SPRINGER OPTIMIZATION
AND ITS APPLICATIONS 31

Chrissoleon T. Papadopoulos
Michael E. J. O'Kelly · Michael J. Vidalis
Diomidis Spinellis

Analysis and Design of Discrete Part Production Lines

Springer

ANALYSIS AND DESIGN OF DISCRETE PART PRODUCTION LINES

Springer Optimization and Its Applications

VOLUME 31

Aims and Scope
Optimization has been expanding in all directions at an astonishing rate during the last few decades. New algorithmic and theoretical techniques have been developed, the diffusion into other disciplines has proceeded at a rapid pace, and our knowledge of all aspects of the field has grown even more profound. At the same time, one of the most striking trends in optimization is the constantly increasing emphasis on the interdisciplinary nature of the field. Optimization has been a basic tool in all areas of applied mathematics, engineering, medicine, economics and other sciences.

The *Springer Optimization and Its Applications* series publishes undergraduate and graduate textbooks, monographs and state-of-the-art expository works that focus on algorithms for solving optimization problems and also study applications involving such problems. Some of the topics covered include nonlinear optimization (convex and nonconvex), network flow problems, stochastic optimization, optimal control, discrete optimization, multiobjective programming, description of software packages, approximation techniques and heuristic approaches.

For other titles in this series, go to www.springer.com/series/7393

Analysis and Design of Discrete Part Production Lines

By

CHRISSOLEON T. PAPADOPOULOS
Aristotle University of Thessaloniki, Greece

MICHAEL E. J. O'KELLY
Waterford Institute of Technology, Ireland

MICHAEL J. VIDALIS
University of the Aegean, Greece

DIOMIDIS SPINELLIS
Athens University of Economics and Business, Greece

Springer

Chrissoleon T. Papadopoulos
Aristotle University of Thessaloniki
Department of Economics
Thessaloniki, Greece
hpap@econ.auth.gr

Michael E. J. O'Kelly
Waterford Institute of Technology
Waterford, Ireland
jokelly@eircom.net

Michael J. Vidalis
University of the Aegean
Department of Business Administration
Chios, Greece
mvid@ba.aegean.gr

Diomidis Spinellis
Athens University of Economics and Business
Department of Management Science
and Technology
Athens, Greece
dds@aueb.gr

ISSN: 1931-6828
ISBN: 978-0-387-89493-5 e-ISBN: 978-0-387-89494-2
DOI 10.1007/978-0-387-89494-2
Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2009926035

Mathematics Subject Classification (2000): 90-02, 90B30, 90B25, 90B50, 91B74, 90C90

Cover illustration: Photo taken by Elias Tyligadas

Printed on acid-free paper

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

Dedicated to those who are so much more important to us than mere production lines, but who could have reasons to inquire 'why then spend so much time with production lines?'

Maridora, Katerina, Elisavet — Jane, John, Eamon, Ita, Paul, David, Jenny — Katerina, Ioannis, Athanasia, Daphni — Eliza, Dionysis, Eleana

Preface

Initially, during discussions among the four colleagues about this writing project, we used "on the optimal design of production lines" as the working title of the book. However, it must be understood that all models involve assumptions and unless these assumptions are valid, the results could not be described as optimal. So basically, what this text is offering is a set of best solutions to the models as described in the various chapters. The models and the algorithms presented are generally accepted by internationally respected scholars to give very good solutions following extensive simulation and comparison with actual systems. We, therefore, see the process of the optimal design of production lines as a complementary activity between the scholars and the practitioners. The scholars provide models and associated algorithms and the practitioners, in their turn, ensure the appropriateness of the assumptions of the models used together with the validity of the data used, and hence, in effect there is a joint responsibility to achieve the optimal or near optimal design of production lines. It is our experience that practitioners in industry and consultancy companies often have considerable difficulties with the academic and research papers which appear in international journals due to the complexity of the mathematical analysis involved and the lack of readily available efficient algorithms for the solution of the models presented. The literature consists of a large number of excellent papers and it is extremely difficult for the practitioner to have an opportunity to examine the appropriateness of each paper to the design problem on hand. We thought that this project could assist the practitioner in this regard by providing a set of models which have been found useful in specific situations. Of course, it is not claimed that these models cover every conceivable situation, but the authors believe that they provide an extremely useful starting point for the understanding of production lines. Furthermore, we thought that it would be very useful to have in one place a collection of relevant analysis and design material of production lines. For this reason, we decided to put the algorithms on a web site: http://purl.oclc.org/NET/prodline. Here, we would like to sincerely acknowledge the generosity of many colleagues across the world who gave us access to the relevant algorithms. Without such generosity and cooperation our project would have been a total failure.

Production lines in the context of this work are a subset of general manufacturing systems. There are various types of manufacturing systems such as job shops, flexible manufacturing systems (FMS), flexible assembly systems (FAS), production or flow lines and automatic transfer lines. The usual features of a production or flow line are dedicated work-stations, manual or automatic, usually producing a single product with a fixed routing and an asynchronous movement of material between the work-stations and a high mean production rate (throughput). Production lines are complex systems. Full understanding of such systems requires skilled analysis in order to facilitate the development of a competent design. Some important design problems associated with production lines consist of decisions in relation to three main issues, viz., work-load allocation, buffer allocation and server allocation. The objective of this book is to provide the reader with a set of models and solutions to these problems (work-load allocation problem (WAP), buffer allocation problem (BAP) and server allocation problem (SAP)) which are accepted by experienced researhers and practitioners to be of value in the design of these systems. To assist in the solution of these design problems, it is necessary to make use of both evaluative and generative (optimization) algorithms.

During the course of a project like this, a number of changes of perspective and vision, as time progresses, are inevitable. Accordingly, we decided to change the working title of the book to "Discrete Part Production Lines." It is the authors' view that the models presented may be used in either of two modes, viz., analysis and design. For actual existing lines, the models may be used to predict performance under existing conditions or if certain changes are made, for example, to the number of buffer slots before a particular station. If a new design is contemplated, then, of course, a range of models may be used having in mind the objectives of the design including cost considerations. We hope that the Analysis and Design Decision Network, given in the book's web site, will assist the readers in choosing appropriate models for their investigations. Researchers and practitioners alike have sometimes questioned the usefulness as well as the benefits derived from very detailed and somewhat complex analysis of production lines. It is, of course, not always feasible to adopt in practice what may be the theoretical optimal or near optimal solution to a design problem in production lines, developed from system modeling. However, if one knows the optimal or near optimal solution, the theoretical prime cost of adopting a more 'practical' solution would be of interest. Clearly, the software associated with this text would be of assistance in discussions of these matters.

In Chapter 1, "*Manufacturing Systems: Types and Modeling,*" an overview of the evolution and classification of manufacturing systems is given as well as an introduction to models and modeling.

Chapter 2, "*Evaluative Models of Discrete Part Production Lines,*" describes four predictive models of performance evaluation of production or flow lines: the Markovian model, the expansion method, the aggregation method and the decomposition approach applied both to single-machine station and parallel-machine station production lines. A short section on simulation modeling is given at the end of this chapter.

Chapter 3, "*The Design of Production Lines,*" introduces the reader to the design problems of production lines and the concept of improvability.

Chapter 4, "*Work-Load and Server Allocation Problems,*" describes two separate problems, viz., the work-load allocation problem and the server allocation problem.

Chapter 5, "*The Buffer Allocation Problem,*" describes this important problem within the context of production lines.

Chapter 6, "*Double and Triple Optimization,*" considers the combinations of the three pure work-load allocation, server allocation and buffer allocation problems, taken two at a time or all three together.

Chapter 7, "*Cost Considerations,*" examines cost considerations in the design of production lines using profit maximization and cost minimization objective functions.

In Appendix A, a review of some mathematical fundamentals is given, mainly from linear algebra, probability theory, discrete Markov processes (Markov chains) and queueing theory.

Appendix B contains details concerning the code available on the book's web site. For each algorithm we provide its author, its coder, a short description, the corresponding output, and key bibliographic references.

Appendix C gives the glossary.

The authors are conscious of the debt of gratitude they owe to a very large number of researchers and practitioners, much too numerous to list, in the area of the design of production lines and manufacturing systems in general. We believe that we must make a special mention of those colleagues who participated either as presenters or attendees at the five Hellenic International Conferences on Analysis, Design and Optimization of Manufacturing Systems which were held in Greece (four at the Islands of the Aegean Archipelagos at Samos, Tinos, Tinos and Samos, respectively, and one at Zakynthos Island of the Ionian Sea) and at the 30th Computers & Industrial Engineering International Conference which was held on Tinos Island and who assisted us so much in crystallizing our understanding of the research work in this area. As we are reluctant to list any specific colleagues for special acknowledgment, we give in Appendix D a list of all colleagues who participated as presenters or attendees at the five Hellenic International Conferences on Analysis, Design and Optimization of Manufacturing Systems.

Appendix E presents an Arena simulation model of a reliable production line.

In conclusion, the authors hope that the background theory, details of the relevant algorithms, tabulations of actual computer runs and the provision of the algorithms at the website associated with this text will together form a reservoir of knowledge to assist the designers of practical production lines. In particular, the authors hope that the guides to the use of these algorithms given throughout the text and in Appendix B will assist the busy designers and practitioners in choosing appropriate computational tools for their analyses. The individual contributions of the authors are given in the book, but, of course, the composite contributions of many other researchers which are included and acknowledged in the text far outweigh what any one of the authors could hope to contribute.

Although Dr. Alexandros Diamantidis's name appears in both Appendix B and Appendix D, all the authors wish to make a special acknowledgment of his contribution to our work particularly in relation to the development of the effective evaluative decomposition algorithm for solving multi-station multi-server production lines and for running various problems sets at our request.

Needless to say, as any academic will attest, we are individually very much in debt to our students who over the years have assisted us in advancing our understanding of the fascinating subject of production lines.

We wish to acknowledge very sincerely the patience of the publisher with the delay in producing this text caused inter alia by one of the authors being indisposed for a relatively long period of time.

Finally, the authors would be very pleased to hear from researchers or practitioners who wish to have an algorithm/procedure, developed by them, to be considered for inclusion at the website. No claim is made, at this point in time, that the algorithms presented can handle all possible realistic design problems for either short or long production lines and it is in that spirit that the authors invite other researchers to make available their algorithms so that the issues related to the design of production lines are finally closed. Hopefully, in time, a very comprehensive set of algorithms/procedures for the analysis/design of production lines would become available for all to use. This could well be the first step to having on a website a set of algorithms/procedures which have been found to be of value in the design and analysis of general manufacturing systems.

Thessaloniki, 2009 *Chrissoleon T. Papadopoulos*
Waterford, 2009 *Michael E.J. O'Kelly*
Chios, 2009 *Michael J. Vidalis*
Athens, 2009 *Diomidis Spinellis*

Contents

List of Figures

List of Tables

1

Manufacturing Systems: Types and Modeling

Designers in the past were well aware of the need for effective production systems but were hampered in the development of such systems by a lack of appropriate manufacturing technology and system design techniques. In the 1950s, the emphasis of production management was essentially on throughput and standardization. The economic philosophy was based on the economics of scale with a significant orientation toward production to stock. Today the situation is very much changed due to the introduction of new manufacturing technologies and management philosophies. The focus now is more on the economics of scope, the customization of products and the preeminence of the market. In the meantime, there has been a considerable advance in the range of tools available to the designer of production systems.

In this chapter we give a brief overview of the significant technological changes which have occurred since the 1950s. The importance of information technology in manufacturing systems and the need for the designer to have performance measures other than throughput in mind during the design process is treated. A presentation of some areas in mathematical analysis, which are important for our work, is contained in Appendix A. In Section 1.1, the evolution and classification of manufacturing systems is covered. Section 1.2 treats mathematical models and the modeling process. Section 1.3 attempts a general classification of manufacturing systems with a view to showing the inherent complexities. Section 1.4 discusses models in the context of manufacturing systems, whereas Section 1.5 treats methods of analysis of such models. Finally, Section 1.6 presents measures of performance in manufacturing systems.

1.1 Manufacturing Systems: Evolution and Classification

Manufacturing is a transformation process as shown in Figure 1.1.

In this model, the inputs (capital, raw materials, energy, educated and trained personnel, equipment and facilities, tools and software, and customer demand) are transformed to finished products which are demanded by the market. Inevitably there is some waste and scrap produced. The management of such a transforming process

C. T. Papadopoulos et al., *Analysis and Design of Discrete Part Production Lines*, Springer Optimization and Its Applications,
DOI: 10.1007/978-0-387-89494-2_1,

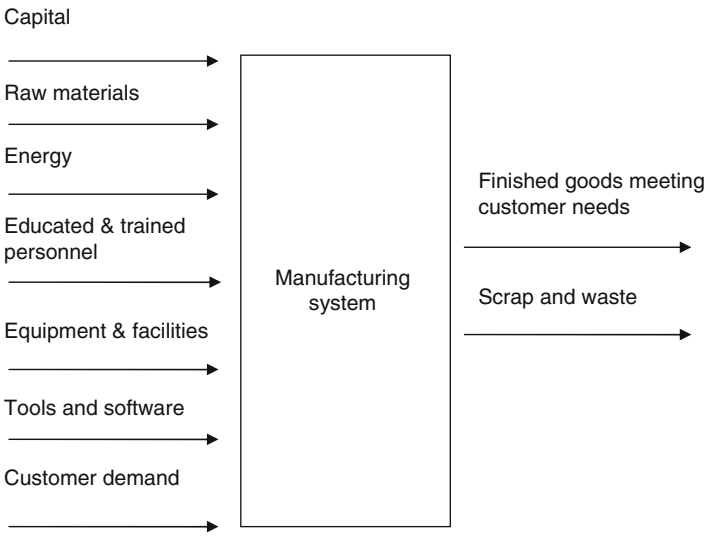


Fig. 1.1. Manufacturing transformation process

is quite complex, as it must have regard for the business imperatives as well as the technical possibilities of the manufacturing process. In the past, the transformation process tended to take place within the four walls of a factory but this is no longer the case with the formation of virtual enterprises.

