w]^2.$$

Although these expressions are exact, they are not exactly what we want. This result gives us the density of waiting time at a random instant, or the so-called *virtual waiting time*. We are actually interested in the waiting time experienced by an arriving customer. These two quantities will be equal if and only if the arrival process is Poisson (Wolff, 1982). If sufficiently motivated, we could find the waiting time experienced by an arriving customer by replacing the $p_i$ in all of the above expressions with $r_i$, the probability that an arriving customer finds i customers in

the system. Note that the arrival instants are regeneration points for the G/M/c finite source queue. As a result, one could analyze the embedded Markov chain that represents the number of customers in the system at the arrival instants. Finding the steady state probabilities $r_i$ of this Markov chain involves determining the PMF for the number of customers served between arrivals, conditioned on the number of customers present in the system, and then solving a linear system of equations. See Kleinrock (1975) for an example of this technique applied to the G/M/c infinite source queue. We elect to use the virtual waiting time distribution and not undertake this effort, for reasons described below.

Before proceeding further, a number of assumptions made by this queueing model should be made explicit. First, we have already mentioned the fact that this model assumes that setup times are i.i.d. exponential, which might be quite different from the reality of a particular situation. Second, the elegant closed form results that we obtain are for the virtual waiting time, not the waiting time experienced by an arriving customer. Third, the model assumes that the population of arriving machines are homogeneous, sharing the same setup time distribution and interarrival distribution. Fourth, the model assumes that any machine can be serviced by any setup crew, and further, that the setup crews all work at the same rate. Fifth, we assume that the system is back in steady-state each time we observe it.

Some of these assumptions will be more realistic than others, depending on the environment in which they are applied. For example, the assumption of exponentially distributed setup times might be reasonable for certain types of equipment in a stamping plant, where a setup is a complex multi-stage task involving a series of teams that each must perform tasks in a certain order. In this case, there is high variability in the total completion time for all tasks, so that a

coefficient of variation of one might be quite reasonable. In other environments, setup times will be much closer to deterministic than exponentially distributed.

Indeed, it seems highly unlikely that all the assumptions of this model will hold for any realistic production system. We believe that this is not a cause for great concern. The intent of the setup crew submodel is only to produce a distribution of waiting times that is reflective of something that might be encountered in a real manufacturing environment. Further, we would like this distribution to behave in a reasonable way as the utilization of the setup crew(s) is varied, as the setup frequencies are varied, and as the setup times are varied. Since our model captures the finite source nature of the queueing system and is independent of all but the first moment of the distribution of time between setups, we feel this model will serve our purposes well.

**Implementation of policies**

We now describe a few important details about our implementation of some of the policies for the purpose of simulation.

As mentioned in the introduction, for all of the simulations we assume that stockouts are backordered. In our implementation we make the further simplifying assumption that once production starts, the production process can outpace demand so that further stockouts do not occur.

Recall that for P1 and P2, the production quantity is determined in advance of the start of production, so we need only to know the length of time required to build this quantity of parts. Policies P3 and P4 are more complex, since once a minimum number of parts have been built, the production process will be interrupted when a

setup crew arrives, unless the setup crew does not arrive by the time some maximum number of parts have been produced. To implement this, we do the following for each production run. First the amount of time to produce the minimum number of parts is determined. Next, we envision the machine "entering the queue" for setup at this point in time, by sampling from the setup crew wait distribution. If the waiting time for the setup crew is zero, then production stops immediately. If the wait is non-zero, we determine the amount of time it would take to produce-up-to the maximum production level. If this time is less than the waiting time for the setup crew, then we achieve the maximum production level and wait the *remainder* of the time for the setup crew to arrive. If the setup crew arrives before the maximum production level is achieved, we must estimate the amount of output achieved at the moment the setup crew arrives. That is, we wish to determine the production over an interval of length $t_1$ given a known quantity of production over an interval of length $t_2 > t_1$. Using the results of Chapter 2, we could work this out exactly by applying Bayes' rule; the resulting density is a ratio of three modified Bessel functions. Rather than incur this extra complication, we make the simple approximation that the output over the shorter interval is proportional to the output of the longer interval, with ratio equal to the lengths of the two intervals. That is, if it takes one hour to produce 100 units, then in half an hour we approximate the output as 50 units. This results in the same average output.

Next, we wish to describe how we have modeled idle times for P5. Idle times allow us to control the setup frequency of the machine. Let T be the desired cycle length, i.e., the desired time between starts of the production sequence. Of course, we require that the expected production time and expected setup time for the sequence be less than T. We could insert idle time for each cycle equal to the difference

between T and the expected production and setup time. This would achieve an average cycle length of T. Instead, we will dynamically determine the amount of idle time to insert as follows. If at the end of the $n^{th}$ cycle, the amount of elapsed time is less than nT, we insert the amount of idle time so that the next cycle begins at nT. If the elapsed time is greater than nT, we do nothing. The amount of idle time will therefore vary from one cycle to the next, but provided that the utilization of the machine is less than 1, the average cycle length will be T. We feel this is a more sensible way to insert idle time because we do not idle when we are behind, instead idling when we are ahead. We hope this will make P5 more competitive in our simulations. It is important to note, however, that while a static strategy has been successfully analyzed by Federgruen and Katalan (1994a, 1994b), the idling strategy we have suggested will be more difficult to analyze.

Lastly, we comment on our implementation of P7. Among the variety of ways to interpret "most critical", we have chosen the following. Suppose the machine is setup for production every T time units. We define the most critical part as the one that has the highest probability of stockout over the next T time units if that part is not produced in the upcoming production interval. This definition attempts to take into account not only the stock on-hand and the average demand rate (i.e., the "time supply"), but also the variability of the demand process. If the inventory position for a part is negative, then the probability of stockout will be 1. If more than one part has a negative inventory position, then we need a way to break the tie. We (somewhat arbitrarily) select the part with the most backorders.

## 4.4 Validation of simulations

The purpose of this section is to describe some of the validations that can be (and have been) performed in an attempt to confirm that our simulations are behaving correctly. The validations that we will describe consist of collecting statistics on inventory levels, production times, waiting times, and so forth. We will perform two types of validations. The first is to compare certain statistics against known theoretical results. As described above, the simulation of random phenomena is often accomplished by generating pseudorandom values from a known distribution. When this is the case, we will validate the behavior of the simulation by collecting the sample mean and sample variance of the pseudorandom sequence and comparing these to known exact expressions. A second technique, which we describe at the end of this section, is checking certain statistics against one another for internal consistency. We will not describe every validation that we have performed, but instead highlight some of the more important tests.

### Time between reorders

When simulating reorder point-based policies, one concern is that the reorder point mechanism is functioning properly. To test this, we measure the time between reorders for each part while the simulation is running. As described in Section 4.3, we model demand as a Brownian motion, so the distribution of time between orders is a first passage time distribution for a Brownian process. In Section 4.3 we gave simple expressions for the mean and variance of this distribution.

### Demand process

For each policy we track how much demand occurs over the course of the simulation, and divide by the length of the simulation to obtain the sample mean

demand rate. For some of the policies where we observe demand at equally spaced intervals, we can also compute the sample variance.

**Wait for setup**

In Section 4.3 we described the G/M/c finite source queueing model that we use to model the competition among machines for use of the setup crew(s). We showed how to compute the density of waiting time for setup, and gave expressions for the mean and variance of this distribution.

**Production process**

In Section 4.3 we described how we utilize the model of Chapter 2 to simulate the output of an unreliable machine. Equations (18) and (25) of Chapter 2 give the mean and variance of machine uptime over an interval of given length. Exact expressions are also given in Chapter 2 and above in Section 4.3 for the mean and variance of the time to produce a given number of parts.

**Cross-checking**

Suppose we collect statistics on the average production time, the average minimum inventory level (the inventory level just before production starts), and the average maximum inventory level (the inventory level just after production ends). We expect these to be related in the following way. The difference between the maximum and minimum inventory level, subtracted from the average number of parts produced, is the average demand during a production run. This should be approximately equal to the average demand rate multiplied by the average production time.

For reorder point-based policies, consider the difference between the reorder point and the average minimum inventory level (the inventory level just before production starts). This quantity is the average demand over the leadtime. Clearly, this quantity should be approximately equal to the average demand rate multiplied by the average leadtime, where the average leadtime is the sum of the average waiting time in queue at the machine, the average waiting time for a setup crew, and the setup time. A similar verification of the minimum inventory level can be performed for non-reorder point based policies as well.

**Summary**

We have described a variety of different validations that we performed. Through these tests, we have been able to validate almost every performance metric of interest. One metric that we were not able to independently validate with the methods described above is the average time spent in queue at the machine. This is because the underlying mathematical model is a finite source G/G/1 queue with a non-homogeneous customer population. However, we have carefully observed the behavior of the queue and validated the other aspects of the queueing system such as the production times and setup times. The other metric that is difficult to independently validate is the service level. This is difficult to validate for the reorder point-based systems because the distribution of leadtime demand is not known. The remaining policies are difficult to analyze. We have therefore relied on validations of the inventory level and demand process to convince ourselves that the service level statistics are accurate.

## 4.5 Simulation experiments

We now describe the simulation experiments that we have conducted in an effort to enhance our understanding and allow richer comparisons of the policies that we have chosen to study. We will be using real (although disguised) data from a GM metal stamping plant. In this section we will study two stamping lines from this plant, one that produces three different parts and another that produces nine parts. We have chosen to study these two production lines because they are reasonably representative of the different stamping lines in this particular plant, yet quite different from one another.

We begin by describing a single simulation of one of the policies for the three part line. This simulation will serve as a base case, and will be compared to a variety of simulations of the other policies. The nine part line will be studied in the same fashion. To put our observations on more firm ground, we will also explore a variety of parameter changes to our nine part base case.

**Base Case I: Inputs**

The first case we study has three parts that each have the same parameters. These parts are high volume for this stamping plant, each with demands of 12,400 per week. The plant operates 120 hours per week, so this equates to 103.33 per hour. Over the course of a week, the standard deviation in demand is approximately 2,480, so the coefficient of variation in demand over a week is 0.2. When the stamping line is operating, it produces parts at 495 per hour. The machine fails on average once per hour and requires 15 minutes on average to repair. These parts require one hour to be setup once a setup crew arrives.

In this base case we will simulate P1, so we must choose a lot size and a reorder point. When making lot sizing decisions, this plant considers the traditional EOQ costs (setup and inventory holding cost) as well as floor space constraints (Kletter, 1994). To avoid disclosure of proprietary information, we will only state that these considerations are minimized for these parts when the setup frequency is approximately twice per week, for a lot size of 6,200. Our study will not directly use cost data of any kind. Excluding waiting times for setup crews, the above parameters result in a machine utilization of 83%*.

This plant targets high service levels (e.g., 98%). An iterative process was used to set the reorder point. The process begins by running a simulation with a guess at the reorder point and observing the mean and standard deviation of leadtime. This is in turn used to make a more accurate guess of the reorder point. We do not consider it critical that a target service level is achieved exactly. Rather, it will be important that the level of service achieved is reasonably high, and that the other policies are parameterized to facilitate comparison. In the end, a reorder point of 3,100 was chosen.