One of the major technological breakthroughs in addition to the abundant supply of energy which led to modern manufacturing was the development of the concept of interchangeable production, credited to Eli Whitney in the early 1800s. Up to that time, craft persons tended to make complete and often individualized products and so interchangeability of parts was not of major importance. The philosophical concept of division of labor, developed by Adam Smith, with its associated cost advantages further added to the development of manufacturing as a form of production. The interested reader is referred to the very rich literature on operations management to appreciate the work of such pioneers as Taylor, Gantt, Babbage, and Hawthorn, among others.

At the strategic level, a company must decide which markets it wishes to compete in and what will be its competitive advantages in these markets. Clearly, appropriate technology will confer competitive advantage on a company provided the other essential ingredients for success are also in place. In addition to training and education, such additional ingredients include the layout of the plant and the basic choice of the manufacturing process.

In all types of manufacturing systems there are a number of basic functions or activities or operations which must be performed during the transformation process. These activities include:

- Processing operations
- Assembly operations

- Material handling, transportation and storage
- Product quality assurance, inspection and test
- Process control

It is unfortunate that the word "process" has so many different meanings in the context of manufacturing. Processing operations transform the product from one state to a more advanced state of completion. Such processing operations (e.g., metal removal, distillation) may be classified in different ways but these classifications are unimportant in the context of this work.

Assembly and joining operations (or blending) involve the combination of two or more separate components. In some systems the operations start at assembly, because no other processing is involved. Thus, it is possible to describe a particular manufacturing system as a flexible assembly system (FAS).

Material handling may be manual, semi-automatic or automatic.

Product quality assurance activities are major activities in modern manufacturing. The functions involved may be automated or carried out manually. In some systems, results from quality assurance are fed back to the production machines.

Process control involves the achievement of performance objectives through the manipulation of inputs to the process. There exists a significant body of knowledge, based on statistical methods, to achieve process control.

Speed, reliability, flexibility, cost, rapid product innovation and quality are all related to process choice. The interrelationship between process choice , plant layout and investment in technology is clearly shown in Figure 1.2, based on material given in Brown (1996).

Classically, the layout decision is often described in terms of maximizing the use of equipment and personnel and is essentially considered to be tactical in nature. For example, well-known layout techniques exist which minimize the distance traveled by operatives. However, the layout problem should more properly be conceived as part of a strategic decision which supports the process choice in serving the chosen markets.

It is generally agreed that there are five basic process choices as follows:

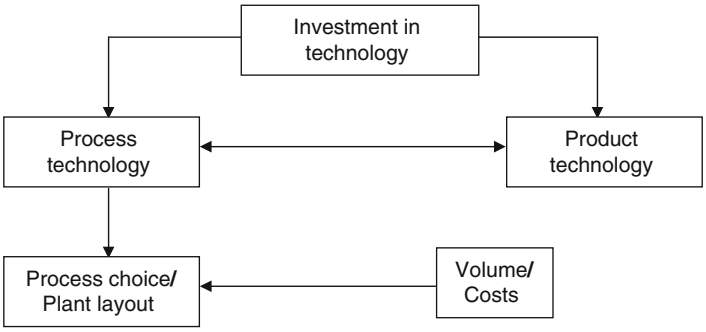


Fig. 1.2. The interrelationship between process choice, plant layout and technology investment

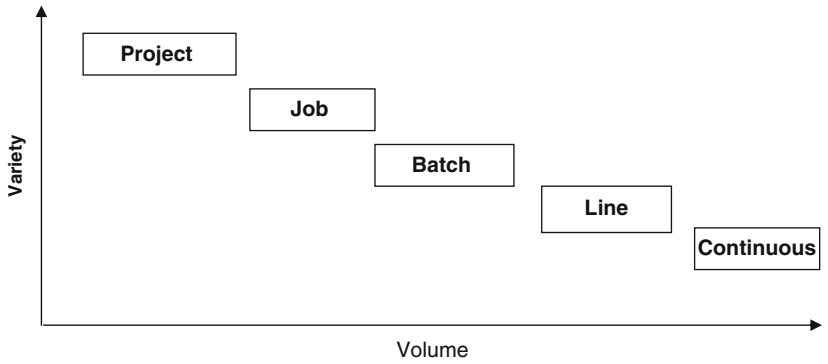


Fig. 1.3. Process choice

1. Project: One large and complex unit.
2. Job Shop: Different products in small lot sizes.
3. Batch: Many standard or similar products to customers' specifications but relatively small volumes.
4. Production/Transfer Line: High-volume repetitive production of discrete units often associated with a moving assembly line.
5. Continuous Process: Flow-process required by the production technology.

Process types 1 to 4 are often considered to be associated with discrete material flow, whereas process type 5 is a continuous flow process. The initial choice of process may be represented in a diagrammatical form as shown in Figure 1.3.

Associated with the choice of process is an appropriate equipment configuration for specific industries as illustrated in Figure 1.4 (see Phillips, 1997, Figure 4.1).

As may be seen from Figure 1.4, there is a manufacturing spectrum (Phillips, 1997) based on the degree of flexibility (this term will be discussed below) ranging from *high-volume, low-variety* production (dedicated and/or automated equipment, these include continuous flow lines and the well-known production/transfer lines) to *low-volume, high-variety* production (standard machinery and equipment, these include job-shop systemswith process layout and individual project-based systems; in practice using either stand-alone NC[1] or CNC[2] machine tools or integrated machining centers). The difference between production or flow lines and transfer lines is dependent on the regularity of the movement of material between the stations. In *transfer lines*, known also as *paced lines*, the movement is *synchronous*, whereas in *production or flow lines*, known also as *unpaced lines*, the movement is *asynchronous*. Usually, production lines and transfer lines are one-product lines with a high output. *Continuous flow lines* refer to high-volume production systems where the material process has liquid properties. In a pure *job-shop* environment, a

[1] Numerical control.
[2] Computer numerical control.

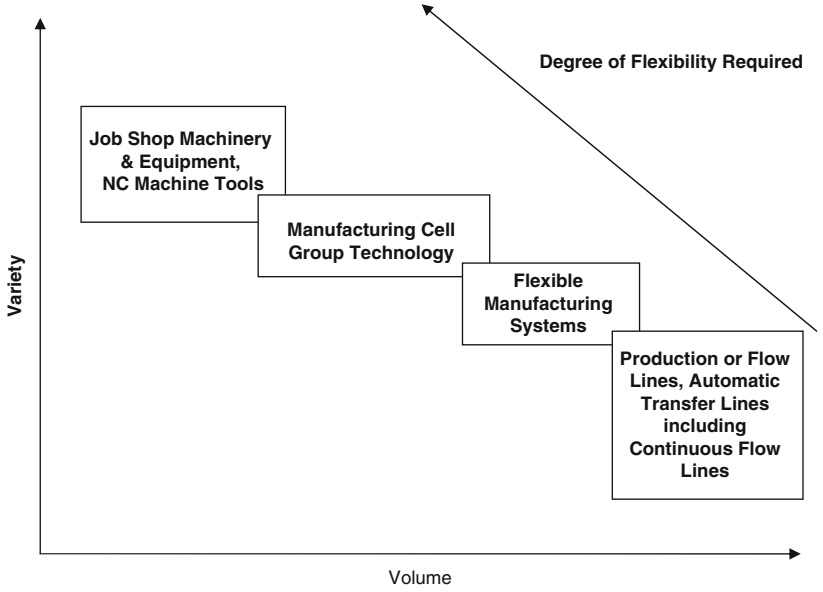


Fig. 1.4. Equipment choice

large variety of products of relatively small volumes are produced. In addition to the two extremes, there is a need for manufacturing systems with a capability of producing mid-range volumes (mid-volume manufacturing) with a significant degree of flexibility. These manufacturing environments are usually catered for by either flexible manufacturing systems (FMS) or flexible manufacturing cells (FMC) including a *group technology* (GT) philosophy of operation.

Essentially, FMS are computer-controlled systems consisting of several stations each specializing in particular operations with an appropriate transport system for the movement of the product. The computer system coordinates the activities, and the essence of FMS is their inherent flexibility. Using an FMS, products may be produced in a number of variations and in different volumes in different time ranges. Historically, FMS were developed because of the high cost of production of small volumes under production line conditions.

As quoted by Schmenner (1990), "*In essence group technology (GT) is the conversion of a job shop layout into a line flow layout. Instead of grouping similar machines together, group technology may call for grouping dissimilar machines together into a line flow process all its own. In the new arrangement, a part can travel from one machine to another without waiting between operations, as would be customary in the job shop.*"

The major benefits of group technology include the rationalization of tooling set-ups, reduction of set-up times, reduction of throughput times and improvements in tool design as well as more efficient production planning and scheduling.

Table 1.1. Processes/products/equipment

Process	Typical Product	Equipment
Project	Airplane, space vehicles (one-offs)	Standard NC, CNC
Job	Instruments, machine tools, prototypes of future products [Low-volume]	Standard NC, CNC machining centers by manufacturing function
Batch	High-end consumer products (e.g. lawn mowers, electric motors, furniture, textbooks) [Mid-volume]	GT cells, focused mini factories, FMS
Line	Telephone screws, light bulbs [High-volume]	Automated equipment, moving assembly lines, flow lines
Continuous	Beer, detergents, chemicals [Very-high-volume]	Continuous-flow fully automated systems

In Table 1.1, a sample list of products is given with the expected associated process choice.

Flexibility is a term that is widely used in the management and engineering literature, often without any great degree of precision of language, and is usually assumed to be a term of approval. In general, whether applied at the overall enterprise or subenterprise level, the concept implies the ability to cope with change. The impression is sometimes given that the existence of any degree of flexibility in industrial organizations, other than zero, is a very modern phenomenon but this is simply not the case.

In an effort to assist in understanding the need for flexibility in enterprises and the different dimensions of flexibility, perhaps it is worthwhile to consider the interests of the customer and the market. An examination of these interests demonstrates the need for the flexibilities required at the enterprise level and consequently at the manufacturing systems level.

Table 1.2 lists the customer/market interests or expectations. These lead to the listed functions of enterprises and these functions imply the five listed flexibilities, arising, as it were, from a customer focus.

Customers expect a rapid response to their demands. Being late to the market with a new competitive product is often much more costly to a firm than significant overruns in either research or development costs. Flexibility in all functional areas of the firm is the key to "*time-based competition,*" a competitive advantage term used to describe efforts to increase innovation, reduce product development time, reduce delivery time and respond "fully" to the individual needs of customers.

The flexibility of a manufacturing system (automatic or manual) is a function of the physical system, its associated software, and how it is operated. Table 1.3, columns 1, 2, and 3 of which are taken with permission from Groover (2001), defines seven types of flexibility (machine, production, mix, product, routing, volume, expansion flexibility) exhibited by manufacturing systems and the factors on

Table 1.2. Specific flexibilities implied by customer interests via enterprise functions

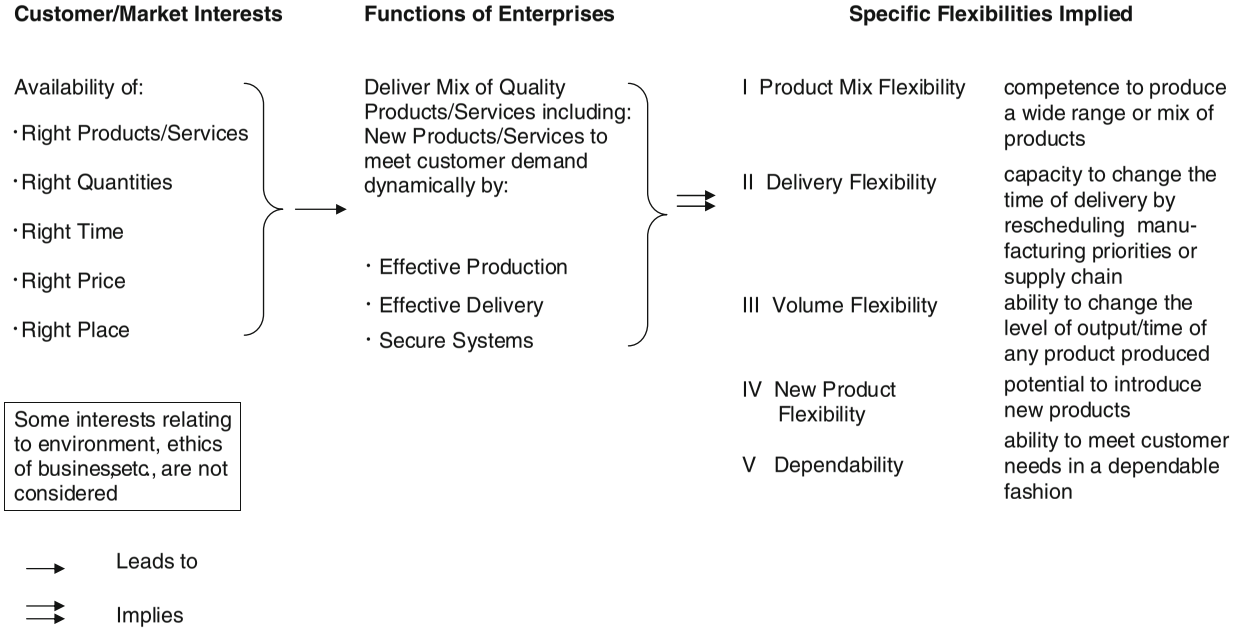


Table 1.3. Types of flexibility in manufacturing

Flexibility Type	Definition	Depends on Factors Such as	Customer Focus Flexibility Type (Table 1.2)
Machine Flexibility	Capability to adapt a given machine (work-station) in the system to a wide range of production operations and part styles. The greater the range of operations and part styles, the greater the machine flexibility.	Setup or changeover time. Ease of machine reprogramming (ease with which part programs can be downloaded to machines). Tool storage capacity of machines. Skill and versatility of workers in the system.	I, IV
Production Flexibility	The range of universe of part styles that can be produced on the system.	Machine flexibility of individual stations. Range of machine flexibilities of all stations in the system.	I, III
Mix Flexibility	Ability to change the product mix while maintaining the same total production quantity; that is, producing the same parts only in different proportions.	Similarity of parts in the mix. Relative work content times of parts produced. Machine flexibility.	I, II
Product Flexibility	Ease with which design changes can be accommodated. Ease with which new products can be introduced.	How closely the new part design matches the existing part family. Off-line part program preparation. Machine flexibility.	IV
Routing Flexibility	Capacity to produce parts through alternative work-station sequences in response to equipment breakdowns, tool failures, and other interruptions at individual stations.	Similarity of parts in the mix. Similarity of work-stations. Duplication of work-stations. Cross-training of manual workers. Common tooling.	V
Volume Flexibility	Ability to economically produce parts in high and low quantities of production, given the fixed investment in the system.	Level of manual labor performing production. Amount invested in capital equipment.	II, III
Expansion Flexibility	Ease with which the system can be expanded to increase total production quantities.	Expense of adding work-stations. Ease with which layout can be expanded. Type of part handling system used. Ease with which properly trained workers can be added.	II, III

which they depend. In column 4 we have added the associated customer focus based flexibilities as defined above. Clearly, engineering and business judgment are required in accessing the "*relative flexibilities*" of manufacturing systems.

The development of information technology played a seminal role in the evolution of manufacturing systems over the past 40 years. Information technology is not only embedded in the equipment being used and the products being produced but it also gives the capability to operations managers to focus on the information flows in the system.

Today, computers are used extensively in design and engineering under the general title of computer-aided design (CAD) and computer-aided engineering (CAE). Similarly, computers are used for production planning and control and tool control using computer-aided manufacturing systems (CAM). Integrated systems including CAD, CAE, CAM, CNC and FMS are referred to as Computer-Integrated Manufacturing Systems (CIM). The manufacturing systems designer must always be conscious of the advantages and disadvantages of using humans or machines for specific operations. It is outside the scope of this work to discuss trends in human-centered automation.

Of particular importance to the operation of manufacturing systems is the distinction between push and pull systems of materials management. In the push system, generally associated with materials requirements planning (MRP), the material is "pushed" through the manufacturing process by the scheduling system and final product is often stored until demanded by customers. Pull systems on the other hand are activated by orders from customers, final and intermediate, and a main characteristic is reduced work-in-process.

Since the 1990s, there has been a strong interest in accurately assessing the cost of products, and an accountancy process known as activity-based costing (ABC) has been developed to accurately determine the manufacturing cost and other costs associated with a particular product or customer.

In recent times, at least three management philosophies of manufacturing have been promoted by consultants and academics. These are "*lean production,*" "*agile manufacturing,*" and "*intelligent manufacturing.*" Womack et al. (1990) described the characteristics of *lean production* as integrated production with low inventories using a just in time (JIT) philosophy, and teamworking with a multi-skilled workforce. In essence the lean manufacturing approach is a combination of JIT and total quality management (TQM) philosophies. Further information may be found in Brown (1996).

The term "*agile manufacturing*" is used to describe a new manufacturing paradigm to replace existing thinking on mass production. There are four principles of agile manufacturing (agility), viz., organize to master change, leverage the impact of people and information, cooperate to enhance competitiveness, and enrich the customer. Agility may be considered a characteristic of the enterprise rather than simply of the manufacturing system. The interested reader is referred to Groover (2001) and Gunneson (1997), among others.

"*Intelligent manufacturing*" is manufacturing, with the minimum of human intervention, by equipment in which is embedded the skills and knowledge of

manufacturing experts so that the products produced are indistinguishable from those produced in conventional manufacturing systems and with similar levels of output and utilization of raw materials and energy. The skills and knowledge of the manufacturing experts (managers, engineers, craft persons and operatives) are embedded in the system by the use of expert systems, databases and data management systems, and intelligent machines such as robots with vision and manipulation possibilities.

As it is clear from the above, decisions in relation to process choice, layout and equipment choice are strategic in nature. These decisions will have a major impact on the long-term viability of the associated company. Such decisions would normally be made before the detailed design of a manufacturing system was undertaken. Of course, any enterprise could make different process choices in relation to different products within its market range or in relation to the same type of product over the manufacturing cycle. For example, a particular product may be made using two or more process choices, i.e., some components of the product may be made on production lines, a few components might be produced in a dedicated manufacturing cell, and the final assembly of the product to customer specification might be performed under flexible assembly system (FAS) conditions. It is unusual for an enterprise to use only one specific "pure" type of manufacturing system, e.g., job shop or transfer line, and a firm may use a mixture of types with the output from one system being an input to another.

A useful summary of the evolution of strategic manufacturing systems is given in Table 1.4, which is taken, with permission, from Ostwald and Munoz (1997).

As may be seen from Table 1.4, the *driving forces* of manufacturing systems have changed over the years from "*cost*" in the 1960s to "*service and value*" in the 2000s. The associated manufacturing strategies have likewise changed from high-volume, cost minimization, and product-focused systems to customer-centered global integration and virtual enterprise systems with a significant concern for the environment and safety.

Systems used to support the strategies have likewise changed over time from an emphasis on production and inventory control systems and numerical control machines in the 1960s to "intelligent" manufacturing systems incorporating flexible and agile automated systems with emphasis on ergonomics and safety systems in the 2000s.

The authors believe that the early decades of this century will see the continuation of a very strong customer-driven intelligent manufacturing (CDIM) paradigm, in which manufacturing will have a strategic focus on catering for the "total" satisfaction of the customer by delivering enlarged products, i.e., physical products plus services, perhaps, delivered via a network of virtual enterprises from different sites on a global basis. Whether the manufacturing community will be satisfied with this role is an open question. Manufacturing expertise has added significantly to the comfort level of human living over the past couple of centuries. It will continue to have a major role in this regard. However, should its mission be confined to this role or should it seek a higher perhaps more spiritual role in assisting to ensure the survival of the human community by realizing the full ambitions of human beings in areas, among others, such as space travel, health care, infrastructure development, and the

Table 1.4. Overview of the evolution of strategic manufacturing systems

	1960s	1970s	1980s	1990s	2000s
Driving Force	Cost	Market	Product Quality	Time to Market	Service & Value
Manufacturing Strategies	High Volume	Functional Integration	Process Control	New Product Introduction	Customer-Centered Mission
	Cost Minimization	Closed Loop	Material Velocity	Responsiveness	Information Sharing
	Stabilize	Automation	World Class Manufacturing	Manufacturing Metrics	Global Integration
	Product Focus	Diversification	Overhead Cost Reduction	Reengineering	Environmental Safety
				Rapid prototyping	Virtual Enterprise
Systems to Support the Strategy	Production and Inventory Control Systems	Material Requirements Planning	Manufacturing Resource Planning	Computer Integrated Manufacturing	"Intelligent" Manufacturing Systems
	Numerical Control	Master Production Scheduling	Just in Time	Decentralization	Flexible and Agile Automated Systems
		Computer Numerical Control	Statistical Quality Control	Simplification	Continuous bench-marking systems
		Push Systems	Computer-Aided Design and Manufacturing	Total Quality Management	Community Involvement
			Simulation	Self-directed Workforce	Continuous Infrastructure Improvement
			Pull Systems	Activity-Based Costing	Paperless Systems
					Ergonomics Safety Systems

reclaiming of the polluted environment? Of course, the manufacturing community could only take on such a higher mission with general political support and leadership. There may well be a limit to the extent to which any community would allow its individual members to selfishly consume for their own satisfaction a portion of the limited manufacturing resources in the context of a customer-driven intelligent manufacturing paradigm when such resources could with advantage be used elsewhere for the benefit of the human community. The resolution of such issues are well outside the scope of this text.

1.2 Models and Modeling

Models are a means for studying phenomena. A useful model yields information about the real system it represents at a lower cost and more quickly than if one undertook experiments on the real system.

Models may be classified in a number of ways. A basic classification is *physical or abstract (mathematical)*. Physical models may be *analogue or iconic*. *Analogue* models exhibit characteristics in some of their variables which are of interest to the model builder. This requires that the underlying physical behavior of the real-life system and of the analogue system are related through a similar set of mathematical equations. For example, a physical vibrating system may be modeled using an electrical network where current in the electrical network corresponds to motion in the physical vibrating system.

Iconic models, on the other hand, are physical models where measurements are made on the physical replicates, often of a reduced scale, of the objects under study. A good example would be architectural models to assist in understanding space utilization.

Abstract or mathematical models are sets of equations with mathematical symbols rather than physical devices. Models are in effect a mental image or an intellectual description of a process. This idea was captured by Robert M. Pirsig in his book (1974) by saying: "*An untrained observer will see only physical labour and often get the idea that physical labour is mainly what the mechanic does. Actually . . . mechanics don't like it when you talk to them because they are concentrating on mental images, hierarchies, and not really looking at you or the physical motorcycle at all . . . . They are looking at underlying form*" (concepts very familiar to Plato).

The initial stage of deriving a model (modeling) is an appropriate simplification or idealization by extracting from the real-life situation those characteristics, properties, or features in which we are interested. This stage may be called the *problem formulation* stage. Once these significant features have been identified, the next stage is to assign mathematical terms to them and to formulate relationships (equations) between these terms. In general, this is not an easy task but it leads to what is normally called the model. A *validation* process takes place throughout the problem formulation stage and after the model itself has been developed. The analyst must have assurance that the problem has been formulated correctly and that the model represents reality appropriately. Clearly, as far as the set of mathematical equations

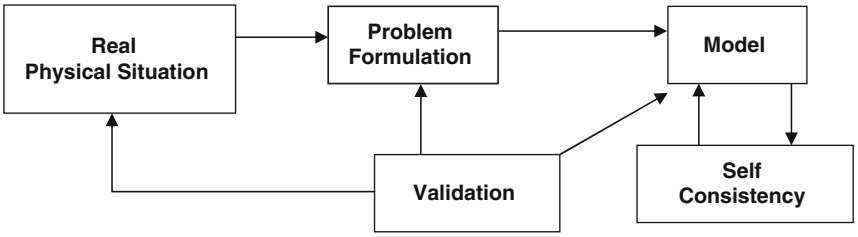


Fig. 1.5. Modeling process

is concerned, there must be *self-consistency*. The overall process may be described as shown in Figure 1.5.

Two quantitative factors, as follows, arise in the development of models:

- *Parameter*, a characteristic or factor which cannot be changed during the analysis of a particular specified system (e.g., the number, K, of work-stations required by process considerations alone in a manufacturing system).
- *Variable*, a factor which assumes more than one value or state during the period of interest.

In modeling, the term *static* implies a relationship that does not change with time, (e.g., the aggregate production planning problem), whereas *dynamic* deals with varying time interactions (e.g., the routing problem in a job shop is a dynamic model). Models may be *linear or non-linear*. *Linear* models have the property of superposition by which is meant that if an input of X produces an output of Y, an input of αX produces an output of αY for any real number α. In *non-linear* models, such a relationship would not hold (e.g., the allocation of a certain amount of buffer space among the work-stations of a production line). *Dynamic* models may be sub-divided into *stable* and *unstable* models, whereas static models must be stable. A *stable* model is one that returns to its original condition after a perturbation, whereas an *unstable* model does not return to its original condition.

Other concepts of interest in modeling are *steady-state* and *transient* solutions. *Steady-state* solutions imply long-term average behavior, whereas *transient* solutions are primarily dependent on the initial conditions of the system. These distinctions are very important in simulation models of manufacturing systems, whereas analytical models of these systems are generally steady-state.

A final distinction may be made between *open* and *closed* models. A *closed* model is one that functions without external input and generates the value of variables through time by the interaction of one variable on another. *Open* systems are open to receive inputs from the outside.

Models may be used in two ways, *on-line* and *off-line*. In some processes a model of the process may be used as a decision support system in real time to assist in control decisions. These *on-line* models, need to be very reliable but usually are relatively simple because of the need to assist the decision maker quickly. *Off-line* models, which are the only ones considered in this text, do not have such constraints.

1.3 Classification of Manufacturing Systems

Our initial objective was to develop a system of classification/notation of manufacturing systems which would be capable of describing comprehensively the essential elements of manufacturing systems particularly from the point of view of the operations manager and the designer of such systems. The classification/notation system would not go into the precise details of each process, i.e., would not go beyond describing a drilling process as a drilling process.

It should be appreciated that in practice, an enterprise may use a combination of what in textbooks are described as "pure" types, of manufacturing systems, e.g., job shop, transfer line, FMC, FMS, etc., in the production of its products. However, major elements of the manufacturing systems used by an individual enterprise could probably be approximately described by some of these "pure" types of manufacturing systems. The detailed classification would be applied in turn to each of these "pure" types and the overall system would be a combination of the individual classifications, maintaining the individual characteristics of the separate elements making up the total system.

Manufacturing systems consist of work-stations at which operations take place. Each work-station may be either *automated or manually operated*. In some systems, because of capacity considerations there may be a number of identical or near-identical work-stations operating in parallel. A basic distinction must be made between manufacturing systems at which only one product is produced (*single-product systems*) and those on which more than one product is produced (*multi-product/mixed-product systems*). In mixed-product systems a particular work-station may carry out a different set of activities depending on the particular product type (e.g., an FMS work-station).

A significant characteristic of manufacturing systems is the difference between fixed and variable routing. In *fixed routing* all piece parts go through the same route, i.e., are processed by each work-station in turn in the same sequence. In *variable routing systems* piece parts are processed through a variety of different station sequences.

Below, a manufacturing systems classification/notation scheme consisting of 12 descriptors, viz., A, B, C, D, E, F, G, H, I, J, K and L, is postulated. This scheme may be applied only to identifiable sections of an enterprise's overall manufacturing system where the sections in question would approximately correspond to the "pure" types of manufacturing systems normally described in operations/manufacturing textbooks. The scheme is essentially numerical in nature using integers. In respect of some of the descriptors, the information is given in the form of a set of numbers which may be arranged in a row (row vector) or in an array (matrix). The usual convention about the dimensions and components of row vectors and matrices are used. A scalar is an ordinary number, in this case, an integer.

A: Number of separate work-stations not including identical or near identical work-stations in parallel.

B: Number of *different products* which may be produced by the manufacturing system.

C: A row-vector of dimension 1 × **A** giving the number of operations that can be undertaken at each work-station.