Lastly, we describe the parameterization of the setup crews. In this particular plant, a single machine is typically serviced by only one crew, but a crew is shared among several machines. For our simulation, we therefore have a single setup crew, and set the average service time for the crew to be one hour (which corresponds to the setup times on the machine that we are studying). In this plant, setup crews are typically utilized between 70 and 80%. In our simulations, the crew serves 5

---

* We define utilization as the fraction of non-idle time, including time for production, setup and repair. Thus, a set of parameters is a feasible machine load if and only if it results in a utilization of less than 1.

machines with requests for setup arriving about once every six hours. This results in a crew utilization of 83%. The input data for the simulation are summarized in Table 1.

| | |
|---|---|
| Demand rate = 12,400/week | MTBF = 1 hour |
| Demand std. dev. = 2,480/week | MTTR = 0.25 hour |
| Production rate = 495/hour | SAA: 80% |
| Setup time = 1 hour | |
| Lot size = 6,200 | # of setup crews = 1 |
| Reorder point = 3,100 | # of machines/crew = 5 |
| Machine utilization: 83.28% | Arrival rate of requests for setup = 1/6 hours |
| | Mean service rate = 1 hour |
| | Setup crew utilization: 83.33% |
| Hours per week = 120 | |

**Table 4.1** Data for Base Case I

**Base Case I: Results**

The base case is a simulation of P1 as described above. The production system was run for 20 simulated weeks in order for a "steady state" to be reached, and then statistics were collected for 100 simulated years (5,200 weeks). This required 5 minutes and 21 seconds on a Power Macintosh 7100/80 running in native mode.

The resulting statistics are reported in Table 4.2. We now discuss each of the rows of this table, introducing the statistics that are generated by the simulation and relating these back to the metrics described in Section 4.2. These metrics will be used to evaluate and compare the different policies.

The first two rows give the mean and standard deviation of the number of parts produced. Recall that the variation in production quantity is Metric 5. For policy P1, we observe no variation, as expected.

The next two rows are the mean and standard deviation of the waiting time for a setup crew; this is Metric 6. The theoretical mean and standard deviation are 1.16 and 1.58, respectively. Although the waiting time distribution is the same for all three parts, we observe some minor variation across parts. Note that by simulating three parts with identical parameters, the variation in statistics across parts provides an indication of the error in our estimates of the metrics.

The mean and standard deviation of the leadtime are reported next. The leadtime is defined as the time spent in queue at the machine, plus the waiting time for a setup crew, plus the setup time. This definition of leadtime is chosen to correspond to the interval over which safety stock provides protection. The mean and standard deviation of leadtime describe the congestion at the machine.

The following two rows are the mean and standard deviation of the time to produce one lot. As described in the previous section, we know that the theoretical mean and standard deviation are 15.66 and 1.25, respectively.

Next, two standard measures of service are reported: Type 1 and Type 2 (Nahmias, 1989). Type 1 service refers to the fraction of reorder cycles in which no stockout occurs. In contrast, Type 2 service is the fraction of demands that are filled from stock. For various reasons, both measures of service are prevalent in the literature and in practice. We note that our choice of lot size and reorder point results in a

Type 2 service level of approximately 98%, which is a "high level of service", as desired. Recall that Metric 2 is the required safety stock level.

The next three rows report on average inventory levels. The first row gives the average inventory level just after production is completed. This is Metric 7, the maximum inventory level achieved over the reorder cycle, which impacts floor space requirements. The next row is the average inventory level just before production begins. This quantity is the safety stock level. The last of the three rows gives the time average inventory level. Since inventory holding costs are proportional to the average inventory level, this is also Metric 1.

| | Part 1 | Part 2 | Part 3 |
|---|---|---|---|
| Average lotsize | 6200 | 6200 | 6200 |
| Standard deviation | 0 | 0 | 0 |
| Avg wait for setup | 1.15 | 1.18 | 1.17 |
| Standard deviation | 1.75 | 1.58 | 1.60 |
| Avg leadtime | 12.63 | 12.76 | 12.63 |
| Standard deviation | 9.89 | 9.99 | 9.90 |
| Avg production time | 15.65 | 15.67 | 15.63 |
| Standard deviation | 1.27 | 1.27 | 1.25 |
| Type 1 Service level | 89.11% | 88.40% | 88.93% |
| Type 2 Service level | 98.16% | 98.01% | 97.89% |
| Avg max inventory | 6334.06 | 6313.81 | 6334.64 |
| Avg min inventory | 1753.89 | 1740.61 | 1747.50 |
| Avg inventory | 4043.97 | 4027.21 | 4041.07 |
| Avg time betwn starts | 60.14 | 59.71 | 60.19 |
| Standard deviation | 18.64 | 18.34 | 18.83 |

**Table 4.2** Results from base case simulation of P1 with 3 parts

The final two rows report the mean and standard deviation of time between production starts. The theoretical mean and standard deviation of the time between

reorders are 60 and 16.97, respectively. Not surprisingly, the variation of time between production starts is greater than the variation of time between reorders. The average time between production starts determines setup cost. This is Metric 3. The variability of time between production starts is Metric 4.

**Comparison of P5-P7 to Base Case I**

We now turn our attention to evaluating and comparing the different policies. We provide the full set of outputs generated by the simulations that we discuss in an appendix to this chapter.

We begin with a simulation of P5. Before simulating, we must decide how to parameterize the policy. To equalize the setup frequency across policies, we choose a production sequence 1-2-3 with a cycle length of 60 hours. This means that the times between setups should be approximately 60 hours on average. We must also specify an order-up-to level. Ideally, the order-up-to level should be chosen so that the policy will have comparable inventory holding costs and floor space requirements with the base case policy P1. Therefore we set the order-up-to level to the average maximum inventory level of P1 (6,330) plus the average quantity of demand during a production run (1,618), for a total of 7,948.

The results from the simulation are reported in Table 4.5 in the appendix. The average time between production starts and the average inventory level are nearly equal to those in the base case. The policy differs from the base case in three ways. First, there is now variability in the lot size, whereas the base case policy experiences none. The coefficient of variation in production quantity is approximately 0.33. Second, we observe a significant decrease in service, or alternatively, we could conclude that this policy requires additional safety stock to achieve the same level of

service as the base case. The Type 1 measure of service is seen to decrease from 89% to 84%, while the Type 2 measure of service decreases from 98% to 96%. Lastly, we observe a decrease in the variability of time between starts. Overall, this policy appears inferior P1 for this particular set of data since it induces extra variability (in terms of the raw material requirements) and requires additional inventory.

We next consider P6. We choose the same sequence 1-2-3 with production intervals of length 20, to maintain the 60 hour average time between production starts. We determine by simulation that using the same order-up-to level as the one used for P5 results in average inventory levels that are too low. This occurs because the order-up-to level is not always reached before the next setup must occur. Accordingly, the order-up-to levels are increased by 13% (475 parts). The results from the simulation are reported in Table 4.6 in the appendix. As desired, the average time between starts, the average inventory level and the average maximum inventory level are nearly equal to those in the base case. There are three important observations. First, this policy induces variability in the lot size. The amount of variability in production quantity is less than P5, with a coefficient of variation of approximately 0.23. Second, unlike P5, there is no variability in the time between production starts. Thus, in terms of the variability that is induced on the upstream stage(s), P6 eliminates one type of variability while inducing another. Like P5, however, we observe a decrease in service compared to the base case. Since the service achieved is slightly greater than the service of P5, this policy is clearly preferred for this particular set of data.

We next turn our attention to P7. Recall that P7 is the same as P6, except that the "most critical" part is chosen to be produced next, rather than producing in a fixed sequence. For this policy, setups occur at regular intervals. To maintain the same

setup frequency as the base case, we choose the time between setups to be 20 hours. The same order-up-to levels as P5 and P6 are used. We determine by simulation that this order-up-to level results in average inventory levels that are too low. This occurs because the target production quantity is not always achieved before the next setup must occur. Accordingly, we increase the order-up-to levels by 7% (250 parts). The results from the simulation are reported in Table 4.7 in the appendix. Once again, we find that we have approximately equalized setup frequency and inventory levels with the base case. This policy achieves a slightly higher level of service than the base case: a 90% Type 1 service level and a 98.3% Type 2 service level. Furthermore, like P5, this policy induces variability in both the timing and quantity of raw material requirements. The variability in quantity is about half that induced by P6, with a coefficient of variation of approximately 0.19, compared to no variability in the base case. The variability in time between production starts is about 80% that of the base case, but is greater than that induced by P5. In summary, this policy does achieve a significant increase in service over P5 and P6 at the expense of inducing additional variability on the upstream stage(s). When compared to P1, it is not clear whether the slight increase in service offered by P7 justifies the variability in production quantity.

**Comparison of P2-P4 to Base Case I**

We now compare the reorder point-based policies to the base case. We begin with P3. To parameterize this policy, we must choose the values of $Q^-$ and $Q^+$. Given one of these parameters, we will choose the other so that the average lot size is the same as the base case (6,200). We will set $Q^-$ equal to 95% of 6,200. Through numerical integration of the waiting time density for the setup crew, it follows that a value of 6,643 for $Q^+$ will result in the desired average lot size. By simulating this policy, however, we observe that the average inventory levels are higher than the base case

because the leadtimes have been reduced. We therefore decrease the reorder points by about 6% (270 parts); the results from the simulation are reported in Table 4.8 in the appendix. We note that the average lot size achieved is indeed 6,200, and the average maximum inventory level and the average inventory level are approximately equal to those in the base case. As expected, there is an increase in service, but it is very small (around 1% for Type 1 service and a few tenths of a percent for Type 2 service). This improvement in service does not come without some penalty. There is an slight increase in the variability of time between production starts (less than 3%). Further, this policy induces variability in the production quantity, although this variability is not great (the coefficient of variation is about 0.05), and in fact, we know from the value of $Q^+$ that the production quantity will not exceed 6,200 by more than 7.1%. This information can be useful to the upstream stage(s) in production planning and inventory sizing.

In the above experiment we arbitrarily set $Q^-$ to 95% of the target lot size. We briefly consider the impact of reducing $Q^-$ further. Note that if the difference between the target average lot size and $Q^-$ is less than the average waiting time for a setup crew multiplied by the effective production rate, then we cannot choose $Q^+$ to achieve the desired target average lot size. This is because, on average, the setup crew will arrive before the target average lot size is reached, irrespective of how large $Q^+$ becomes. For this particular problem, this imposes the constraint $Q^- \geq 0.926\ Q$. As $Q^-$ approaches this bound, $Q^+$ increases to infinity. A choice of $Q^- = 0.93\ Q$ results in $Q^+$ equal to 7,551, which is 22% larger than our target lot size of 6,200. We cannot reduce $Q^-$ much further than this. After adjusting the reorder points (by about 7.5% versus the base case), we simulated this policy; the results are reported in Table 4.9 in the appendix. There is a further increase less than 1% in terms of Type 1 service, and perhaps another 0.1% increase in Type 2 service. The penalties are, of course,

more severe. We have already seen above that the maximum production quantity can exceed the mean by as much as 22%. There is also an increase in the variability of time between production starts (about 7% over the base case).

We conclude this subsection with a comparison of the order-up-to policies P2 and P4. P2 is the most difficult to parameterize for normative purposes, because setting the order-up-to level to achieve a desired average inventory level requires an estimate of the average leadtime, which is itself a function of the order-up-to level. Similar to our calculations for P5, we set the order-up-to level equal to the average maximum inventory level of P1 plus the average quantity of demand during a production run plus the reorder point. We set the reorder point to 3,000 (2.5% less than the base case) and using an estimate of 12 hours for the leadtime, we obtain a order-up-to level of 7,960. The results from the simulation are reported in Table 4.10 in the appendix. Our estimate of 12 hours is very good, so we do achieve an average inventory level, average maximum inventory level and setup frequency equal to that of the base case. There is a very slight increase in Type 1 service, perhaps 0.3%. The fact that there is not a significant increase is somewhat disappointing, since P2 induces variability in the production quantity (the coefficient of variation is about 0.21). However, there is a 15% decrease in the variability of time between production starts. Overall, this policy appears inferior to the base case for this particular set of data, since it introduces extra variability on the upstream stage(s) with almost no benefit in terms of improved service.

Finally, we consider P4. During any production cycle, we will stop production if the setup crew arrives when the difference between the number of parts produced and the target lot size is less than 310 parts. The choice of 310 comes from 5% of 6,200, which is intended to mirror our simulation of P3 with $Q^- = 0.95\,Q$. By simulation,

we determine that the same order-up-to level used for P2 must be reduced by another 250 parts (about 6%). The results from the simulation are reported in Table 4.11 in the appendix. The service achieved is the highest of any policy we have simulated, with an average of 91% Type 1 service and 98.7% Type 2 service. In terms of variability, the policy is roughly equivalent to P2, with slightly lower variability in production quantity and slightly higher variability in the time between production starts. When we compare this policy to P3 at $Q^{-} = 0.93\,Q$, however, we observe that the service of each is nearly the same, while P4 induces far more variability in the production quantity.

Although we have made many interesting observations, we defer any additional remarks until we have obtained further results.

**Base Case II: Inputs**

To broaden our perspective, we now consider a different metal stamping line. This second line manufactures nine parts. Unlike the previous case where each part had the same parameters, on this line the demand rates for the parts differ. The first four parts each have demand of 600/week. The remaining five have demands of 1,900, 3,700, 6,150, 7,720 and 8,000 parts per week. Although this will make this case somewhat harder to analyze, this production line is not atypical for a GM stamping plant.

For each part, the coefficient of variation in demand over a week is 0.2. When this stamping line is operating, it produces parts at 450 per hour. It fails on average once per hour and requires 15 minutes on average to repair. These parts require 30 minutes to be setup once a setup crew arrives. There are 120 hours available for production in a week.

As before, we will simulate P1 for the base case. This requires the specification of lot sizes and reorder points. As before, we will not perform a detailed analysis using cost data. Instead, suppose we know that the typical EOQ considerations are minimized for the highest volume part (8,000/week) when the setup frequency is approximately twice per week, for a lot size of 4,000. If all of the parts have similar inventory holding costs, then we can determine the optimal setup frequencies for the remaining parts using the familiar square root formula. In ascending order of demand, the setup frequencies for the other eight parts are 0.55, 0.55, 0.55, 0.55, 0.97, 1.36, 1.75, and 1.98. For simplicity in scheduling, we round these to (0.5, 0.5, 0.5, 0.5, 1.0, 1.0, 2.0, 2.0). The fact that the setup frequencies are integer multiples of one another will aid us in the construction of schedules. Although these setup frequencies may not be optimal, it is not our concern to find an optimal parameterization, but rather, to select a set of lot sizes that are near optimal while also reflective of what would be encountered in practice. We feel that the 1:4 ratio of setup frequencies is reasonable given the 1:13 ratio between the smallest and largest demand rates. Excluding waiting times for setup crews, the above parameters result in a machine utilization of 87%.

We again wish to set the reorder points to achieve high service levels (e.g., 98%). As before, an iterative process was used to set the reorder point. For each part, the reorder point is set to cover the mean plus two standard deviations of demand over the estimated leadtime. A leadtime estimate of 10 hours was chosen, resulting in a reorder points of 119 for the first four parts, and 378, 736, 1,223, 1,535 and 1,590 for the other five.

Lastly, we parameterize the setup crew in the same way as before. The input data for the simulation are summarized in Table 3.

| Part | Demand rate | Demand std. dev. | Lot size | Reorder point |
|---|---|---|---|---|
| 1-4 | 600 | 120 | 1,200 | 119 |
| 5 | 1,900 | 380 | 1,900 | 378 |
| 6 | 3,700 | 740 | 3,700 | 736 |
| 7 | 6,150 | 1,230 | 3,075 | 1,223 |
| 8 | 7,720 | 1,544 | 3,860 | 1,535 |
| 9 | 8,000 | 1,600 | 4,000 | 1,590 |

| | |
|---|---|
| Production rate = 450/hour | # of setup crews = 1 |
| Setup time = 0.5 hour | # of machines/crew = 5 |
| MTBF = 1 hour | Arrival rate of requests |
| MTTR = 0.25 hour | for setup = 1/6 hours |
| SAA: 80% | Mean service rate = 1 hour |
| Hours per week = 120 | Setup crew utilization: 83.33% |

**Table 4.3** Data for Base Case II

**Base Case II: Results**

Our study of the nine part line will be conducted in much the same way as before. P1 will serve as our base case, and inferences will be drawn by normalizing the policies to this base case and comparing them against one another. The production system was run for 20 simulated weeks in order for a "steady state" to be reached, and then statistics were collected for 100 simulated years (5,200 weeks). This required 7 minutes and 51 seconds on a Power Macintosh 7100/80 running in native mode.

The resulting statistics are reported in Table 4.4. At this point in time, we note only that our choice of reorder point does result in a high level of service (at least 98% Type 2 service). However, there is more than a 1% difference in Type 2 service

| | Part 1 | Part 2 | Part 3 | Part 4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|---|---|---|
| Average lotsize | 1200 | 1200 | 1200 | 1200 | 1900 | 3700 | 3075 | 3860 | 4000 |
| Standard deviation | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Avg wait for setup | 1.14 | 1.20 | 1.17 | 1.17 | 1.15 | 1.20 | 1.15 | 1.15 | 1.17 |
| Standard deviation | 1.56 | 2.22 | 1.58 | 1.58 | 1.56 | 1.59 | 1.56 | 1.57 | 1.57 |
| Avg leadtime | 12.52 | 12.12 | 12.58 | 12.28 | 11.22 | 9.92 | 9.62 | 8.81 | 8.74 |
| Standard deviation | 9.68 | 9.86 | 9.49 | 9.76 | 9.29 | 8.43 | 8.41 | 8.12 | 7.93 |
| Avg production time | 3.41 | 3.35 | 3.36 | 3.33 | 5.29 | 10.33 | 8.58 | 10.74 | 11.16 |
| Standard deviation | 1.22 | 0.80 | 0.99 | 0.69 | 1.06 | 1.84 | 1.65 | 1.39 | 1.72 |
| Type 1 Service level | 83.43% | 83.59% | 83.17% | 83.79% | 86.53% | 89.18% | 89.18% | 90.60% | 90.90% |
| Type 2 Service level | 99.34% | 99.28% | 99.30% | 99.31% | 98.88% | 99.20% | 98.27% | 98.42% | 98.69% |
| Avg max inventory | 1241.06 | 1241.18 | 1239.00 | 1240.42 | 2017.73 | 3813.52 | 3359.12 | 4122.76 | 4252.61 |
| Avg min inventory | 58.27 | 57.94 | 55.66 | 57.14 | 202.05 | 428.23 | 724.87 | 956.81 | 999.37 |
| Avg inventory | 649.67 | 649.56 | 647.33 | 648.78 | 1109.89 | 2120.87 | 2042.00 | 2539.78 | 2625.99 |
| Avg time betwn starts | 240.50 | 238.61 | 239.70 | 239.65 | 119.90 | 120.31 | 59.83 | 59.84 | 59.93 |
| Standard deviation | 35.08 | 37.31 | 36.39 | 35.94 | 26.56 | 26.63 | 19.57 | 19.21 | 18.99 |

**Table 4.4** Results from base case simulation of P1 with 9 parts

across the parts. This is expected, since each part experiences a different leadtime distribution (e.g., the parts with longer average production times have shorter average waits in queue).

**Comparison of P5-P7 to Base Case II**

We now compare the different policies to our base case. We will focus our discussion on where the results obtained agree or differ from those obtained for the 3 part line.

We begin with a simulation of P5. Before simulating, we must choose a production sequence. Since the minimum production frequency is 1/2 weeks, a natural production cycle length is 2 weeks. We choose the sequence 1-2-7-8-9-5-6-7-8-9-3-4-7-8-9-5-6-7-8-9. For parts 7, 8 and 9, the average time between production starts will not be the same. Unfortunately, it is not possible to exactly equalize the times between production starts for all of the parts. For the chosen sequence, the difference in the time between production starts is only about 20%. As before, we set the order-up-to levels so that the average inventory level will be the same as the base case. The results of the simulation are reported in Table 4.12 in the appendix. For simplicity, only the average of the statistics for parts 1-4 are reported. The setup frequency, average inventory level and average maximum inventory level are in close agreement with the base case. P5 performs much worse on this line relative to the base case than it did on the 3 part line. Compared with the base case, there is an 8%-13% decrease in Type 1 service and a 9%-14% decrease in Type 2 service. There is variability in production quantity; the coefficient of variation ranges from 0.3 to 0.7. Lastly, the variability in time between production starts is 70 to 105% higher than the base case.

Next, we simulate P6. The same production sequence and order-up-to levels from P5 are used. The cycle length of 240 hours is allocated to the parts in proportion to their demand rates. If a part is setup more than once, then its production time is allocated equally among the setups for that part. We determine by simulation that the order-up-to levels must be increased (between 1% and 4%, depending on the part) to achieve the same average inventory level as the base case. The results with these adjusted order-up-to levels are reported in Table 4.13 in the appendix. The service levels achieved are much higher than P5 (4%-6% greater Type 1 and 6%-14% greater Type 2 service) but are still much lower than the base case (3%-7% less Type 1 and 2%-3% less Type 2 service). The variability in production quantity is less than half that of P5, and P6 also has almost no variability in time between production starts. Thus, as in the 3 part line, P6 is clearly preferred to P5.

To evaluate P7, we use the same order-up-to levels as P5 and P6. The machine will be setup every 12 hours so that there are 10 setups/week, as in the base case. The results after the usual adjustment in order-up-to levels (2% to 26%, depending on the part) are reported in Table 4.14 in the appendix. Although the setup frequency is maintained at 10 setups/week, the average time between production starts for the individual parts is not the same as the base case. The time between starts is longer for parts 1-5 and 7, and shorter for parts 6, 8 and 9. This is because the 12 hour production time is almost never completely used for some of the parts (1-5), yet it is often insufficient to bring the inventory level back up to the order-up-to level for some of the others (e.g., part 9 uses all 12 hours 67% of the time). This reveals a flaw in this operating policy: production capacity can be wasted when the parts require substantially different amounts of production time. As with the three part line, the service level achieved by P7 is higher than both P5 and P6 but not superior to the base case (service improves slightly for some of the parts but degrades slightly for

others, compared to the base case). The variability in the time between production starts is equivalent to the base case (and so is much less than P5), and the variability in production quantity much less than P5 or P6. Note, however, that P6 does enjoy almost no variability in time between production starts.