D: A set of size **A** vectors/scalars, one for each work-station, indicating the type of operations which may be undertaken at each work-station, e.g., 1 = processing operation, 2 = assembly operation, 3 = materials handling, transportation and storage, 4 = product quality assurance, inspection and test, and 5 = process control (a finer degree of operation classification is possible). The dimension of each vector in set D is determined from the corresponding elements of C. The information contained in descriptors C and D may be given by one descriptor alone.

E: A row-vector of dimension 1 × **A** with each element corresponding to each of the work-stations in the system with components 0 or 1, where 0 indicates *manual* and 1 indicates *automatic*.

F: A row-vector of dimension 1 × **A** which indicates the total number of identical or near identical machines or work centers working in parallel associated with each of the separate work-stations listed in A.

G: A set of row-vectors, one for each product, giving the sequence of work-stations on the preferred route, according to a numbering scheme for the work-stations in the manufacturing system. This scheme must be clearly defined.

H: A set of row-vectors each member of which is associated with each element of vector **C** which is greater than 1, indicating the *micro-route* for each product using the associated work-station. The control systems of FMS and machining centers could complicate this descriptor.

I: A set of row-vectors, one for each product, giving the expected processing time spent in each work-station on the preferred route in accordance with the numbering scheme of the work-stations as specified in descriptor **G**. Descriptor **I** is obtained using information contained in **G** and **H** and the relevant expected processing times.

J: A set of row-vectors, each of dimension 1 × 2, for each work-station indicating the number of its incoming and outgoing buffers.

K: A set of row-vectors, one for each work-station, indicating the maximum capacities (sizes) of its associated incoming buffers from other work-station(s) (capacity is based on the mix of products, i.e., a "standard" product).

L: A set of row-vectors for each work-station indicating the capacities (sizes) of its associated outgoing buffers to other work-station(s) (capacity is based on the mix of products, i.e., "standard product").

One needs to be careful when discussing the number of work-stations in a manufacturing system from the viewpoint of an individual product.

This classification system, developed above, although perhaps somewhat academic and which, in some circumstances, could be open to criticism on the basis of lack of completeness, clearly illustrates the complexity of manufacturing systems in general and demonstrates the need to confine one's attention in the context of today's computing abilities to well-defined and well-structured manufacturing systems. It is clear that manufacturing systems are significantly more complex than the queueing systems found in textbooks often described by Kendall's well-known notation.

1.4 Models of Manufacturing Systems

Investments in manufacturing are generally considered strategic because of their size and impact for the enterprise concerned. Once a proposal has been made to invest in such systems, there is a need to agree on the outline conceptual design of the manufacturing system. Subsequently, the detailed design of the system and modes of operation of the system must be developed. Finally, appropriate control strategies of the manufacturing system have to be designed. It is clear therefore that there is a need for a large number of different types of models to assist the different actors (business analysts, design engineers, operations managers, finance managers, marketing managers and manufacturing personnel) in their decision-making tasks. Among the reasons why one large overall model of a manufacturing system is not realistic are

- the physical and information flow complexities of such systems as partially illustrated in Section 1.3;
- the legitimate different interests of the various actors involved;
- all design is an iterative process with information becoming available as a sequence of decisions are made. Accordingly, there is, in effect a hierarchy of decisions which in turn leads to a hierarchy of appropriate models.

Arising from the above considerations, three generic types of models may be described:

- Planning model
- Design model
- Control/operation model

The *planning model* is used to test initial assumptions in relation to such issues as the number and type of work-stations, the type of transportation systems and the information and control systems to meet the business requirements in relation to products, finance and return on investments. In planning models the level of detail is usually low as precise detail of the system has not been developed fully. An economic justification model falls into this category.

The *design model*, having as input the information derived from the planning model, is used to determine such matters as the location and size of inventory buffers, the number of parallel work-stations in a series-parallel structure, the work-load allocation among the work-stations, the details of the physical transportation system, the number of pallets and the tool storage capacity. The control strategy of the system needs to be specified and tested at this stage. The level of detail of these models is considerably higher than in the planning models.

The *control/operation model* is normally based on a fully specified physical manufacturing system. These models are used in the day to day operation of the manufacturing system to examine such operating issues as the input control (determination of the sequence and timing of the release of jobs to the system), the scheduling of work on each work-station and the behavior of the transportation systems between work-stations.

Planning models and design models are applicable to a large variety of manufacturing systems, whereas control/operation models are normally developed for job shops, FMS, flexible manufacturing cells (FMC) and flexible assembly systems (FAS).

Although the level of detail increases from planning model through design model to control/operation model, it is unlikely that a model at one level is a simple upgrading and enhancement by way of greater detail of a model at a previous stage. Models at the various levels may be used to assist the development and validate the models at the higher levels of detail.

Modeling practice has given rise in the literature to a description of two types of models, as follows:

- *Evaluative or predictive models* in their basic form assume a particular configuration of the manufacturing system under study and performance measure(s) of the system are determined/evaluated. Such models could be used as planning models or design models in the context of the classification already given.
- *Generative or optimization models* have as their basic purpose the determination of an optimal solution to the system parameters given an overall manufacturing system structure and an objective function to be optimized. In particular, two types of generative models are used in manufacturing systems analysis: (i) models used to determine specified parameters of the systems to maximize throughput and (ii) models used to determine specified parameters of the system that minimize a specified cost objective while achieving a feasible target throughput. Generative models are particularly useful in determining optimal design specifications.

There is a great potential for a synergistic relationship between these two types of models as illustrated in Figure 1.6, which is taken, with permission, from Papadopoulos et al. (1993).

The evaluative model, as noted above, does not necessarily give the user an optimal solution but instead evaluates certain decisions leading to the determination of performance measures of the system. The generative model, in turn, seeks to maximize these performance measures subject to overall constraints that are globally consistent with the decisions inherent in the evaluative models.

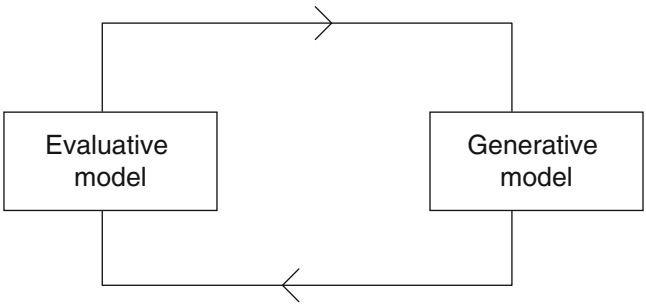


Fig. 1.6. Synergistic relationship between evaluative and generative models

An example may clarify the issues involved. A manufacturing system has been specified by a certain number of work-stations and a configuration of same with a given transportation system. In the evaluative model, the capacity of the storage locations (buffers) associated with each machine is specified as are the product range, product mix, product routes and production control mechanism. The throughput or other measures of performance of the system is determined by the evaluative model. The generative model, in turn, considers a given total amount of storage space allocated to the buffers to be a constraint and finds the maximum possible throughput with the decision variables being the allocated buffer slots (space) to each buffer associated with each work-station consistent with the overall amount of storage space being allocated.

The main thrust and focus of this text is toward design models of production lines, although a number of the modeling techniques presented and algorithms described could with some modifications be used as a basis for planning models and to a lesser extent would be of value in developing operations models of manufacturing systems.

1.5 Methods of Analysis

There are two distinct approaches to the analysis of models of manufacturing systems, *simulation methods and analytical methods*. The *simulation method* involves the representation of the real manufacturing system in a computer-based model via the use of an appropriate simulation package such as Arena or eM-plant. Certain computer packages are more suitable for simulating specific types or parts of manufacturing systems such as FMS or materials handling systems. A major problem with simulation is the validation of the model, particularly if the manufacturing system is not actually built. The solution to evaluative models may be obtained through simulation.

Analytical methods on the other hand involve formal mathematical solutions to the problems. Because of the complexity of the mathematical models involved, two approaches to obtaining a solution are used. An *exact*, sometimes closed form solution, may be obtained to a simplified problem or an *approximate* solution may be derived often by means of an appropriate and efficient algorithm to the actual mathematical problem.

It is instructive to consider the diagrams shown in Figures 1.7 to 1.11, developed from ideas presented in Archetti et al. (1989), which gives a comparison of simulation and analytical methods of solution of models.

Simulation methods are capable of handling more complex model structures than are analytical methods, particularly than are those models associated with exact analytical solutions. In respect of flexibility, it is relatively easy to change the values of the parameters in analytical models but difficult to change the structure of these models. As far as simulation models are concerned, parameter values and the structure may be modified with some degree of difficulty. Transparency, as a characteristic of solution methods, may be considered from the point of view of the modeler or of the user. As far as users are concerned there is a low degree of transparency in regard to

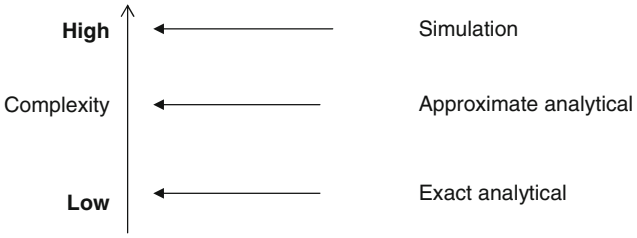


Fig. 1.7. Complexity of the model

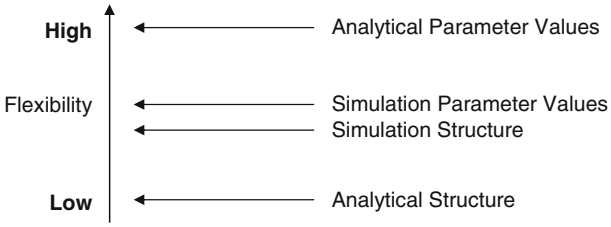


Fig. 1.8. Flexibility of the model

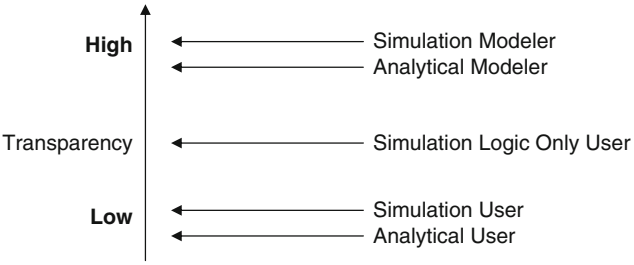


Fig. 1.9. Transparency to the modeler and to the user

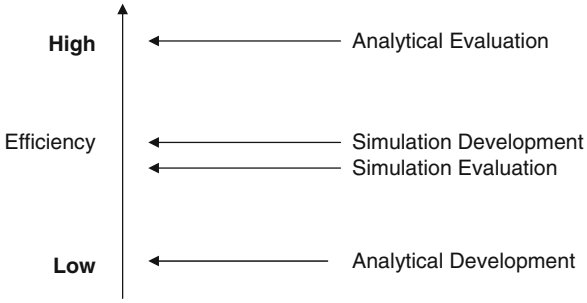


Fig. 1.10. Efficiency of model development and evaluation

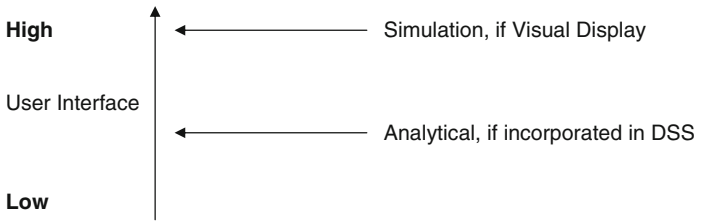


Fig. 1.11. User interface

both solution methods, although the user of simulation models may appreciate the logic of the model better. The modeler of course in both cases would fully understand the solution method. Efficiency has two dimensions: (i) in respect of the development of the model and the solution process and (ii) the efficiency of evaluation. As far as simulation solutions are concerned, both the evaluation and the development are relatively efficient, whereas in the case of analytical methods the development process tends to have a low efficiency but the evaluation of the model is often highly efficient. Finally, in regard to the user interface, the simulation model is highly user friendly if a visual display unit is part of the simulation solution, and analytical models may be made more user friendly by incorporation into easy to operate decision support systems (DSS).

Mathematical techniques found useful in the solution to the models under discussion include queueing theory (single/multiple station models and queueing network models), Markovian models, graph theory, Petri net models, mathematical programming models (non-linear programming, stochastic programming, dynamic programming), search methods, gradient techniques, perturbation methods, simulated annealing methods, Tabu search, various heuristic algorithms, and genetic programming, among others.

A rather different approach to the development of solutions to problems associated with the design and operation of manufacturing systems is the artificial intelligence (AI) expert systems based techniques. The expert system software is embedded with knowledge obtained from manufacturing specialists and arranged according to a rule base. These techniques have been mainly applied to operational problems particularly where heuristic solutions are the norm. Such methods are outside the scope of this work.

1.6 Measures of Performance

The typical performance objectives of manufacturing operations from the point of view of either the operations manager or the customer are generally listed under the five headings of quality, speed, dependability, flexibility and cost. Arising out

of these objectives, the following are some of the (technically based) performance measures commonly used with respect to manufacturing systems:

- *Throughput* or *mean production rate* or *mean output rate*, X, is the expected number of parts produced per time unit, given a specified product mix.
- *Mean production time*, $1/X$, is the expected time a "standard" product spends in the system from the time of its entry to the time of exit. The "standard" product concept takes into account the product mix. The mean production time is the reciprocal of the throughput.
- *Mean work-in-process (progress)*, $\overline{WIP}$, is the number of parts present in the whole system (being processed in the work-stations and awaiting for processing). For clarification, it should be noted that transportation within the manufacturing system should be considered a process in this context.
- *Utilization*, ρ_i, of a work-station i is the proportion of time that the work-station is busy (processing parts).
- *System or global utilization* is the mean of the utilizations of the work-stations making up the system (including work-stations operating in parallel). For operational reasons, the utilization of transport systems is often neglected in calculating global utilization, particularly in FMS.
- *Availability*, A_i, of a work-station i is the proportion of time the work-station is capable of processing parts whether required to do so or not. A work-station can break down and so is not available during the repair period.
- *Efficiency*, e_i, of a work-station i is the ratio of the mean output rate of the work-station as part of the manufacturing system divided by the mean output rate achievable without the constraints of being embedded in such a system. When part of a manufacturing system, a work-station may be blocked from processing parts or starved because no parts are available for processing thus affecting the arrival pattern of parts.
- *Blocking time proportion* for a work-station embedded in a manufacturing system producing a "standard" mix of products is the proportion of time a particular work-station, although available for processing, is unable to continue the processing of parts because its output buffer or the next work-station, in case where there is no buffer, cannot accept any more product.
- *Holding time* or *completion time* of a work-station for a "standard" product is the product of the "standard" processing time by (1 plus the blocking time proportion).
- Other measures of performance of manufacturing systems include: the *mean number of busy work-stations* of any set of work-stations working in parallel, the *mean queue lengths* of "standard" products awaiting processing at each work-station, the *mean waiting time* for a "standard" product before being processed at a particular work-station including waiting for transportation between work-stations and measures of the relative importance of *set-up time*, particularly in job shop environments, among others.

1.7 Related Bibliography

There are many excellent textbooks covering the general areas of production and operations management. Early classics include Wild (1985) and Buffa (1973). These textbooks tended to emphasize the tactical aspects of operations management and covered material such as product design, quality, inventory and aggregate production planning, line balancing and forecasting, among others. Subsequently, emphasis was given to quality assurance and materials requirements planning in such texts as Montgomery (1992), Evans and Lindsay (1996), Orlichy (1975), Waters (1998), Waters (2006) and Noble (1986). In more recent times, arising from the greater emphasis on the strategic importance of manufacturing and operations particularly in MBA executive programs, the texts of Russell and Taylor (2005), Evans (1997), and Anderson, Sweeney and Williams (1991) were widely read.

An excellent overall text with a strong quantitative orientation is Groover (2001). Specific texts related to probability and queueing systems include Gross and Harris (1998), Feller (1961), Feller (1991), and Parzen (1962) as well as classic books in operations research including Taha (2002), Hillier and Liebermann (2005), Solberg et al. (1987), and Perros (1994), among others.

There are a large number of texts devoted to the subject of simulation modeling such as Law and Kelton (1999), Pidd (2004) as well as texts devoted to specific simulation languages such as Arena (Kelton et al., 1998), and eM-Plant (`http://www.plm.automation.siemens.com/en_us/products/tecnomatix/`).

References

1. Archetti, F., Lucertini, M. and Serafini, P. (1989), *Operations Research Models in Flexible Manufacturing Systems*, Springer.
2. Anderson, D.R., Sweeney, D.J. and Williams, T.A. (1991), *Introduction to Management Science: Quantitative Approaches to Decision Making*, 6th Edition, West Publishing Company.
3. Brown, S.E. (1996), *Strategic Manufacturing for Competitive Advantage, Transforming Operations from Shop Floor to Strategy*, Prentice Hall.
4. Buffa, E.S. (1973), *Modern Production Management*, John Wiley.
5. Buzacott, J.A. (1989), Flexible models of flexible manufacturing systems, in *Operations Research Models in Flexible Manufacturing Systems*, edited by F. Archetti, M. Lucertini and P. Serafini, Springer.
6. eM-Plant, http://www.ugs.com/products/tecnomatix/plant_design/em_plant.shtml.
7. Evans, J.R. (1997), *Production & Operations Management: Quality, Performance and Value*, 5th Edition, West Publishing Company.
8. Evans, J.R. and Lindsay, W.M. (1996), *The Management and Control of Quality*, 3rd Edition, West Publishing Company.
9. Feller, W. (1961), *An Introduction to Probability Theory and Its Applications, Volume I*, John Wiley.
10. Feller, W. (1991), *An Introduction to Probability Theory and Its Applications, Volume II*, 6th Edition, John Wiley.

11. Groover, M.P. (2001), *Automation, Production Systems, and Computer Integrated Manufacturing*, Second Edition, Prentice Hall.
12. Gross, D. and Harris, C.M. (1998), *Fundamentals of Queueing Theory*, 3rd Edition, John Wiley.
13. Gunneson, A.O. (1997), *Transitioning to Agility; Creating the 21st Century Enterprise*, Addison-Wesley Publications.
14. Hillier, F.S. and Liebermann, G.J. (2005), *Introduction to Operations Research*, 8th Edition, McGraw-Hill.
15. Kelton, W.D., Sadowski, R.P. and Sadowski, D.A. (1998), *Simulation with Arena*, McGraw-Hill.
16. Law, A. and Kelton, W.D. (1999), *Simulation Modeling and Analysis*, 3rd Edition, McGraw-Hill.
17. Montgomery, D.C. (1992), *Introduction to Statistical Quality Control*, Second Edition, John Wiley.
18. Noble, D.F. (1986), *Forces of Production: A Social History of Industrial Automation*, Oxford University Press.
19. Orlichy, J. (1975), *Materials Requirements Planning*, McGraw-Hill.
20. Ostwald, P.F. and Muñoz, J. (1997), *Manufacturing Processes and Systems*, Ninth Edition, Wiley.
21. Papadopoulos, H.T., Heavey, C. and Browne J. (1993), *Queueing Theory in Manufacturing Systems Analysis and Design*, Chapman & Hall.
22. Parzen, E. (1962), *Modern Probability Theory and Its Applications*, John Wiley.
23. Perros, H.G. (1994), *Queueing Networks with Blocking: Exact and Approximate Solutions*, Oxford University Press.
24. Phillips, E.J. (1997), *Manufacturing Plant Layout: Fundamentals and Fine Points of Optimum Facility Design*, Society of Manufacturing Engineers (SME).
25. Pidd, M. (2004), *Computer Simulation in Management Science*, 5th Edition, John Wiley.
26. Pirsig, R.M. (1974), *Zen and the Art of Motorcycle Maintenance*, William Morrow & Company and the Bodley Head.
27. Russell, R.S. and Taylor III, B.W. (2005), *Operations Management*, 4th Edition, Prentice Hall.
28. Schmenner, R. (1990), *Production/Operations Management*, Macmillan.
29. Phillips, D.T., Ravindran, A. and Solberg, J.J. (1987), *Operations Research*, 2nd Edition, John Wiley.
30. Taha, H.A. (2002), *Operations Research: An Introduction*, 7th Edition, Prentice Hall.
31. Waters, C.D.J. (1998), *A Practical Introduction to Management Science*, Addison-Wesley.
32. Waters, C.D.J. (2006), *Operations Strategy*, Thomson Learning.
33. Wild, R. (1985), *Essentials of Production and Operations Management*, 2nd Edition, Holt, Rinehart and Winston.
34. Womack, J., Jones, D. and Roos, D. (1990), *The machine That Changed the World*, MIT Press (Rawson Associates, N.Y.).

2

Evaluative Models of Discrete Part Production Lines

The focus here is on discrete part production lines with asynchronous movement where each part produced is distinct. Production lines processing fluids and other continuous materials are not considered. From here on, when reference is made to production lines, discrete part production lines will be understood. In a production or flow line, all jobs are required to pass through each station in the same sequence once. These lines are usually associated with scale rather than scope, and a major advantage of production lines is the associated simple materials handling requirements.

A production line consists of work-stations, materials, human resources, and inter-work-station storage facilities. Storage facilities have a finite capacity. Randomness is introduced due to random processing times and the random behavior of work-stations in relation to failure and repair. In terms of classical queueing theory, production lines would be described as finite buffer tandem queueing systems where the work-stations are the servers, storage facilities are the buffers or the waiting lines, and the jobs are the customers.

In Figue 2.1, which depicts a K-work-station production line, $WS_i, i = 1,2,\ldots,K$ represents work-station i and B_i, $i = 1,2,\ldots,K$ denotes the buffer capacity of the buffer located in front of station WS_i. As there are K work-stations, there are $K-1$ intermediate buffers. As described in Chapter 1, the goal of evaluative models is to calculate some performance measures of the system under study. The most usual performance measure determined is throughput. Each station may consist of a single perfectly reliable machine or an unreliable machine or a number of identical parallel reliable or unreliable machines. For notational purposes only, it should be understood that the word "machines" may cover operators.

In Section 2.1, Markovian analysis of production lines is presented using the underlying queueing system structure of production lines. It produces an exact analysis of such lines. In sub-section 2.1.1, a numerical approach is presented for solving the system of linear equations derived from the Markovian analysis. In sub-section 2.1.2, an algorithm is given for the generation of the conservative matrix A for the case of an exponential production line with inter-station buffers. In sub-section 2.1.3, a simple merge model of a two-station production line with merge operations (a non-linear model) is analyzed using exact Markovian methods.

C. T. Papadopoulos et al., *Analysis and Design of Discrete Part Production Lines*, Springer Optimization and Its Applications,
DOI: 10.1007/978-0-387-89494-2_2,

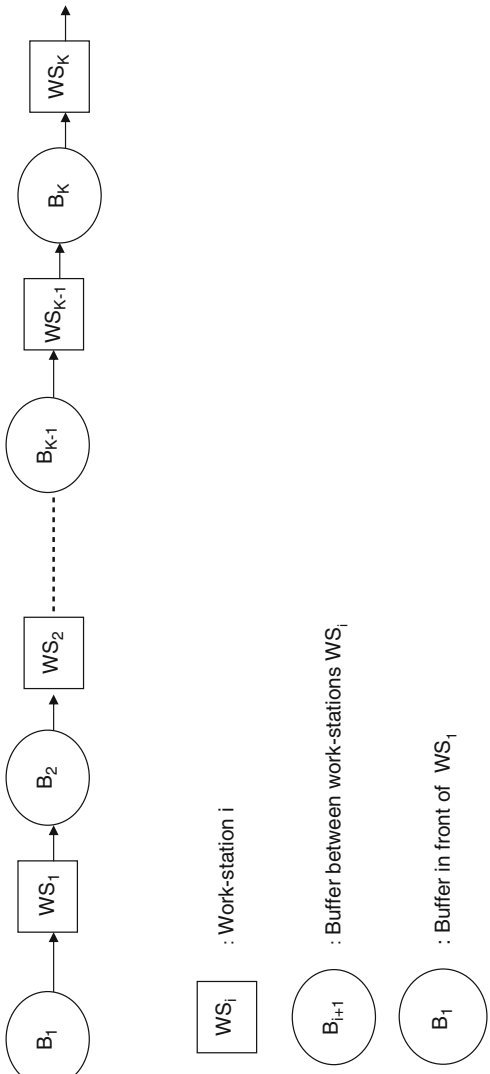


Fig. 2.1. A K-work-station production line

However, even for linear production lines with a large number of stations ($K > 6$) and reasonable buffer sizes, it is not possible to develop exact numerical results due to the complexity of the numerical calculations involved. As a result of this restriction, approximate solutions were sought. The decomposition approach is described in Section 2.2. Essentially, the process involves the decomposition of a large production line into a number of smaller lines with suitable provision for their inter-connection so that the behavior of the inter-connected system approximates the behavior of the original large production line.

Another approximation technique named the expansion method is given in Section 2.3.

Although the focus up to now has been on prefectly reliable machines at each station, it should be noted that the Markovian and decomposition methods can each handle unreliable machines. The aggregation method, which is a different approximation approach, was specifically developed to analyze transfer lines with asymptotically reliable stations. The aggregation method is covered in Section 2.4.

Up to this point the models used have been of a serial type, in the reliability sense, such that if a particular work-station was not operating due to breakdown or otherwise, the work-stations downstream from that particular work-station would eventually be starved. Work-stations in parallel were introduced with the result that the breakdown of a particular work-station would not necessarily lead to the starvation of stations downstream. The solution of production lines with parallel machines at each work-station is given in Section 2.5. The exact analysis of a simple parallel-machine production line consisting of two work-stations is presented in sub-section 2.5.1. An alternative exact analysis for solving the same two-station production line with parallel machines at each station which serves as building block in decomposing larger lines is given in sub-section 2.5.2. In sub-section 2.5.3 the approximate solution using decomposition method of large serial production lines with parallel-machine stations is given.

Simulation is often used when analytical methods prove intractable or are confined to rather simplified assumptions. In the case of production lines, simulation models may be used to assess the results of all approximate models and to obtain results using distributions for processing, failure, and repair times other than exponential or phase-type. Such models are explored in Section 2.6.

2.1 Markovian Model

Consider the model as depicted in Figure 2.1. Jobs enter station 1 from buffer B_1 of unrestricted capacity according to a Poisson distribution with arrival rate λ. Each job enters the line at station 1, passes through all stations in order and leaves the K^{th} station (last) in finished form. All jobs at each station are processed according to a First-In-First-Out (FIFO) queueing discipline.

The assumptions of the model are summarized below:

- (i) The processing or service times are exponentially or Erlang distributed random variables with mean rates equal to μ_i, i = 1, 2, ..., K. In general, the service rates need not be identical (i.e., $\mu_i \neq \mu_j$ for $i \neq j$).
- (ii) All buffers between successive stations have finite capacities not necessarily of the same size.
- (iii) Blocking of a station occurs if the downstream buffer is full at the time of service completion.
- (iv) A station may be assumed to be perfectly reliable or subject to random failure according to an exponential distribution with mean rates equal to β_i, i = 1, 2, ..., K. In general, the failure rates need not be identical (i.e., $\beta_i \neq \beta_j$ for $i \neq j$). However, it is assumed that a failure of a station can only occur when it is operating, i.e., operational-dependent breakdowns.
- (v) If a station fails, the part which the station was processing remains at the station, i.e., it is not placed in the preceding buffer.
- (vi) Once a failed station is repaired, it resumes processing at the same phase of service at which it failed, on the job that was not completed, and as a result of the memoryless property of the exponential distribution, the remaining processing time in that phase is exponentially distributed.
- (vii) The repair times are exponentially or Erlang distributed random variables with mean rates equal to r_i, i = 1, 2, ..., K. In general, the repair rates need not be identical (i.e., $r_i \neq r_j$ for $i \neq j$).
- (viii) The general rule that deliberate idleness at a station is not allowed applies.
- (ix) A basic assumption is that the first station is never starved and the last station is never blocked. Although the arrival process is assumed to be Poisson, it is a necessary assumption of the model that the first station is never starved. This assumption characterizes the *saturated* line of the saturation model. The fact that the last station is never blocked relates to the storage capacity for final products.