**Comparison of P2-P4 to Base Case II**

We now compare the reorder point-based policies to the base case. We begin with P3. To parameterize this policy, we must choose the values of $Q^-$ and $Q^+$ for each part. We will set $Q^-$ equal to 95% of the average lot size of the base case, and choose $Q^+$ so that the target average lot size is achieved. The results of the simulation after adjustment of the reorder points (1%-4% of the base case) are presented in Table 4.15 in the appendix. The average time between production starts, and average maximum inventory levels and average inventory levels are in close agreement with the base case. Compared to the base case, there is a slight rise in service (less than a 2% increase in Type 1 service and only a few tenths of a percent increase in Type 2 service), and a slight increase in the variability of time between production starts (1%-4%). The variability in production quantity is small (coefficient of variation of about 0.04). In fact, $Q^+$, the maximum number of parts that will be produced, is only 3 to 5% higher than the target average lot sizes. This suggests that we can reduce $Q^-$ further.

In fact, using the calculations that were described for the 3 part line, we determine that $Q^-$ can not be reduced further than 89% of the target lot sizes. We select 90% since $Q^+$ becomes unrealistically large as $Q^-$ approaches this lower bound. The results from this experiment are reported in Table 4.16 in the appendix. The Type 1 service level increases 1%-2.5% and the Type 2 service level increases less than 0.5%, compared to the 95% case. However, the variability of time between production

starts increases as much as 12% over the 95% case, the variability in production quantity is 200%-300% higher than the 95% case, and the maximum number of parts produced can be as much as 33% higher than the target average lot size. Despite this, the variability in production quantity is still less than P5 or P6. In summary, this 90% case achieves some improvement in service compared to the 95% case, but induces significantly more variability on the upstream stage(s). The policy clearly outperforms P5, however, since it offers more service with less variability. Comparison to P7 is inconclusive: depending on the part, the variability in production quantity may be more or less, and the service level may be more or less.

We next consider P2. We set the reorder points to the same levels as in the base case, and use the iterative procedure described for the 3 part line to obtain an estimate of the leadtime. At the last iteration we also adjust the order-up-to levels. The results from the final simulation are reported in Table 4.17 in the appendix. Our estimates of the leadtime are very good, so we do achieve average inventory levels, average maximum inventory levels and setup frequencies equal to those of the base case. There is a slight decrease in service, as much as 0.5% for Type 1 service, and 2% for Type 2 service. The fact that there is no improvement in service is somewhat disappointing, since P2 induces variability in the production quantity (the coefficient of variation varies from 0.06 to 0.20). However, there is a slight decrease for some of the parts (around 10%) in the variability of time between production starts. The slightly lower service and the variability in the production quantity appear to make this policy inferior to the base case for this particular set of data.

Finally, we consider P4. We choose the parameters for this policy so that we will stop production if the setup crew arrives when the difference between the number

of parts produced and the target production quantity is less than 5% of the average lot size. The choice of 5% is intended to allow comparison with our simulation of P3 with $Q^- = 0.95\,Q$. We use the same order-up-to levels that were used for P2, adjusted in the usual way. The results from the simulation are reported in Table 4.18 in the appendix. The waiting time for a setup crew is indeed comparable to P3 with $Q^- = 0.95\,Q$, however, the service achieved is less and the variability in the production quantity is far greater. In terms of variability, the policy is roughly equivalent to P2, with higher variability for some parts and lower variability for others. Compared to the base case, there is a 1% increase in Type 1 service and no improvement in Type 2 service. If one is willing to trade-off variability in production quantity in exchange for higher service, P3 performs much better than P4 for this particular line because it achieves the same service levels with less variability.

**Impact of machine utilization on 9 part line**

Our study thus far has shown the reorder point-based policies P1-P4 to provide superior levels of service compared to those with a fixed production sequence (P5-P6). We now briefly consider increasing the utilization of the machine and examining the impact on the relative performance of P1, P5, P6 and P7. It is important to note that the machine utilization is greater for the parts that must wait for a setup crew. Therefore, at very high machine utilizations it could happen that a policy that waits for setup crews has a machine utilization above 100% and is therefore infeasible, while another policy that does not wait for setup crews (i.e., P6 or P7) will have a machine utilization below 100%.

The utilization of the machine is increased by reducing the production rate from 450 parts/hour to 390 parts/hour. This results in a utilization of almost 94% for P1 and

P5, and a utilization of 84% for P6 and P7. The 94% machine utilization level was chosen because it is representative of the highest utilization that a GM planner would consider for a metal stamping line.

The results of the four simulations are reported in Tables 4.19-4.22 in the appendix. Even at such a high machine utilization, P1 still achieves a higher level of Type 2 service than the other three policies. The results for P7 warrant further comment. For some of the parts, the 12 hour production interval is rarely sufficient to produce the number of parts dictated by the order-up-to level. As a result, the targeted average maximum inventory levels are not achieved for some of the parts. The average inventory is as much as 27% lower than P1, but increasing the order-up-to level further does not affect this. Thus, while the performance of P1 can be improved by increasing the reorder points, this does not in general hold true for P7.

Also of interest is the fact that the variability of time between production starts for P1 does not, on the whole, increase as a result of the increase in machine utilization.

**Impact of waiting time for setup crews on 9 part line**

In the experiments above, we have seen that P3 and P4 can be effective in improving the level of service of P1 and P2. Clearly, as the percentage of machine hours spent waiting for crews increases, the benefits of P3 and P4 will increase rapidly. We now briefly consider *decreasing* the waiting time for setup crews and examining the impact on the relative performance of P1 and P3.

We reduce the waiting time for setup crews by increasing the service rate from 1.0 to 1.4. This decreases the average waiting time for a setup crew from 1.16 to 0.56, and reduces the setup crew utilization from 83% to about 60%. For P3, we set $Q$ equal to

95% of the average lot size of the base case, and choose $Q^+$ so that the target average lot size is achieved. The results of the simulations of P1 and P3 are presented in Tables 4.23 and 4.24 in the appendix. The average and maximum inventory levels of the two policies are within about 1% of one another, and the setup frequencies are in close agreement. Compared with P1, the service level provided increases 1.3%-2.5% in terms of Type 1 service and as much as 0.3% in terms of Type 2 service. The coefficient of variation of production quantity for P3 ranges from 0.05 to 0.08. However, the maximum number of parts produced can be as much as 135% of the target average lot size, depending on the part. In this way, the 95% case acts more like the 90% case now that the setup crew utilization is lower.

We therefore consider increasing $Q^-$ equal to 97.5% of the average lot size of the base case, and adjusting $Q^+$ accordingly. The resulting simulation is reported in Table 4.25 in the appendix. The maximum number of parts produced is now at most 4% of the target average lot size, and the coefficient of variation of production quantity ranges from 0.02 to 0.03. However, compared with P1, the service level provided now increases 0.5%-1% in terms of Type 1 service, while the improvement in terms of Type 2 service is not measurable.

As we expected, the benefits that are achievable from P3 are reduced when the waiting time for setup crews is reduced. Furthermore, we have seen that the penalties for achieving these benefits (in terms of the increase in $Q^+$) can increase rapidly as the waiting time for setup crews is reduced.

## 4.6 Conclusions

We now summarize some of the more important observations we have made based on the experiments of the previous section. The reader should be aware that any conclusions that we reach are based on two very specific examples described in the previous section, and is advised against drawing inferences to production systems that differ greatly from the ones that we have studied.

With this caveat in mind, we begin with the observation that P1 and P3 are highly desirable operating policies for these production lines. P1 induces no variability in the production quantity, and provides relatively high levels of service. P3 is a *generalization* of P1, which in the presence of moderate setup crew wait times can provide some additional service at the expense of variability in the production quantity. However, P3 results in a known upper bound on the production quantity, which can assist in the coordination of production and inventory levels between production stages. The amount of variability in production quantity can be chosen by the decision maker to optimize the tradeoff between this variability and the benefits associated with less waiting for setup. It is not our intent to describe how to optimize a multi-echelon production/inventory system. We remark only that the tradeoff between safety stock and the amount of variability induced on the upstream stage(s) is a function of the relative cost of holding inventory at the two stages, as well as the relative production capacities at the two stages.

P2 and P4 do not seem to perform as well as P1 and P3. P2 attains a slightly higher level of service in the three part example but does worse in the nine part example. P2 also induces variability in both the timing and quantity of raw material demands. P4 induces much more variability than P3 for the same level of service. We also

note that P2 and P4 have greater information systems requirements than either P1 or P3.

In all of our experiments, P5 consistently performs poorly. It provides relatively low service yet induces variability in both the timing and quantity of raw material demands. This is an important result, in part because P5 has received much attention in the literature.

P6 eliminates or virtually eliminates variability in the timing of raw material demands, so it is also desirable for multi-stage coordination. However, it provides less service for the same amount of inventory when compared to the reorder point-based policies P1-P4.

In the presence of demand variability, P5 and P6 are at a disadvantage because production will proceed in a fixed sequence even in situations when it is clearly non-optimal to do so. In contrast, consider a situation where the demand rates of the products are equal and there is little or no variability in demand. In such a case, even if setup and production times are highly variable, a fixed sequence will not be a disadvantage. Further, our study has assumed that setup costs were sequence independent. In an environment where the sequence of setups is a serious cost concern, P5 and P6 may be the only reasonable alternatives. For production lines similar to the ones we have studied, P6 is clearly preferred.

In an environment where machine utilization is very high and there is substantial waiting for setup crews, P6 and P7 may be the only feasible policies. For both of our production lines, P7 provides a higher level of service than P6, but P7 induces variability in the timing of raw material demand. P7 is indeed a strange policy. As

we increase the utilization on the nine part line where the parts have very different parameters, the behavior of P7 becomes awkward and not fully controllable. However, this might not be the case if setups were very frequent, e.g., in the case of parts with extremely high inventory holding costs.

The relative success of the reorder point-based policies suggests that they are worthy of further examination. We believe that the reorder point-based policy that minimizes the amount of inventory required to achieve a given service level will dynamically determine production quantities depending on the inventories of all of the items. Further improvement may be possible if reorder points are also determined dynamically. Note, however, that while such policies may improve the performance of the single stage, they will induce additional variability on the upstream stage(s) compared to a static, fixed lot size policy. This effect, together with the difficulty of implementing such a dynamic policy, makes such policies seem less promising, but still worthy of further exploration.

Queue discipline is another aspect of the reorder point-based policies that may be worthy of re-examination. The first-in-first-out queue discipline in P1-P4 is simple but clearly flawed, since fluctuations in demand that occur while the parts wait in queue can cause imbalances in the relative priorities of producing the different parts. For this reason, a policy such as P8 may be particularly interesting to explore further. We do not expect that changes in the queue discipline will induce additional variability on the upstream stage(s).

We began this section with a warning about the generality of these results. It is easy to imagine that, in a production environment that is vastly different from the one we have studied, the conclusions reached could be quite different. For example, if

the production process were extremely variable but the variability in the demand process were minimal, we might reach different conclusions. It has not been our intent to make broad generalizations, but rather to begin to provide a relative understanding of the policies we have chosen to study by offering a framework and methodology for comparison and some interesting empirical results to highlight the differences and sensitivities of the various policies. We believe that there is still much to be learned about the policies that we have studied, as well as potential for the development and analysis of new operating policies that offer superior performance.