The system under consideration is a two-dimensional stochastic process $N(t) = [N_1(t), N_2(t)]$. Both coordinate random variables are integer valued and nonnegative. $N_1(t)$ represents the number of jobs queued up in front of the first station at time t, and N_1 is the expected value of this quantity at equilibrium (the limit of $N_1(t)$, as t tends to infinity). There is no upper limit for N_1. $N_2(t)$ represents the state of the sub-network at time t, which consists of stations $1, 2, 3, \ldots, K$ and the intermediate buffers. In effect, $N_2(t)$ is a vector representing the situation in each station and in each of the intermediate buffers of the production line at time t. The number of states in the sub-network equals m, for some finite m. When $N_1 = 0$, the number of states in the sub-network equals m_0, $m_0 < m$.

The changes in the state of the system are caused by the occurrence of various events. The occurrence times for all events have negative exponential or Erlang distributions with strictly positive means. Thus the process is Markovian. Its state-space is $S = \{(i, j) : i \geq 0, 1 \leq j \leq m\}$ with the index i specifying the total number of jobs queued up at the first station. Such customers are called "I-customers." The index j

determines the state of the sub-network (Unit-II). It is important to note that **upon entering service at station 1, a customer becomes a "II-customer."**

The transition matrix P that describes the model has the following block tri-diagonal form:

$$P = \begin{vmatrix} A_{01} & A_0 & & & & & \\ A_2 & A_1 & A_0 & & & & \\ & A_2 & A_1 & A_0 & & & \\ & & A_2 & A_1 & A_0 & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \cdot \end{vmatrix} \tag{2.1}$$

and the equilibrium equations $\pi P = 0$ can be expressed in matrix-difference form as

$$\pi_k A_0 + \pi_{k+1} A_1 + \pi_{k+2} A_2 = 0 \tag{2.2}$$

for $k = 0, 1, 2, \ldots$ and

$$\pi_0 A_{01} + \pi_1 A_2 = 0 \tag{2.3}$$

for the boundary equations, where,

A_0 is an (m $\times$ m)-matrix describing the transitions in the sub-network, which simultaneously produce inputs to the first queue.

A_1 is an (m $\times$ m)-matrix describing transitions in the sub-network which produce neither inputs to nor outputs from the first queue assuming that the queue is *not empty*.

A_2 is an (m $\times$ m)-matrix describing transitions in the sub-network which simultaneously produce outputs from the first queue, and

A_{01} is an (m $\times$ m)-matrix describing transitions in the sub-network which produce neither inputs to nor outputs from the first queue, assuming that the queue is *empty*.

A Markov chain whose equilibrium equations have the form of equations (2.2) and (2.3) is known as a Quasi-Birth and Death (QBD) process.

Let P_2 be the steady-state probabilities of the sub-network, assuming that the first queue is never empty. Solving the equations

$$\begin{aligned} P_2 A &= 0 \\ P_2 \mathbf{e} &= 1 \end{aligned} \tag{2.4}$$

where A is the conservative stable matrix given by

$$A = A_0 + A_1 + A_2 \tag{2.5}$$

and $\mathbf{e}$ is an (m $\times$ 1) column-vector, with all elements equal to 1, will give explicit results for $P_2(j)$, $j = 1, 2, \ldots, m$.

The equilibrium condition is given by

$$P_2 A_2 \mathbf{e} > P_2 A_0 \mathbf{e}. \tag{2.6}$$

Table 2.1. Notation

Symbol	*Meaning*
K	Number of stations
B_i	Buffer capacity preceding the i^{th} station. Note: when $B_i = B_j$ for all i, then the buffer capacity is denoted by B
n_i	Status of buffer i
s_i	Status of station i
P_i	Denotes the number of phases of the service (processing) distribution of the i^{th} station
R_i	Denotes the number of phases of the repair distribution of the i^{th} station
$m_{K,P}^{B,R}$	Number of states in the sub-network with K stations, each buffer having the same capacity B, each service distribution having P phases and each repair distribution having R phases
$m_{K,P_1,P_2,..,P_K}^{B_2,..,B_K,R_1,R_2,..,R_K}$	Number of states in the sub-network of a K station system with buffer capacities B_2, $\ldots,B_K$. The number of phases of each station's service distribution is equal to $P_1,P_2,\ldots,P_K$ phases and the number of phases of each station's repair distribution is equal to $R_1,R_2,\ldots,R_K$

From this relationship the critical mean input rate (λ^*) to the system can be determined. In the steady-state, this critical input rate is identical to the maximum throughput rate of the production line. By calculating the throughput of the system as outlined above, we exclude the states of the system where the first station is empty, i.e., sub-matrix A_{01} is not included. Therefore, the throughput of the system is governed by the assumption that the first queue is never empty (saturation model).

The notation used is shown in Table 2.1.
The states of the sub-network are described by the following vector:

$$(s_1,n_2,s_2,n_3,\ldots n_K,s_K) \tag{2.7}$$

where, s_i can take any value from 0 to $(P_i + R_i \times P_i)$:

$$\begin{aligned} s_i &= 0 \text{ – station is idle,} \\ s_i &= 1,\ldots,P_i \text{ – station is in service,} \\ s_i &= (P_i+1),\ldots,(P_i+P_i\times R_i) \text{ – station is in repair.} \end{aligned}$$

After a station has been repaired, it is assumed that service is resumed at the phase in which the station was interrupted. Therefore, there is a need to keep a record of the phase of service in which the station was interrupted. It should be noted that the need to account for the phase is a modeling requirement and may not have any

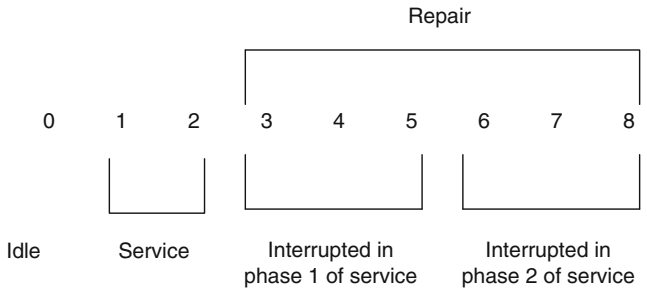


Fig. 2.2. The states of s_i for $P_i = 2, R_i = 3$

corresponding physical meaning. This results in using $(P_i + P_i \times R_i)$ states to describe a station's repair process. It also necessitates the use of equation (2.8) to transfer s_i from a state in service to a state in repair and equation (2.9) to do the reverse.

$$\text{Beginning repair state} = (P_i + ((s_i - 1) \times R_i) + 1) \tag{2.8}$$

$$\text{Phase to resume service at} = \frac{(s_i - P_i)}{R_i}. \tag{2.9}$$

The states that s_i can take, for the parameters $P_i = 2$ and $R_i = 3$, are illustrated in Figure 2.2.

n_i can take values from 0 to $(B_i + 1)$. The values from 0 to B_i denote the number of items in buffer B_i with B_i also denoting the capacity of buffer B_i. When $n_i = (B_i + 1)$, station $(i - 1)$ is blocked.

The following recursive relationship was obtained (in Heavey, Papadopoulos and Browne, 1993) to calculate the number of, states of a system with K stations with parameters P_i, R_i, B_j, $i = 1, \ldots, K$, $j = 2, \ldots, K$.

The number of states for a two-station system is first calculated using the parameters $P_{K-1}, P_K, R_{K-1}, R_K, B_K$.

Two-station system:

$$\Xi_1 = (P_K + (P_K \times R_K) + 1) \tag{2.10}$$

$$\Xi_2 = (((P_{K-1} \times (B_K + 1)) + (P_{K-1} \times R_{K-1} \times (B_K + 1))) + 1) \tag{2.11}$$

$$\Xi_3 = ((B_K \times (P_{K-1} + (P_{K-1} \times R_{K-1}))) + 1) \tag{2.12}$$

$$\Omega_2 = ((\Xi_1 \times \Xi_2) - \Xi_3). \tag{2.13}$$

Ω_2 will equal the number of states of the system if $K = 2$. To calculate the number of states for systems with $K > 2$, the following recursive scheme is used. Before entering the loop below, the variable Ω_1 is set equal to $(P_K + P_K \times R_K)$ and the variable

Table 2.2. Number of states for $P = 1, R = 1$ and identical buffer capacities

# *of Stations*	*Buffer Size*				
	0	1	2	3	4
2	8	12	16	20	24
3	30	70	126	198	286
4	112	408	992	1,960	3,408
5	418	2,378	7,810	19,402	40,610
6	1,560	13,860	61,488	192,060	483,912
7	5,822	80,782	484,094	1,901,198	5,766,334
8	21,728	470,832	3,811,264	18,819,920	68,712,096
9	81,090	2,744,210	30,006,018	188,119,920	818,778,818

Ω_2 takes its value from equation (2.13), with the parameters $P_{K-1}, P_K, R_{K-1}, R_K, B_K$ of the K-station system used in equations (2.10), (2.11) and (2.12).

DO $I = (K-1)$ to $2, -1$

$$Y_1 = ((\Omega_2 - \Omega_1)/(P_I \times (R_I + 1))) + \Omega_1 \tag{2.14}$$

$Y_2 :=$ Using parameters N_I, P_I, P_{I-1}, R_I, R_{I-1} calculate the number of states for a two-station system as above, i.e., let $B_K = N_I, P_{K-1} = P_{I-1}, P_K = P_I$, $R_{K-1} = R_{I-1}, R_K = R_I$ in equations (2.10), (2.11), (2.12) above. (2.15)

$$Y_{31} = (P_{I-1} + P_{I-1} \times R_{I-1})$$
$$Y_{32} = (P_I + P_I \times R_I)$$
$$Y_{33} = (B_I \times (Y_{31} \times (Y_{32} - 1)))$$
$$Y_3 = ((Y_{31} \times Y_{32}) + (Y_{32} - 1) + Y_{33}) \times \Omega_1 \tag{2.16}$$
$$\Omega_1 = \Omega_2$$
$$\Omega_2 = ((Y_1 \times Y_2) - Y_3)$$

END DO I

$$m_{K,P_1,..,P_K}^{B_2,..,B_K,R_1,R_2,..,R_K} = \Omega_2 \tag{2.17}$$

Table 2.2 gives the number of states for K-station systems, with identical buffers equal to B, $P = 1$ and $R = 1$. As one can observe from Table 2.2, the number of states becomes very large, even for relatively small systems.

2.1.1 A numerical approach

As is well known, there are a number of ways of solving sets of homogeneous linear equations. To name a few, Gaussian elimination, iterative methods based on

Gauss-Siedel approximation, Jacobian elimination, and matrix recursive methods. In the solution of such sets of homogeneous linear equations, the analyst is primarily concerned with efficiency of calculations and rapidity of convergence and estimation of the degree of approximation, if appropriate. Clearly, because of the number of states, in any realistic model of a production line, there is a need for an efficient algorithm to determine the steady-state probabilities associated with the states of the system.

The algorithm outlined below is based on Gaussian elimination with a dynamically adjusted successive over-relaxation factor to achieve rapid convergence. The essential components of this algorithm are

- Ordering of the states
- Generation of the transition matrix
- Solution of the resulting system of linear equations

This algorithm was coded in C++ by Dr. Cathal Heavey and is based on the work of Heavey, Papadopoulos and Browne (1993), Papadopoulos, Heavey and O'Kelly (1989, 1990), Papadopoulos and O'Kelly (1989) and Papadopoulos (1989) and with appropriate instructions is available at the website associated with this book with the abbreviated name MARKOV. The user inputs into the algorithm the following parameters: K the number of stations; $B_2, \ldots, B_K$ the buffer capacities; $P_1, \ldots, P_K$ the number of phases of the service distribution for each station; $R_1, \ldots, R_K$, the number of phases of the repair distribution for each station; μ_i the mean service rates; r_i the mean repair rates; and β_i the mean breakdown rates. Thus, a very general algorithm has been developed which generates the transition probability matrix, A, and then solves the set of linear equations via the use of the SOR method and gives as output the throughput, X_K, of a K-station production line with finite intermediate buffers and with the service and repair times following a phase-type distribution and the times to failure being exponentially distributed.

The ordering of the states affects the structure of the conservative matrix A. The objective is to find an ordering of the states such that matrix A will have as simple a structure as possible from a computational point of view. This will facilitate the development of a very efficient algorithm for the generation of matrix A. To select an appropriate ordering of the states, a criterion for the structure of matrix A must be selected. In the algortithm included at the website associated with this book, the criterion used was to keep the non-zero elements of the conservative matrix A as close as possible to the diagonal elements, i.e., a quasi band diagonal matrix. Because of the increasing number of states, as system complexity increases, it is not possible to assess how close matrix A is to a strict band diagonal matrix.

A recursive algorithm for generating the conservative matrix A has been developed based on the generation of a series of sub-matrices (Heavey, Papadopoulos and Browne, 1993). Specific details of the matrix generation process for the case of a reliable exponential production line with inter-station buffers (Papadopoulos, Heavey and O'Kelly, 1989) are given in sub-section 2.1.2.

Table 2.3. Exponential service, repair, and failure, $K = 3$

$B_2 = 2, B_3 = 4$ $\mu_1 = 1.5, \mu_2 = 2.0, \mu_3 = 1.9$ $r_1 = 0.1, r_2 = 0.02, r_3 = 0.15$ $\beta_1 = 0.02, \beta_2 = 0.01, \beta_3 = 0.09$	
Analytical Results	Simulation Results 95% CI
$X_3 = 0.7346$	0.721 – 0.737 – 0.752
$B_2 = 5, B_3 = 3$ $\mu_1 = 2.6, \mu_2 = 3.0, \mu_3 = 3.2$ $r_1 = 0.5, r_2 = 0.03, r_3 = 0.15$ $\beta_1 = 0.03, \beta_2 = 0.01, \beta_3 = 0.02$	
Analytical Results	Simulation Results 95% CI
$X_3 = 1.2985$	1.26 – 1.28 – 1.31

An iterative method was used to solve the system of linear equations. The iterative method used was the Successive Over Relaxation (SOR) method. SOR is more efficient than the Gauss-Seidel method, but SOR has one main drawback, the unknown optimal value of the relaxation factor. A process of dynamically adjusting the relaxation factor has been introduced into the algorithm, which worked well in practice.