# Appendix: Output from simulations

| | Part 1 | Part 2 | Part 3 |
|---|---|---|---|
| Average lotsize | 6208.92 | 6205.44 | 6182.83 |
| Standard deviation | 2048.88 | 2107.74 | 2257.88 |
| Avg wait for setup | 1.19 | 1.14 | 1.16 |
| Standard deviation | 1.61 | 1.55 | 1.56 |
| Avg production time | 15.69 | 15.71 | 15.64 |
| Standard deviation | 5.37 | 5.59 | 5.91 |
| Type 1 Service level | 83.59% | 84.16% | 83.52% |
| Type 2 Service level | 96.18% | 95.98% | 95.13% |
| Avg max inventory | 6327.03 | 6313.79 | 6315.29 |
| Avg min inventory | 1738.93 | 1742.41 | 1765.02 |
| Avg inventory | 4032.98 | 4028.10 | 4040.15 |
| Avg time betwn starts | 60.00 | 60.00 | 60.00 |
| Standard deviation | 9.81 | 11.57 | 13.93 |

**Table 4.5** Results from simulation of P5 with 3 parts

| | Part 1 | Part 2 | Part 3 |
|---|---|---|---|
| Average lotsize | 6203.46 | 6219.22 | 6195.10 |
| Standard deviation | 1411.70 | 1411.13 | 1413.48 |
| Cycles at capacity | 31.8% | 32.4% | 31.4% |
| Avg production time | 15.64 | 15.68 | 15.61 |
| Standard deviation | 3.58 | 3.58 | 3.58 |
| Type 1 Service level | 86.72% | 86.37% | 87.20% |
| Type 2 Service level | 96.38% | 95.91% | 96.31% |
| Avg max inventory | 6339.64 | 6292.77 | 6358.35 |
| Avg min inventory | 1759.43 | 1706.07 | 1768.55 |
| Avg inventory | 4049.53 | 3999.42 | 4063.45 |
| Avg time betwn starts | 60.00 | 60.00 | 60.00 |
| Standard deviation | 0 | 0 | 0 |

**Table 4.6** Results from simulation of P6 with 3 parts

| | Part 1 | Part 2 | Part 3 |
|---|---|---|---|
| Average lotsize | 6220.61 | 6203.46 | 6225.27 |
| Standard deviation | 1164.83 | 1170.69 | 1187.35 |
| Cycles at capacity | 24.3% | 24.1% | 24.6% |
| Avg production time | 15.69 | 15.64 | 15.67 |
| Standard deviation | 3.02 | 3.04 | 3.08 |
| Type 1 Service level | 90.19% | 90.72% | 90.44% |
| Type 2 Service level | 98.35% | 98.39% | 98.28% |
| Avg max inventory | 6337.03 | 6336.29 | 6333.35 |
| Avg min inventory | 1734.16 | 1757.74 | 1725.25 |
| Avg inventory | 4035.60 | 4047.02 | 4029.30 |
| Avg time betwn starts | 60.06 | 59.76 | 60.18 |
| Standard deviation | 14.76 | 14.54 | 14.92 |

**Table 4.7** Results from simulation of P7 with 3 parts

| | Part 1 | Part 2 | Part 3 |
|---|---|---|---|
| Average lotsize | 6200.05 | 6201.51 | 6204.23 |
| Standard deviation | 314.23 | 316.14 | 317.48 |
| Avg wait for setup | 0.86 | 0.85 | 0.87 |
| Standard deviation | 1.50 | 1.50 | 1.71 |
| Avg leadtime | 10.51 | 10.44 | 10.33 |
| Standard deviation | 8.74 | 8.58 | 8.64 |
| Avg production time | 15.31 | 15.31 | 15.29 |
| Standard deviation | 1.35 | 1.61 | 1.34 |
| Type 1 Service level | 90.44% | 90.88% | 90.93% |
| Type 2 Service level | 98.36% | 98.67% | 98.50% |
| Avg max inventory | 6315.96 | 6356.62 | 6372.50 |
| Avg min inventory | 1701.55 | 1738.98 | 1732.77 |
| Avg inventory | 4008.75 | 4047.80 | 4052.64 |
| Avg time betwn starts | 59.88 | 60.03 | 60.12 |
| Standard deviation | 19.09 | 18.98 | 19.27 |

**Table 4.8** Results from simulation of P3 with 3 parts, minimum fraction = 95%

| | Part 1 | Part 2 | Part 3 |
|---|---|---|---|
| Average lotsize | 6203.74 | 6209.96 | 6210.76 |
| Standard deviation | 542.49 | 549.43 | 551.40 |
| Avg wait for setup | 0.56 | 0.55 | 0.54 |
| Standard deviation | 1.34 | 1.32 | 1.31 |
| Avg leadtime | 9.82 | 9.83 | 9.85 |
| Standard deviation | 8.35 | 8.38 | 8.29 |
| Avg production time | 15.48 | 15.49 | 15.53 |
| Standard deviation | 1.65 | 1.68 | 2.08 |
| Type 1 Service level | 91.33% | 91.42% | 91.34% |
| Type 2 Service level | 98.60% | 98.56% | 98.74% |
| Avg max inventory | 6347.87 | 6335.95 | 6341.21 |
| Avg min inventory | 1736.40 | 1725.26 | 1735.37 |
| Avg inventory | 4042.14 | 4030.60 | 4038.29 |
| Avg time betwn starts | 60.08 | 59.84 | 60.07 |
| Standard deviation | 19.72 | 19.55 | 19.71 |

**Table 4.9** Results from simulation of P3 with 3 parts, minimum fraction = 93%

| | Part 1 | Part 2 | Part 3 |
|---|---|---|---|
| Average lotsize | 6207.25 | 6211.49 | 6218.72 |
| Standard deviation | 1326.10 | 1335.01 | 1323.13 |
| Avg wait for setup | 1.17 | 1.17 | 1.16 |
| Standard deviation | 1.58 | 1.60 | 1.60 |
| Avg leadtime | 12.07 | 12.01 | 12.03 |
| Standard deviation | 10.17 | 10.23 | 10.13 |
| Avg production time | 15.65 | 15.67 | 15.71 |
| Standard deviation | 3.58 | 3.67 | 3.66 |
| Type 1 Service level | 89.47% | 89.33% | 89.12% |
| Type 2 Service level | 98.16% | 98.14% | 98.17% |
| Avg max inventory | 6341.57 | 6318.23 | 6338.71 |
| Avg min inventory | 1747.76 | 1743.51 | 1736.29 |
| Avg inventory | 4044.66 | 4030.87 | 4037.50 |
| Avg time betwn starts | 60.06 | 59.92 | 60.09 |
| Standard deviation | 15.73 | 15.55 | 15.68 |

**Table 4.10** Results from simulation of P2 with 3 parts

| | Part 1 | Part 2 | Part 3 |
|---|---|---|---|
| Average lotsize | 6196.89 | 6182.90 | 6172.99 |
| Standard deviation | 1185.19 | 1210.00 | 1178.05 |
| Avg wait for setup | 0.87 | 0.88 | 0.86 |
| Standard deviation | 1.51 | 1.52 | 1.49 |
| Avg leadtime | 9.83 | 9.85 | 9.72 |
| Standard deviation | 8.68 | 8.84 | 8.64 |
| Avg production time | 15.30 | 15.27 | 15.21 |
| Standard deviation | 3.34 | 3.30 | 3.22 |
| Type 1 Service level | 91.25% | 91.21% | 91.87% |
| Type 2 Service level | 98.76% | 98.59% | 98.78% |
| Avg max inventory | 6343.10 | 6343.91 | 6346.89 |
| Avg min inventory | 1724.64 | 1728.22 | 1744.85 |
| Avg inventory | 4033.87 | 4036.07 | 4045.87 |
| Avg time betwn starts | 59.99 | 59.94 | 59.76 |
| Standard deviation | 16.25 | 16.26 | 16.24 |

**Table 4.11** Results from simulation of P4 with 3 parts

| | Parts 1-4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|
| Average lotsize | 1200.34 | 1900.98 | 3710.91 | 3063.80 | 3855.38 | 4013.72 |
| Standard deviation | 354.08 | 915.80 | 1846.83 | 1947.41 | 2638.91 | 2341.04 |
| Avg wait for setup | 3.36 | 5.28 | 10.32 | 8.56 | 10.75 | 11.16 |
| Standard deviation | 1.38 | 2.66 | 5.34 | 5.68 | 7.50 | 6.63 |
| Avg production time | 1.18 | 1.13 | 1.18 | 1.19 | 1.14 | 1.17 |
| Standard deviation | 1.59 | 1.55 | 1.60 | 1.61 | 1.56 | 1.58 |
| Type 1 Service level | 70.13% | 75.62% | 76.72% | 81.37% | 82.21% | 80.02% |
| Type 2 Service level | 90.29% | 84.91% | 84.55% | 83.56% | 82.34% | 85.50% |
| Avg max inventory | 1239.91 | 2018.37 | 3808.40 | 3360.18 | 4119.41 | 4248.41 |
| Avg min inventory | 56.33 | 200.59 | 418.99 | 732.96 | 957.41 | 979.02 |
| Avg inventory | 648.12 | 1109.48 | 2113.69 | 2046.57 | 2538.41 | 2613.72 |
| Avg time betwn starts | 240.00 | 120.00 | 120.00 | 60.00 | 60.00 | 60.00 |
| Standard deviation | 61.33 | 52.24 | 54.64 | 34.12 | 37.41 | 30.62 |

**Table 4.12** Results from simulation of P5 with 9 parts

| | Parts 1-4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|
| Average lotsize | 1201.20 | 1898.40 | 3724.91 | 3074.68 | 3863.28 | 4013.05 |
| Standard deviation | 172.79 | 368.14 | 725.52 | 837.26 | 1057.28 | 1092.61 |
| Cycles at capacity | 6.5% | 7.5% | 6.0% | 13.8% | 12.8% | 13.8% |
| Avg production time | 3.32 | 5.29 | 10.34 | 8.53 | 10.73 | 11.16 |
| Standard deviation | 0.67 | 1.19 | 2.21 | 2.43 | 3.01 | 3.12 |
| Type 1 Service level | 76.43% | 79.96% | 81.37% | 85.62% | 86.32% | 85.83% |
| Type 2 Service level | 96.22% | 96.11% | 96.31% | 95.94% | 96.30% | 96.10% |
| Avg max inventory | 1241.00 | 2017.94 | 3824.97 | 3361.35 | 4123.48 | 4262.04 |
| Avg min inventory | 56.66 | 201.83 | 416.78 | 722.65 | 956.09 | 989.93 |
| Avg inventory | 648.83 | 1109.88 | 2120.88 | 2042.00 | 2539.78 | 2625.99 |
| Avg time betwn starts | 240 | 120 | 120 | 60 | 60 | 60 |
| Standard deviation | 0 | 0 | 0 | 6.16 | 6.16 | 6.16 |