The results of the algorithm have been compared with available analytical results (systems with a small number of states) and simulation studies on systems with relatively large number of states and has been found to be satisfactory.

A sample of the throughput rate, X_K, from the analytical model, compared with results from a simulation model are given below. Two arbitrary examples are given for systems with: (1) Exponential service, repair, and failure (see Table 2.3); (2) Erlang service, exponential repair, and failure (see Table 2.4); (3) Erlang service, repair, and exponential failure (see Table 2.5). $\mu_i, r_i, \beta_i, i = 1, 2, 3$ are the mean service rates, repair rates, and failure rates, respectively.

As can be seen from Tables 2.3, 2.4 and 2.5, the point estimates of the throughput from the simulation model are very close to the results from the analytical model and all the analytical results are covered by the 95% confidence intervals (CI). These results are typical of all the models tested against simulation. Therefore, it can be safely concluded that the analytical model yields the correct results.

In sub-section 2.1.2, the detailed development of the conservative matrix A for the reliable exponential case is developed. Details of more general cases (phase-type distribution of service and repair times) are available in the literature listed at the end of this chapter.

Table 2.4. Erlang service, exponential repair, and failure, $K = 4$

$P_1 = 3, P_2 = 2, P_3 = 3, P_4 = 2$ $B_2 = 3, B_3 = 2, B_4 = 3$ $\mu_1 = 5.0, \mu_2 = 4.5, \mu_3 = 5.2, \mu_4 = 3.7$ $r_1 = 0.05, r_2 = 0.03, r_3 = 0.07, r_4 = 0.1$ $\beta_1 = 0.02, \beta_2 = 0.001, \beta_3 = 0.003, \beta_4 = 0.05$	
Analytical Results	Simulation Results 95% CI
$X_4 = 2.0025$	1.99 – 2.04 – 2.10

$P_1 = 2, P_2 = 3, P_3 = 3, P_4 = 2$ $B_2 = 3, B_3 = 5, B_4 = 3$ $\mu_1 = 1.5, \mu_2 = 0.9, \mu_3 = 0.9, \mu_4 = 1.5$ $r_1 = 0.1, r_2 = 0.03, r_3 = 0.2, r_4 = 0.3$ $\beta_1 = 0.02, \beta_2 = 0.01, \beta_3 = 0.09, \beta_4 = 0.2$	
Analytical Results	Simulation Results 95% CI
$X_4 = 0.4591$	0.450 – 0.456 – 0.462

Table 2.5. Erlang service, repair, and exponential failure, $K = 4$

$P_1 = 3, P_2 = 2, P_3 = 3, P_4 = 2$ $R_1 = 2, R_2 = 3, R_3 = 4, R_4 = 2$ $B_2 = 2, B_3 = 1, B_4 = 3$ $\mu_1 = 2.5, \mu_2 = 1.9, \mu_3 = 2.6, \mu_4 = 3.0$ $r_1 = 0.05, r_2 = 0.03, r_3 = 0.07, r_4 = 0.1$ $\beta_1 = 0.02, \beta_2 = 0.001, \beta_3 = 0.003, \beta_4 = 0.05$	
Analytical Results	Simulation Results 95% CI
$X_4 = 1.1325$	1.10 – 1.12 – 1.14

$P_1 = 2, P_2 = 3, P_3 = 3, P_4 = 2$ $R_1 = 3, R_2 = 2, R_3 = 3, R_4 = 2$ $B_2 = 1, B_3 = 2, B_4 = 3$ $\mu_1 = 5.0, \mu_2 = 4.5, \mu_3 = 5.2, \mu_4 = 3.7$ $r_1 = 0.1, r_2 = 0.03, r_3 = 0.2, r_4 = 0.3$ $\beta_1 = 0.02, \beta_2 = 0.01, \beta_3 = 0.09, \beta_4 = 0.2$	
Analytical Results	Simulation Results 95% CI
$X_4 = 1.5958$	1.53 – 1.54 – 1.60

Table 2.6. Notation

Symbol	*Meaning*
K	Number of stations.
B_i	Buffer capacity preceding the i^{th} station. Note: when $B_i = B_j$ for all i, then the buffer capacity is denoted by B.
n_i	Status of buffer i.
s_i	Status of station i (see Table 2.7).
m_K^B	Number of states in the sub-network of a K station system with identical buffers, each of capacity B.
$m_K^{B_2,\ldots,B_K}$	Number of states in the sub-network of a K station system with non-identical buffers, with buffer capacities $B_2, \ldots, B_K$.

Table 2.7. States of station i

s_i	*Meaning*
0	Station is idle.
1	Station is busy.
2	Station is busy and blocking preceding station.

2.1.2 The algorithm for the generation of the conservative matrix A for the reliable exponential production lines with inter-station buffers

For purposes of illustration, in this sub-section, the recursive algorithm will be applied to the case of a reliable exponential production line only (see Papadopoulos, Heavey and O'Kelly, 1989). Table 2.6 lists the notation used in this sub-section.

The algorithm for generating the conservative matrix A is divided into two parts. The first part generates sub-matrix $\mathrm{Y1}^*_{K=k}$ for the appropriate system. The second part generates sub-matrix $\mathrm{Y2}^*_{K=k}$ from sub-matrix $\mathrm{Y1}^*_{K=k}$, and the non-diagonal elements $P_1\mu_1, R_1 r_1$ and β_1. In the first part of the algorithm, the non-zero elements of $\mathrm{Y1}^*_{K=k}$, the column coefficients, and the number of elements in each row are stored in separate one dimensional arrays.

The second part of the algorithm is executed during the execution of the solution procedure. As a consequence, the non-zero elements of sub-matrix $\mathrm{Y2}^*_{K=k}$, and the non-diagonal elements $P_1\mu_1, R_1 r_1$ and β_1 need not be stored in memory. This can greatly reduce the amount of memory required to solve a system.

The states of the sub-network are described by the following vector:

$$(n_2, s_2, n_3, s_3, \ldots, n_K, s_K) \tag{2.18}$$

s_1 is not included in the state vector because it is always equal to 1, i.e., the first station is never idle. This does not mean that the first station cannot be blocked by buffer B_2 or work-station WS_2. s_i can take any of the values listed in Table 2.7, and n_i can take any value from 0 to B_i, as it denotes the number of items in buffer i.

The set of linear equations for the solution of P_2, the marginal p.d.f. for the sub-network, can be written in the following two ways.

$$P_2 A = 0 \tag{2.19}$$

$$A^T P_2 = 0. \tag{2.20}$$

In the rest of this sub-section, A^T is examined. This is because in order to generate matrix A efficiently, the relationship between its columns (rows of A^T) needs to be examined. In order to simplify the notation, A denotes A^T in the rest of this sub-section.

Number of States

A prerequisite to the development of the algorithm is the derivation of an equation to calculate the number of states in the sub-network. The case where buffers are identical is investigated first and then the case of buffers being non-identical.

Identical Buffers

For this case, where buffers are of equal capacity, say $B = N$, the following difference equation is obtained, in a way analogous to that used for the case where buffers were not allowed (see Papadopoulos, 1989 and Papadopoulos and O'Kelly, 1989):

$$m^N_{K+2} - (N+3)\, m^N_{K+1} + m^N_K = 0. \tag{2.21}$$

Then, its *characteristic equation* is

$$x^2 - (N+3)x + 1 = 0,$$

with two real roots:

$$x_1 = \frac{(N+3) + \sqrt{(N+3)^2 - 4}}{2}, \quad x_2 = \frac{(N+3) - \sqrt{(N+3)^2 - 4}}{2}.$$

Therefore the general solution of (2.21) is

$$\begin{aligned} m^N_K &= c_1 x_1^K + c_2 x_2^K \\ &= c_1 \left(\frac{(N+3) + \sqrt{(N+3)^2 - 4}}{2} \right)^K + c_2 \left(\frac{(N+3) - \sqrt{(N+3)^2 - 4}}{2} \right)^K. \end{aligned}$$

The initial conditions: $m_0 = 0$ and $m_1 = 1$ give

$$c_1 + c_2 = 0,$$

and

$$c_1 \left(\frac{(N+3) + \sqrt{(N+3)^2 - 4}}{2} \right) + c_2 \left(\frac{(N+3) - \sqrt{(N+3)^2 - 4}}{2} \right) = 1.$$

Hence,

$$c_1 = -c_2 = \frac{1}{\sqrt{(N+3)^2-4}} = \frac{\sqrt{(N+3)^2-4}}{(N+3)^2-4},$$

and the general solution becomes

$$m_K^N = \left\{ \left(\frac{(N+3)+\sqrt{(N+3)^2-4}}{2} \right)^K - \left(\frac{(N+3)-\sqrt{(N+3)^2-4}}{2} \right)^K \right\} \left(\frac{1}{\sqrt{(N+3)^2-4}} \right). \tag{2.22}$$

Equation (2.22) was used to calculate the number of states for the systems in Table 2.8. It is clear from Table 2.8 that the number of states increases tremendously with an increase in the size of the buffer and in the number of stations. This places strict limits on the size of the system for which exact results can be obtained.

Non-identical Buffers

For this case, where buffers are of unequal capacity, say $B_2, B_3, \ldots, B_K$, the difference equation may be shown to be similar to that obtained for Case 1, where buffers were of equal capacity (equation (2.21)), i.e.,

$$m_{K+2}^{B_2,B_3,\ldots,B_{K+2}} = (B_{K+2}+3)\, m_{K+1}^{B_2,B_3,\ldots,B_{K+1}} - m_K^{B_2,B_3,\ldots,B_K}. \tag{2.23}$$

Applying the initial conditions $m_0 = 0$ and $m_1 = 1$ to equation (2.23), for $K = 0, 1, \ldots$, sequentially,

(1) $K = 0$**:**

$$\begin{aligned} m_2^{B_2} &= (B_2+3)\, m_1 - m_0 \\ &= (B_2+3)\,(1) - 0 \\ &= B_2 + 3. \end{aligned} \tag{2.24}$$

(2) $K = 1$**:** Combination of equations (2.23) and (2.24) gives

$$\begin{aligned} m_3^{B_2,B_3} &= (B_3+3)\, m_2^{B_2} - m_1 \\ &= (B_2+3)\,(B_3+3) - 1. \end{aligned} \tag{2.25}$$

(3) $K = 2$**:** Combination of equations (2.23), (2.24) and (2.25) gives

$$\begin{aligned} m_4^{B_2,B_3,B_4} &= (B_4+3)\, m_3^{B_2,B_3} - m_2^{B_2} \\ &= (B_4+3)\,[(B_3+3)\,(B_2+3) - 1] - (B_2+3) \\ &= (B_2+3)\,[(B_3+3)\,(B_4+3) - 1] - (B_4+3). \end{aligned} \tag{2.26}$$

Table 2.8. Number of states of the system

# *of stations*	*buffer size*										
	0	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1	1
2	3	4	5	6	7	8	9	10	11	12	13
3	8	15	24	35	48	63	80	99	120	143	168
4	21	56	115	204	329	496	711	980	1,309	1,704	2,171
5	55	209	551	1,189	2,255	3,905	6,319	9,701	14,279	20,305	28,055
6	144	780	2,640	6,930	15,456	30,744	56,160	96,030	155,760	241,956	362,544
7	377	2,911	12,649	40,391	105,937	242,047	449,121	950,559			
8	987	10,864	60,604	235,416	726,103						
9	2,584	40,545	290,376								
10	6,765	151,316									
11	17,711	564,719									
12	46,368										
13	121,393										
14	317,811										

(4) $K = 3$**:** Combination of equations (2.23), (2.25) and (2.26) gives

$$\begin{aligned} m_5^{B_2,B_3,B_4,B_5} &= (B_5+3)\, m_4^{B_2,B_3,B_4} - m_3^{B_2,B_3} \\ &= (B_5+3)\left\{(B_4+3)\left[(B_3+3)(B_2+3)-1\right]-(B_2+3)\right\} \\ &\quad -\left[(B_3+3)(B_2+3)-1\right] \\ &= \left[(B_2+3)(B_3+3)-1\right]\left[(B_4+3)(B_5+3)-1\right] \\ &\quad -(B_2+3)(B_5+3). \end{aligned} \tag{2.27}$$

(5) $K = 4$**:** Combination of equations (2.23), (2.26) and (2.27) gives

$$\begin{aligned} m_6^{B_2,\ldots,B_6} &= (B_6+3)\, m_5^{B_2,\ldots,B_5} - m_4^{B_2,\ldots,B_4} \\ &= (B_6+3)\left\{\left[(B_2+3)(B_3+3)-1\right]\left[(B_2+3)(B_3+3)-1\right]\right. \\ &\quad \left.-(B_2+3)(B_5+3)\right\} \\ &\quad -\left\{(B_2+3)\left[(B_3+3)(B_4+3)-1\right]-(B_5+3)\right\}, \end{aligned}$$

and after some algebra,

$$\begin{aligned} m_6^{B_2,\ldots,B_6} &= (B_4+3)\left[(B_2+3)(B_3+3)-1\right]\left[(B_5+3)(B_6+3)-1\right] \\ &\quad -(B_6+3)\left[(B_2+3)(B_3+3)-1\right] \\ &\quad -(B_2+3)\left[(B_5+3)(B_6+3)-1\right]. \end{aligned} \tag{2.28}$$

The examples illustrated above suggest the following iterative scheme to calculate the number of states of a system with non-identical buffers, i.e., a system with K stations and buffer capacities $B_2, B_3 \ldots, B_K$.