**Table 4.13** Results from simulation of P6 with 9 parts

| | Parts 1-4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|
| Average lotsize | 1202.24 | 1910.18 | 3746.12 | 3142.71 | 3825.30 | 3910.69 |
| Standard deviation | 79.61 | 235.69 | 367.16 | 669.70 | 541.18 | 491.97 |
| Cycles at capacity | 0.1% | 0.1% | 31.1% | 13.5% | 54.5% | 62.4% |
| Avg production time | 3.35 | 5.31 | 10.39 | 8.73 | 10.58 | 10.80 |
| Standard deviation | 0.66 | 1.00 | 1.13 | 1.98 | 1.42 | 1.24 |
| Type 1 Service level | 88.12% | 89.60% | 88.88% | 90.08% | 89.44% | 89.36% |
| Type 2 Service level | 99.27% | 98.91% | 98.89% | 98.17% | 98.03% | 98.02% |
| Avg max inventory | 1253.43 | 2060.45 | 3891.47 | 3499.58 | 4134.21 | 4197.09 |
| Avg min inventory | 68.10 | 235.54 | 456.15 | 802.64 | 986.42 | 1010.80 |
| Avg inventory | 660.77 | 1148.00 | 2173.81 | 2151.11 | 2560.32 | 2603.95 |
| Avg time betwn starts | 240.00 | 120.65 | 121.60 | 61.35 | 59.58 | 58.59 |
| Standard deviation | 34.73 | 25.09 | 26.06 | 18.53 | 19.46 | 19.57 |

**Table 4.14** Results from simulation of P7 with 9 parts

| | Parts 1-4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|
| Average lotsize | 1199.67 | 1901.71 | 3699.64 | 3075.09 | 3860.92 | 3999.16 |
| Standard deviation | 47.72 | 77.99 | 167.26 | 134.59 | 175.70 | 184.55 |
| Avg wait for setup | 1.02 | 1.03 | 1.05 | 1.04 | 1.04 | 1.06 |
| Standard deviation | 1.51 | 1.54 | 1.51 | 1.54 | 1.52 | 1.51 |
| Avg leadtime | 11.02 | 10.01 | 8.85 | 8.53 | 7.79 | 7.62 |
| Standard deviation | 9.19 | 8.62 | 7.90 | 7.89 | 7.47 | 7.44 |
| Avg production time | 3.25 | 5.17 | 10.05 | 8.31 | 10.43 | 10.81 |
| Standard deviation | 0.83 | 1.35 | 2.30 | 1.80 | 1.84 | 1.78 |
| Type 1 Service level | 83.41% | 87.04% | 89.78% | 89.97% | 91.63% | 92.24% |
| Type 2 Service level | 99.32% | 99.01% | 99.28% | 98.44% | 98.64% | 98.77% |
| Avg max inventory | 1237.77 | 2016.77 | 3812.94 | 3347.23 | 4122.69 | 4251.10 |
| Avg min inventory | 54.28 | 197.27 | 420.32 | 698.54 | 927.08 | 975.61 |
| Avg inventory | 646.03 | 1107.02 | 2116.63 | 2022.88 | 2524.88 | 2613.35 |
| Avg time betwn starts | 240.24 | 119.81 | 120.36 | 59.64 | 59.87 | 59.95 |
| Standard deviation | 37.67 | 27.10 | 26.67 | 19.33 | 19.43 | 19.30 |

**Table 4.15** Results from simulation of P3 with 9 parts, minimum fraction = 95%

| | Parts 1-4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|
| Average lotsize | 1199.23 | 1902.68 | 3703.24 | 3082.74 | 3865.59 | 4008.84 |
| Standard deviation | 101.89 | 172.47 | 437.29 | 327.43 | 473.54 | 511.48 |
| Avg wait for setup | 0.64 | 0.63 | 0.72 | 0.68 | 0.76 | 0.80 |
| Standard deviation | 1.33 | 1.32 | 1.37 | 1.34 | 1.37 | 1.42 |
| Avg leadtime | 9.70 | 8.91 | 7.97 | 7.51 | 6.88 | 6.86 |
| Standard deviation | 8.36 | 8.01 | 7.60 | 7.36 | 6.92 | 6.90 |
| Avg production time | 3.17 | 5.04 | 10.14 | 8.28 | 10.63 | 11.04 |
| Standard deviation | 0.76 | 1.11 | 2.36 | 1.76 | 2.48 | 2.24 |
| Type 1 Service level | 85.22% | 88.19% | 91.08% | 91.51% | 92.85% | 93.44% |
| Type 2 Service level | 99.45% | 99.16% | 99.38% | 98.64% | 98.97% | 98.93% |
| Avg max inventory | 1237.82 | 2022.30 | 3813.48 | 3369.61 | 4132.85 | 4264.00 |
| Avg min inventory | 54.55 | 198.22 | 427.25 | 710.13 | 949.70 | 995.70 |
| Avg inventory | 646.18 | 1110.26 | 2120.37 | 2039.87 | 2541.27 | 2629.85 |
| Avg time betwn starts | 239.42 | 120.25 | 119.74 | 60.29 | 60.22 | 60.06 |
| Standard deviation | 41.15 | 28.67 | 29.89 | 20.56 | 19.96 | 20.58 |

**Table 4.16** Results from simulation of P3 with 9 parts, minimum fraction = 90%

| | Parts 1-4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|
| Average lotsize | 1200.12 | 1883.14 | 3695.42 | 3080.75 | 3865.54 | 3985.24 |
| Standard deviation | 69.81 | 210.11 | 365.45 | 610.40 | 725.80 | 734.04 |
| Avg wait for setup | 1.17 | 1.16 | 1.18 | 1.15 | 1.22 | 1.16 |
| Standard deviation | 1.58 | 1.59 | 1.61 | 1.57 | 1.63 | 1.60 |
| Avg leadtime | 13.04 | 11.71 | 10.12 | 9.73 | 8.98 | 8.63 |
| Standard deviation | 11.53 | 10.81 | 9.54 | 9.45 | 9.01 | 8.82 |
| Avg production time | 3.35 | 5.22 | 10.31 | 8.59 | 10.78 | 11.13 |
| Standard deviation | 0.86 | 1.09 | 1.90 | 2.51 | 2.80 | 2.83 |
| Type 1 Service level | 81.76% | 84.54% | 88.49% | 88.58% | 90.72% | 91.01% |
| Type 2 Service level | 99.08% | 98.44% | 98.97% | 97.93% | 98.33% | 98.41% |
| Avg max inventory | 1237.68 | 1988.53 | 3799.15 | 3363.46 | 4130.84 | 4262.39 |
| Avg min inventory | 54.16 | 191.60 | 425.65 | 719.77 | 956.66 | 1011.85 |
| Avg inventory | 645.92 | 1090.07 | 2112.40 | 2041.61 | 2543.75 | 2637.12 |
| Avg time betwn starts | 239.88 | 118.62 | 119.91 | 60.20 | 60.05 | 59.76 |
| Standard deviation | 34.83 | 24.93 | 24.67 | 17.26 | 17.12 | 17.66 |

**Table 4.17** Results from simulation of P2 with 9 parts

| | Parts 1-4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|
| Average lotsize | 1201.06 | 1903.01 | 3712.40 | 3074.75 | 3862.64 | 4014.19 |
| Standard deviation | 83.41 | 207.44 | 373.38 | 575.91 | 670.66 | 711.40 |
| Avg wait for setup | 1.01 | 1.04 | 1.03 | 1.04 | 1.05 | 1.05 |
| Standard deviation | 1.63 | 1.54 | 1.52 | 1.67 | 1.54 | 1.51 |
| Avg leadtime | 11.67 | 10.45 | 9.02 | 8.52 | 7.73 | 7.64 |
| Standard deviation | 11.29 | 10.09 | 8.89 | 8.69 | 8.22 | 8.13 |
| Avg production time | 3.25 | 5.16 | 10.10 | 8.34 | 10.42 | 10.85 |
| Standard deviation | 0.78 | 1.47 | 2.79 | 2.51 | 2.32 | 2.51 |
| Type 1 Service level | 83.88% | 86.57% | 89.74% | 90.05% | 91.64% | 91.82% |
| Type 2 Service level | 99.15% | 98.76% | 99.14% | 98.33% | 98.68% | 98.59% |
| Avg max inventory | 1241.15 | 2020.74 | 3822.86 | 3365.76 | 4139.05 | 4271.89 |
| Avg min inventory | 56.51 | 199.03 | 418.88 | 718.23 | 940.51 | 980.08 |
| Avg inventory | 648.83 | 1109.89 | 2120.87 | 2041.99 | 2539.78 | 2625.99 |
| Avg time betwn starts | 239.48 | 120.39 | 120.42 | 60.24 | 59.83 | 60.01 |
| Standard deviation | 36.06 | 25.63 | 25.10 | 18.31 | 17.20 | 17.20 |

**Table 4.18** Results from simulation of P4 with 9 parts, minimum fraction = 95%

| | Parts 1-4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|
| Average lotsize | 1200 | 1900 | 3700 | 3075 | 3860 | 4000 |
| Standard deviation | 0 | 0 | 0 | 0 | 0 | 0 |
| Avg wait for setup | 1.13 | 1.15 | 1.12 | 1.15 | 1.17 | 1.16 |
| Standard deviation | 1.56 | 1.59 | 1.56 | 1.56 | 1.60 | 1.57 |
| Avg leadtime | 20.68 | 19.33 | 17.22 | 17.54 | 16.55 | 16.33 |
| Standard deviation | 13.10 | 12.71 | 12.12 | 12.42 | 11.97 | 11.88 |
| Avg production time | 3.86 | 6.13 | 11.86 | 9.89 | 12.38 | 12.84 |
| Standard deviation | 1.04 | 1.58 | 1.09 | 1.43 | 1.11 | 1.38 |
| Type 1 Service level | 62.59% | 66.05% | 71.18% | 69.45% | 71.40% | 71.92% |
| Type 2 Service level | 97.79% | 96.22% | 97.00% | 91.54% | 92.45% | 92.49% |
| Avg max inventory | 1195.96 | 1875.04 | 3540.70 | 2815.99 | 3456.17 | 3543.27 |
| Avg min inventory | 15.28 | 73.55 | 205.75 | 253.02 | 388.00 | 403.46 |
| Avg inventory | 605.62 | 974.29 | 1873.22 | 1534.50 | 1922.08 | 1973.36 |
| Avg time betwn starts | 239.11 | 119.88 | 119.90 | 59.96 | 59.90 | 59.69 |
| Standard deviation | 41.41 | 27.87 | 27.28 | 19.22 | 18.43 | 18.77 |

**Table 4.19** Results from simulation of P1 with 9 parts, production rate = 390 parts/hour

| | Parts 1-4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|
| Average lotsize | 1200.90 | 1899.43 | 3706.84 | 3076.58 | 3859.56 | 4012.70 |
| Standard deviation | 379.65 | 757.60 | 1518.68 | 1558.65 | 2040.07 | 1944.29 |
| Avg wait for setup | 3.87 | 6.14 | 11.90 | 9.87 | 12.39 | 12.92 |
| Standard deviation | 1.63 | 2.81 | 5.21 | 5.21 | 6.74 | 6.53 |
| Avg production time | 1.14 | 1.16 | 1.16 | 1.17 | 1.19 | 1.17 |
| Standard deviation | 1.55 | 1.56 | 1.55 | 1.59 | 1.58 | 1.59 |
| Type 1 Service level | 68.92% | 71.77% | 73.15% | 73.41% | 75.14% | 74.04% |
| Type 2 Service level | 88.32% | 86.08% | 86.32% | 83.97% | 84.25% | 85.23% |
| Avg max inventory | 1195.82 | 1873.83 | 3534.07 | 2814.64 | 3455.52 | 3528.86 |
| Avg min inventory | 14.33 | 71.99 | 199.82 | 244.53 | 392.36 | 385.00 |
| Avg inventory | 605.08 | 972.91 | 1866.94 | 1529.58 | 1923.94 | 1956.93 |
| Avg time betwn starts | 239.99 | 120.00 | 120.00 | 60.00 | 60.00 | 60.00 |
| Standard deviation | 67.14 | 41.91 | 43.04 | 25.06 | 26.71 | 23.52 |

**Table 4.20** Results from simulation of P5 with 9 parts, production rate = 390 parts/hour

| | Parts 1-4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|
| Average lotsize | 1200.92 | 1906.89 | 3704.37 | 3083.24 | 3851.91 | 3999.04 |
| Standard deviation | 170.59 | 343.76 | 636.86 | 710.68 | 890.82 | 908.86 |
| Cycles at capacity | 20.7% | 27.9% | 23.6% | 36.5% | 35.1% | 35.6% |
| Avg production time | 3.82 | 6.11 | 11.85 | 9.86 | 12.32 | 12.81 |
| Standard deviation | 0.65 | 1.15 | 2.15 | 2.26 | 2.86 | 2.92 |
| Type 1 Service level | 71.27% | 72.63% | 75.21% | 75.00% | 75.99% | 76.85% |
| Type 2 Service level | 94.23% | 92.36% | 93.86% | 89.81% | 90.79% | 90.80% |
| Avg max inventory | 1196.47 | 1879.27 | 3542.48 | 2821.76 | 3454.35 | 3539.40 |
| Avg min inventory | 14.76 | 69.32 | 203.96 | 247.26 | 389.82 | 407.32 |
| Avg inventory | 605.62 | 974.30 | 1873.22 | 1534.51 | 1922.08 | 1973.36 |
| Avg time betwn starts | 240 | 120 | 120 | 60 | 60 | 60 |
| Standard deviation | 0 | 0 | 0 | 6.16 | 6.16 | 6.16 |

**Table 4.21** Results from simulation of P6 with 9 parts, production rate = 390 parts/hour

| | Parts 1-4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|
| Average lotsize | 1292.79 | 2102.42 | 3605.03 | 3340.51 | 3597.32 | 3596.11 |
| Standard deviation | 133.57 | 309.68 | 286.07 | 473.54 | 270.20 | 268.34 |
| Cycles at capacity | 0.1% | 0.4% | 97.5% | 59.5% | 96.7% | 97.6% |
| Avg production time | 4.16 | 6.73 | 11.49 | 10.66 | 11.46 | 11.47 |
| Standard deviation | 0.81 | 1.27 | 0.11 | 1.41 | 0.27 | 0.25 |
| Type 1 Service level | 70.01% | 70.21% | 68.99% | 69.90% | 68.69% | 68.51% |
| Type 2 Service level | 95.81% | 94.56% | 94.32% | 91.86% | 90.83% | 90.47% |
| Avg max inventory | 1254.62 | 2002.09 | 3265.30 | 2912.09 | 2978.03 | 2967.11 |
| Avg min inventory | -17.40 | 6.20 | 13.21 | 119.91 | 118.56 | 137.42 |
| Avg inventory | 618.61 | 1004.14 | 1639.25 | 1516.00 | 1548.29 | 1552.27 |
| Avg time betwn starts | 258.67 | 132.99 | 116.87 | 65.13 | 55.95 | 53.87 |
| Standard deviation | 40.23 | 28.38 | 29.38 | 19.71 | 19.34 | 18.92 |

**Table 4.22** Results from simulation of P7 with 9 parts, production rate = 390 parts/hour

| | Parts 1-4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|
| Average lotsize | 1200 | 1900 | 3700 | 3075 | 3860 | 4000 |
| Standard deviation | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | |
| Avg wait for setup | 0.56 | 0.55 | 0.56 | 0.56 | 0.57 | 0.56 |
| Standard deviation | 0.93 | 0.91 | 0.91 | 0.92 | 0.95 | 0.92 |
| | | | | | | |
| Avg leadtime | 9.55 | 9.00 | 7.57 | 7.26 | 6.54 | 6.43 |
| Standard deviation | 8.24 | 8.00 | 7.17 | 7.20 | 6.80 | 6.65 |
| | | | | | | |
| Avg production time | 3.34 | 5.28 | 10.30 | 8.60 | 10.73 | 11.14 |
| Standard deviation | 0.75 | 1.06 | 1.63 | 1.95 | 1.39 | 1.57 |
| | | | | | | |
| Type 1 Service level | 89.47% | 91.05% | 93.38% | 93.78% | 94.66% | 94.67% |
| Type 2 Service level | 99.63% | 99.35% | 99.56% | 99.02% | 99.20% | 99.30% |
| | | | | | | |
| Avg max inventory | 1254.51 | 2053.29 | 3882.61 | 3477.04 | 4272.68 | 4415.58 |
| Avg min inventory | 71.27 | 236.25 | 500.67 | 841.77 | 1101.43 | 1158.66 |
| Avg inventory | 591.62 | 908.52 | 1690.97 | 1317.64 | 1585.62 | 1628.46 |
| | | | | | | |
| Avg time betwn starts | 239.56 | 120.31 | 119.69 | 60.12 | 60.02 | 59.94 |
| Standard deviation | 36.00 | 26.45 | 26.18 | 19.91 | 18.91 | 18.64 |

**Table 4.23** Results from simulation of P1 with 9 parts, waiting for setup crews reduced

| | Parts 1-4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|
| Average lotsize | 1199.8 | 1897.28 | 3709.4 | 3074.96 | 3861.07 | 4006.39 |
| Standard deviation | 66.1018 | 111.24 | 283.419 | 207.972 | 304.189 | 333.737 |
| | | | | | | |
| Avg wait for setup | 0.29 | 0.29 | 0.34 | 0.33 | 0.37 | 0.38 |
| Standard deviation | 0.76 | 0.75 | 0.80 | 0.80 | 0.83 | 0.83 |
| | | | | | | |
| Avg leadtime | 8.46 | 7.74 | 6.84 | 6.51 | 5.86 | 5.67 |
| Standard deviation | 7.88 | 7.55 | 6.70 | 6.66 | 6.39 | 6.22 |
| | | | | | | |
| Avg production time | 3.26 | 5.16 | 10.21 | 8.45 | 10.67 | 11.11 |
| Standard deviation | 0.80 | 1.20 | 1.43 | 2.03 | 1.73 | 1.93 |
| | | | | | | |
| Type 1 Service level | 91.96% | 93.46% | 95.08% | 95.08% | 96.16% | 96.15% |
| Type 2 Service level | 99.72% | 99.51% | 99.67% | 99.31% | 99.34% | 99.32% |
| | | | | | | |
| Avg max inventory | 1260.42 | 2072.51 | 3919.99 | 3526.90 | 4332.31 | 4472.10 |
| Avg min inventory | 77.20 | 257.38 | 525.99 | 886.01 | 1155.08 | 1199.76 |
| Avg inventory | 591.61 | 907.56 | 1697.00 | 1320.45 | 1588.62 | 1636.17 |
| | | | | | | |
| Avg time betwn starts | 240.58 | 120.00 | 119.94 | 59.85 | 60.13 | 60.11 |
| Standard deviation | 38.36 | 26.95 | 26.69 | 19.27 | 19.34 | 19.32 |

**Table 4.24** Results from simulation of P3 with 9 parts, waiting for setup crews reduced, minimum fraction = 95%

| | Parts 1-4 | Part 5 | Part 6 | Part 7 | Part 8 | Part 9 |
|---|---|---|---|---|---|---|
| Average lotsize | 1201.04 | 1899.93 | 3700.84 | 3074.54 | 3861.78 | 4000.17 |
| Standard deviation | 32.4193 | 50.553 | 108.545 | 86.369 | 115.174 | 118.695 |
| Avg wait for setup | 0.50 | 0.49 | 0.53 | 0.50 | 0.52 | 0.52 |
| Standard deviation | 0.90 | 0.89 | 0.93 | 0.89 | 1.04 | 0.93 |
| Avg leadtime | 9.25 | 8.52 | 7.33 | 6.92 | 6.27 | 6.00 |
| Standard deviation | 8.43 | 8.21 | 7.28 | 7.12 | 6.81 | 6.43 |
| Avg production time | 3.30 | 5.22 | 10.15 | 8.48 | 10.56 | 10.97 |
| Standard deviation | 0.84 | 1.06 | 1.99 | 2.17 | 1.22 | 2.13 |
| Type 1 Service level | 90.30% | 91.53% | 93.82% | 94.42% | 95.40% | 95.70% |
| Type 2 Service level | 99.65% | 99.34% | 99.57% | 99.22% | 99.29% | 99.35% |
| Avg max inventory | 1258.30 | 2057.63 | 3899.35 | 3504.83 | 4310.53 | 4463.49 |
| Avg min inventory | 73.47 | 241.09 | 510.24 | 866.31 | 1128.66 | 1185.13 |
| Avg inventory | 665.88 | 1149.36 | 2204.80 | 2185.57 | 2719.59 | 2824.31 |
| Avg time betwn starts | 240.27 | 119.90 | 120.17 | 59.98 | 60.13 | 60.11 |
| Standard deviation | 36.02 | 26.84 | 26.28 | 18.91 | 18.73 | 18.76 |

**Table 4.25** Results from simulation of P3 with 9 parts, waiting for setup crews reduced, minimum fraction = 97.5%

## References for Chapter 4


Abramowitz, Milton and Irene A. Stegun. Handbook of Mathematical Functions, Applied Mathematics Series, vol. 55. Washington: National Bureau of Standards, 1964 (reprinted by Dover Publications, New York, 1965).

Apple Computer, Inc. Apple Numerics Manual, Second Edition. Reading, MA: Addison-Wesley, 1988.

Ashcroft, H. "The Productivity of Several Machines Under the Care of One Operator". Journal of the Royal Statistical Society, Series B, 12, pp. 145-151, 1950.

Baker, Kenneth R. "Requirements Planning", Chapter 11 in Graves, S. C., A. H. G. Rinnooy Kan and P. H. Zipkin, eds., Logistics of Production and Inventory, Amsterdam: Elsevier, 1993.

Benson, F and D. R. Cox. "The Productivity of Machines Requiring Attention at Random Intervals". Journal of the Royal Statistical Society, Series B, 13, pp. 65-82, 1951.

Bielecki, T. and P. R. Kumar. "Optimality of Zero-Inventory Policies for Unreliable Manufacturing Systems". Operations Research, 36(4), pp. 532-541, 1988.

Bratley, Paul, Bennett L. Fox, and Linus E. Schrage. A Guide to Simulation, Second Edition. New York: Springer-Verlag, 1987.

Bunday, B. D. and R. E. Scraton. "The G/M/r Machine Interference Model". European Journal of Operational Research, 4, pp. 399-402, 1980.

Buzacott, John A. and J. George Shanthikumar. "A General Approach for Coordinating Production in Multiple-Cell Manufacturing Systems". Production and Operations Management, 1(1), pp. 34-52, 1992.

Buzacott, John A. and J. George Shanthikumar. Stochastic Models of Manufacturing Systems. Englewood Cliffs, NJ: Prentice-Hall, 1993.

Cox, D. R. and Walter L. Smith. Queues. London: Methuen & Co., Ltd., 1961.

Dongarra, J. J. and E. Grosse. "Distribution of Mathematical Software via Electronic Mail". Communications of the ACM, 30(5), pp. 403-407, May 1987.

Federgruen, Awi and Ziv Katalan. "The Stochastic Economic Lot Scheduling Problem: Cyclical Base-Stock Policies with Idle Times". Working Paper, Philadephia, PA: The Wharton School, University of Pennsylvania, 1994 (a). Forthcoming in Management Science.

Federgruen, Awi and Ziv Katalan. "Customer Waiting Time Distributions Under Base-Stock Policies In Single Facility Multi-Item Production Systems". Working Paper, Philadephia, PA: The Wharton School, University of Pennsylvania, 1994 (b).

Gershwin, Stanley B. Manufacturing Systems Engineering. Englewood Cliffs, NJ: Prentice-Hall, 1994.

Graves, Stephen C. "The Multi-Product Production Cycling Problem". AIIE Transactions, 12, pp. 233-240, 1980.

Gross, Donald and Carl M. Harris. Fundamentals of Queueing Theory. New York: John Wiley & Sons, Inc., 1985.

Hadley, G. and T. M. Whitin. Analysis of Inventory Systems. Englewood Cliffs, NJ: Prentice-Hall, 1963.

Heyman, Daniel P. and Matthew J. Sobel. Stochastic Models in Operations Research, Volume I. New York: McGraw-Hill, 1982.

Hopp, Wallace J. and Mark L. Spearman. Factory Physics: Foundations of Manufacturing Management. Chicago, IL: Irwin, 1996.

Jaiswal, N. K. Priority Queues. New York: Academic Press, 1968.

Karmarkar, Uday S. "Lot Sizes, Lead Times and In-Process Inventories". Management Science, 33(3), pp. 409-418, 1987.

Karmarkar, Uday S. "Manufacturing Lead Times, Order Release and Capacity Loading", Chapter 6 in Graves, S. C., A. H. G. Rinnooy Kan and P. H. Zipkin, eds., Logistics of Production and Inventory, Amsterdam: North-Holland, 1993.

Kimemia, Joseph and Stanley B. Gershwin. "An Algorithm for the Computer Control of a Flexible Manufacturing System". IIE Transactions, 15(4), pp. 353-362, 1983.

Kleinrock, Leonard. Queueing Systems, Volume 1: Theory. New York: John Wiley & Sons, Inc., 1975.

Kletter, David B. "Determining Production Lot Sizes and Safety Stocks for an Automobile Stamping Plant". S.M. Thesis, MIT, 1994.

Knuth, Donald E. The Art of Computer Programming, Second Edition: Volume 2, Seminumerical Algorithms. Reading, MA: Addison-Wesley, 1981.

Krajewski, L. J., B. E. King, L. P. Ritzman and D. S. Wong. "Kanban, MRP, and Shaping the Manufacturing Environment". Management Science, 33(1), pp. 39-57.

Law, Averill M. and W. David Kelton. Simulation Modeling and Analysis, Second Edition. New York: McGraw-Hill, 1991.

Lee, Hau L. and Stephen Nahmias. "Single-Product, Single-Location Models", Chapter 1 in Graves, S. C., A. H. G. Rinnooy Kan and P. H. Zipkin, eds., Logistics of Production and Inventory, Amsterdam: Elsevier, 1993.

Lin, Li and Jeffery K. Cochran. "Metamodels of Production Line Transient Behaviour for Sudden Machine Breakdowns". International Journal of Production Research, 28(10), pp. 1791-1806, 1990.

Maimon, Oded Z. and Stanley B. Gershwin. "Dynamic Scheduling and Routing for Flexible Manufacturing Systems that Have Unreliable Machines". Operations Research, 36(2), pp. 279-292, 1988.

Markowitz, David M., Martin I. Reiman and Lawrence M. Wein. "The Stochastic Economic Lot Scheduling Problem: Heavy Traffic Analysis of Dynamic Cyclic Policies". Working Paper 3863-95, Cambridge, MA: Sloan School of Management, MIT, 1995.

Muckstadt, John A. and Robin O. Roundy. "Analysis of Multistage Production Systems", Chapter 2 in Graves, S. C., A. H. G. Rinnooy Kan and P. H. Zipkin, eds., Logistics of Production and Inventory, Amsterdam: Elsevier, 1993.

Nahmias, Stephen. Production and Operations Analysis. Homewood, Illinois: Irwin, 1989.

Orlicky, Joseph. Material Requirements Planning. New York: McGraw-Hill, 1975.

Press, William H., Brian P. Flannery, Saul A. Teukolsky and William T. Vetterling. Numerical Recipes in Pascal. Cambridge: Cambridge University Press, 1989.

Pritsker, A. Alan B. Introduction to Simulation and SLAM II, Fourth Edition. New York: John Wiley & Sons, Inc., 1995.

Ross, Sheldon M. Stochastic Processes. New York: John Wiley & Sons, Inc., 1983.

Saaty, Thomas L. Elements of Queueing Theory. New York: McGraw-Hill, 1961.

Spearman, Mark L., David L. Woodruff and Wallace J. Hopp. "CONWIP: A Pull Alternative to Kanban". International Journal of Production Research, 28(5), pp. 879-894, 1990.

Spearman, Mark L. and M. A. Zazanis. "Push and Pull Production Systems: Issues and Comparisons". Operations Research, 40(3), pp. 521-532, 1992.

Stecke, K. E. and J. E. Aronson. "Review of Operator/Machine Interference Models". International Journal of Production Research, 23(1), pp. 129-151, 1985.

Suri, Rajan, Jerry L. Sanders and Manjunath Kamath. "Performance Evaluation of Production Networks", Chapter 5 in Graves, S. C., A. H. G. Rinnooy Kan and P. H. Zipkin, eds., Logistics of Production and Inventory, Amsterdam: North-Holland, 1993.

Takács, Lajos. Introduction to the Theory of Queues. New York: Oxford University Press, 1962.

Wein, Lawrence M. "Dynamic Scheduling of a Multiclass Make-to-Stock Queue". Operations Research, 40(4), pp. 724-735, 1992.

Wolff, Ronald W. "Poisson Arrivals See Time Averages". Operations Research, 30(2), pp. 223-231, 1982.

Zipkin, Paul H. "Models for Design and Control of Stochastic, Multi-Item Batch Production Systems". Operations Research, 34(1), pp. 91-104, 1986.

Zipkin, Paul H. "Performance Analysis of a Multi-Item Production-Inventory System Under Alternative Policies". Management Science, 41(4), pp. 690-703, 1995.

# 5. Conclusions and Future Research

In this brief chapter, we provide some thoughts on what has been and what remains to be accomplished in the areas we have studied. We organize our discussion by chapter.

**Chapter 2: A model of an unreliable machine**

We have made great progress for the case of i.i.d. exponential failures and i.i.d. exponential repair times, in deriving expressions for the amount of time required to produce a fixed number of parts, and for the number of parts produced over a fixed time interval. For both of these random variables, we have derived the moments, probability density functions, cumulative distribution functions and Laplace transforms (sometimes in terms of modified Bessel functions).

We have made very little progress when the assumptions on repair times and failure times do not hold. We note that in some cases, the repair time distribution may not be independent of the time since the last repair, and the time until the next failure may not be independent of the repair time. Further, although we believe (based on unpublished data) that the exponential distribution is reasonable for GM metal stamping lines, there will be situations in which a different distribution must be used. Barlow and Proschan (1965) and Proschan and Pyke (1967) describe a statistical method to test if data was generated by an exponential distribution.

Our models also assume that the parameters required are known exactly. It is easy to imagine situations in which the MTBF, MTTR, number of parts to be produced or the length of time available is not known with certainty. For example, the MTBF and MTTR might be uncertain if the machine, die, automation, jigs or fixtures, the

part itself, or the operator are new or different. The number of parts that must be produced might be uncertain if some random quantity of parts that are produced are defective. The length of time available for production might be uncertain if, for example, the number of workers available is uncertain, or if the amount of machine time that must be reserved for other activities – such as preventative maintenance, changeover, or make-to-order parts – is uncertain.

Clearly, we have only scratched the surface. Many interesting and useful extensions to our results could be explored.

**Chapter 3: Dynamic overtime decision model**

We believe we have presented a useful operational model that could be used as part of a real-time decision support system to aid in the decision of when and how much overtime to run on an unreliable machine. We briefly comment on a few of the extensions that would make this model even more useful.

As mentioned in Chapter 3, the incorporation of stochastic demand would increase the number of production environments in which this model could be successfully applied. Our assumption that demand is known over some short horizon will not be true in some settings. We have made some progress toward this goal by studying special cases.

The addition of stochastic setup times to the model would better reflect reality in a metal stamping plant (i.e., see Chapter 4). Furthermore, the cost of overtime would ideally be a function of $\tau$, since a setup crew may or may not be needed during the overtime shift, depending on the value of $\tau$.

We might wish to model stockout costs as a function of the time that the order is outstanding, instead of incurring a one time penalty. This might be desirable simply because it is a better reflection of the cost structure incurred in a particular situation. However, we conjecture that the difficulties that we have with the optimal policy (the existence of a lower envelope, and possibly more than two critical values) would also be solved by such a change to the model. This would require new probability models in which we are able to compute the joint distribution of cumulative uptime and cumulative backlog.

Lastly, we note that an ideal model would be one that incorporates dynamic rescheduling. We expect this to be very difficult to achieve.

**Chapter 4: Comparison of operating policies for a single unreliable machine**

We will not repeat the conclusions that we presented in the final section of Chapter 4. We instead offer a few remarks about future research opportunities.

Our model assumed that demand was uncorrelated over time and across parts. Since this assumption does not hold true in many environments, it would be worthwhile to explore how, if at all, such correlations change our conclusions.

We note that P7 is a myopic policy, since it bases its production decision on the likelihood of stockout over the next production interval. A better policy might look ahead several intervals each time it must make a decision. An improved policy such as this might also be able to properly respond to cycles of varying lengths, which might reduce the undesirable behavior experienced at high utilizations.

Of course, we identified seven additional policies in Section 4.1 that may be worthy of exploration. Of particular interest is P8, which retains the desirable properties of P1 yet may perform better when demand variability is high. Policies of the type suggested by Graves (1980) that authorize production based on both aggregate and individual item inventory levels may also prove to be highly successful.

## References for Chapter 5

Barlow, Richard E. and Frank Proschan. Mathematical Theory of Reliability. New York: John Wiley & Sons, Inc., 1965.

Graves, Stephen C. "The Multi-Product Production Cycling Problem". AIIE Transactions, 12, pp. 233-240, 1980.

Proschan, Frank and Ronald Pyke. "Tests for Monotone Failure Rate". Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume III, pp. 293-312. Berkeley, CA: University of California Press, 1967.