Initial Values:

$V1 = 1 = m_1$
$V2 = 0 = m_0$
DO $J = 2$ to K
$\quad V = (B_J+3)V1 - V2$
$\quad V2 = V1$
$\quad V1 = V$
END DO J
$m_K^{B_2,\ldots,B_K} = V.$

The iterative scheme above calculates the number of states of a K station system with buffer capacities $B_2, B_3, \ldots, B_K$, by first calculating $m_2^{B_2}$ and then $m_3^{B_2,B_3}$, i.e., by using B_i in the following order, $i = 2, 3, \ldots, K-1, K$. It is interesting to note that the number of states for a system with non-identical buffers can also be calculated using B_i in the reverse order, $i = K, K-1, \ldots, 2$, i.e., calculate $m_2^{B_K}$ first, then $m_3^{B_{K-1},B_K}$ and so on. In the algorithm for the generation of the transition matrix, the latter method is used.

Ordering of States

Each state is represented by the following vector:

$$(n_2, s_2, n_3, \ldots n_K, s_K). \tag{2.29}$$

Each state is altered by the following rule:
If s_i equals 2 and $i > 2$ then

$$s_{i-1}^{altered} = (s_{i-1} - 1).$$

Then the 'altered states' are given a unique numerical value in order to ensure a $1-1$ correspondence, as follows:

$$n_2 \times L^{E-1} + s_2^{altered} \times L^{E-2} + \cdots + n_K \times L^{E-(E-1)} + s_K^{altered} \times L^{E-E} = \text{numerical value}$$

with E equal to the number of elements in the state vector and L given an appropriate integer value as follows:

$$L > \text{MAX}\{B_j, 2\}, \quad j = 2, \ldots, K.$$

L is the base for the numerical values of the states. The numerical values of the 'altered states' are then ordered in increasing value and the states ordered according to this.

The above procedure will be illustrated with an example. Table 2.9 lists the states, the 'altered states', and the numerical values of the 'altered states' for $K = 3$, $B_2 = 0, B_3 = 1$. E equals 4 and L equals 3.

Only states (0,1,1,2) and (0,2,1,2) were altered (see Table 2.9). Table 2.10 gives the numerical values of the 'altered states' ordered in increasing value and the states ordered according to this ordering.

The reason for ordering the states is to give matrix A a relatively simple structure which can be exploited when developing the algorithm to generate matrix A. Matrix

Table 2.9. Altered states and their numerical values

States	Altered States	Numerical Value
(0,0,0,0)	(0,0,0,0)	0
(0,0,0,1)	(0,0,0,1)	1
(0,1,0,0)	(0,1,0,0)	9
(0,1,0,1)	(0,1,0,1)	10
(0,2,0,0)	(0,2,0,0)	18
(0,2,0,1)	(0,2,0,1)	19
(0,0,1,1)	(0,0,1,1)	4
(0,1,1,1)	(0,1,1,1)	13
(0,1,1,2)	(0,0,1,2)	5
(0,2,1,1)	(0,2,1,1)	22
(0,2,1,2)	(0,1,1,2)	14

Table 2.10. Ordering of states

Ordered States	Numerical Value
(0,0,0,0)	0
(0,0,0,1)	1
(0,0,1,1)	4
(0,1,1,2)	5
(0,1,0,0)	9
(0,1,0,1)	10
(0,1,1,1)	13
(0,2,1,2)	14
(0,2,0,0)	18
(0,2,0,1)	19
(0,2,1,1)	22

(2.30) gives matrix A for $K = 3, B_2 = 0, B_3 = 1$, with the states ordered according to Table 2.10. Note that each of the non-diagonal elements, μ_3, μ_2 and μ_1, in matrix (2.30) are always found in the same position relative to the diagonal element, i.e., μ_2 is always two columns to the right of the diagonal element.

$$A = \begin{vmatrix}
-\mu_1 & \mu_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\mu_1-\mu_3 & \mu_3 & 0 & \mu_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -\mu_1-\mu_3 & \mu_3 & 0 & \mu_2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\mu_1-\mu_3 & 0 & 0 & \mu_2 & 0 & 0 & 0 & 0 \\
\mu_1 & 0 & 0 & 0 & -\mu_1-\mu_2 & \mu_3 & 0 & 0 & 0 & 0 & 0 \\
0 & \mu_1 & 0 & 0 & 0 & -\sum_{i=1}^{3}\mu_i & \mu_3 & 0 & \mu_2 & 0 & 0 \\
0 & 0 & \mu_1 & 0 & 0 & 0 & -\sum_{i=1}^{3}\mu_i & \mu_3 & 0 & \mu_2 & 0 \\
0 & 0 & 0 & \mu_1 & 0 & 0 & 0 & -\mu_3 & 0 & 0 & \mu_2 \\
0 & 0 & 0 & 0 & \mu_1 & 0 & 0 & 0 & -\mu_2 & \mu_3 & 0 \\
0 & 0 & 0 & 0 & 0 & \mu_1 & 0 & 0 & 0 & -\mu_2-\mu_3 & \mu_3 \\
0 & 0 & 0 & 0 & 0 & 0 & \mu_1 & 0 & 0 & 0 & -\mu_2-\mu_3
\end{vmatrix} \tag{2.30}$$

Structure of Matrix A

Matrix A equals the summation of sub-matrices A_1, A_2 and A_0. Sub-matrices A_0 and A_2 have very simple structures whereas sub-matrix A_1 has a relatively complicated structure. Sub-matrix A_1 is examined first and then A_0 and A_2.

Description of A_1

Matrix A_1 for any value K ($K > 2$) with identical or non-identical buffers was found to take the form described in Figure 2.3.

C, D, and D^* are all

$$m_{K-1}^{B_3,B_4,\ldots,B_K} \times m_{K-1}^{B_3,B_4,\ldots,B_K}$$

matrices. E and F are

$$\left(m_{K-1}^{B_3,B_4,\ldots,B_K} - m_{K-2}^{B_4,\ldots,B_K}\right) \times \left(m_{K-1}^{B_3,B_4,\ldots,B_K} - m_{K-2}^{B_4,\ldots,B_K}\right)$$

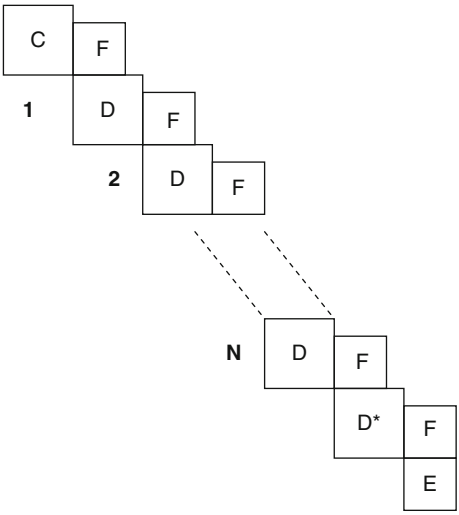


Fig. 2.3. Structure of A_1, $K > 2, B_2 = N, B_3, B_4, \ldots, B_K$

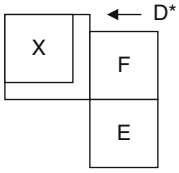


Fig. 2.4. Relationship of sub-matrix E to D^*

matrices. The number of times sub-matrices D and F appear between sub-matrices C and D^* equals $B_2 = N$. The relationships between the sub-matrices are as follows:

1. Sub-matrix C for a K station system with $B_2 = N, B_3, B_4, \ldots, B_K$ is generated from A_1 for $K-1$ station system with $B_3, B_4, \ldots, B_K$, by: (i) Substituting μ_{i+1} for μ_i, $i = K, K-1, \ldots 2$ (i.e., backwards) in $(A_1)_{K-1}$; (ii) Subtracting μ_1 from the last $m_{K-2}^{B_4,\ldots,B_K}$ diagonal elements of $(A_1)_{K-1}$.

2. (i) D is generated from C by subtracting μ_2 from the first $\left(m_{K-1}^{B_3,B_4,\ldots,B_K} - m_{K-2}^{B_4,\ldots,B_K}\right)$ diagonal elements of C.
   (ii) If $B_2 = 0$, then there is no sub-matrix D. Therefore D^* is generated from C. If $B_2 = 0$, then μ_1 is also added to the last $m_{K-2}^{B_4,\ldots,B_K}$ diagonal elements of C.

3. D^* is generated from D by adding μ_1 to the last $m_{K-2}^{B_4,\ldots,B_K}$ diagonal elements of D. If $B_2 = 0$, this relationship does not hold because there will be no sub-matrix D.
4. E is a $\left(m_{K-1}^{B_3,B_4,\ldots,B_K} - m_{K-2}^{B_4,\ldots,B_K}\right) \times \left(m_{K-1}^{B_3,B_4,\ldots,B_K} - m_{K-2}^{B_4,\ldots,B_K}\right)$ matrix which is generated from X, a sub-matrix of D^* (see Figure 2.4), by adding μ_1 to all the diagonal elements of X.

5. F is a square diagonal matrix of order $\left(m_{K-1}^{B_3,B_4,\ldots,B_K} - m_{K-2}^{B_4,\ldots,B_K}\right)$ with μ_2 in the diagonal elements. The first sub-matrix F is positioned on the $\left(m_{K-2}^{B_4,\ldots,B_K}+1\right)$ row and the $\left(m_{K-1}^{B_3,B_4,\ldots,B_K}+1\right)$ column of matrix A_1. Its position relative to D and D^* is the same as its position relative to C.

Once A_1 for $K=2, B_K$ is obtained, using the relationships outlined above, A_1 for any value $K, B_2 = N, B_3, B_4, \ldots, B_K$ can be generated. A_1 for $K=2$ and any value B_K is easy to generate.

Description of A_0 and A_2

In general A_0 is a $\left(m_K^{B_2,B_3,\ldots,B_K} \times m_K^{B_2,B_3,\ldots,B_K}\right)$ matrix with λ in all the diagonal elements and $\mu_K\theta$ in exactly the same positions as $\mu_K\theta'$ is in A_1.

In general A_2 is a $\left(m_K^{B_2,B_3,\ldots,B_K} \times m_K^{B_2,B_3,\ldots,B_K}\right)$ matrix with μ_1 in the I^{th} column and the $\left(m_{K-1}^B + I\right)$ row with $I = 1,2,\ldots,\left(m_K^B - m_{K-1}^B\right)$.

Therefore the basic structure of $A = A_0 + A_1 + A_2$ is given by the structure of sub-matrix A_1 except:

1. A does not contain any λ, i.e., λ in the diagonal elements of A_0 cancels $-\lambda$ in the diagonal elements of A_1.
2. Instead of $\mu_K\theta'$ in A_1, there is a μ_K in A, i.e., $(\theta + \theta') = 1$. This is because $\mu_K\theta$ in A_0 is in exactly the same position as $\mu_K\theta'$ is in A_1.
3. The inclusion of the sub-matrix A_2.

Figure 2.5 gives the structure of sub-matrix A for a K station system ($K > 2$). Sub-matrices C, D, D^*, and E are as described in sub-section 2.1.2 except the changes outlined above, i.e., their diagonal elements do not contain any λ and $\mu_K\theta' \rightarrow \mu_K$. Sub-matrices G and H contain the μ_1 elements of A_2. G is a square matrix of order $m_{K-1}^{B_3,B_4,\ldots,B_K}$ with μ_1 in the diagonal elements. H is a square matrix of order $\left(m_{K-1}^{B_3,B_4,\ldots,B_K} - m_{K-2}^{B_4,\ldots,B_K}\right)$ with μ_1 in the diagonal elements.

Algorithm to Generate Matrix A

The following is a description of the algorithm to generate A, which was coded in C++. The user inputs K the number of stations, $B_2, B_3, \ldots, B_K$ the buffer capacities, the mean service rates $\mu_1, \mu_2, \ldots \mu_K$, and θ the feedback probability.

RULE 1.
- **(i)** The first element (row=1,column=1) is equal to $-\mu_1$ and element (row=1,column=2) equals μ_K.
- **(ii)** This part generates the next $(B_K + 2)$ rows. The next $(B_K + 2)$ diagonal elements (i,i) are put equal to $-\mu_1 - \mu_K$ and the value μ_K is placed in element $(i, i+1)$, for the $(B_K + 2)$ rows, except the last row.

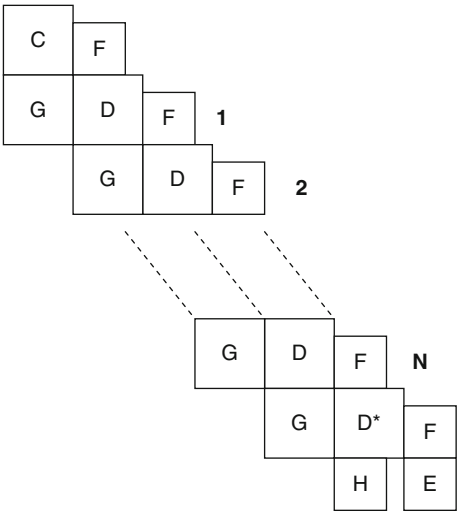


Fig. 2.5. Structure of A, $K > 2, B_2 = N, B_3, B_4, \ldots, B_K$

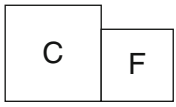


Fig. 2.6. Illustration of Rule 2

(iii) If K=2, then μ_1 is added to the last diagonal element created above. This is matrix A for $K = 2$, go to Rule 6.

Rule 1 will create A if $K = 2$. If $K > 2$ it will create sub-matrix C for $K = 3$. Rules 2, 3, 4, and 5, below, are all contained within a loop (see 'DO $T = 3$ to K' below). 'END DO T' denotes the end of the loop. If $K = 3$, the first iteration of the loop will create A for $K = 3, B_{K-1}, B_K$, if not, then sub-matrix C for $K = 4, B_{K-2}, B_{K-1}, B_K$ is created and so on until A for $K = K$ is created.

DO $T = 3$ to K

$X = (T - 2)$

$Y = (X + 1)$

$W = (K - X)$

RULE 2. Place the top left element of a square matrix of order $\left(m_Y^{B_{W+1},\ldots,B_K} - m_{Y-1}^{B_{W+2},\ldots,B_K}\right)$ with μ_W in its diagonal elements, in the $\left(m_{Y-1}^{B_{W+2},\ldots,B_K} + 1\right)$ row and the $\left(m_Y^{B_{W+1},\ldots,B_K} + 1\right)$ column of A. This is sub-matrix F and its position relative to C is illustrated in Figure 2.6.

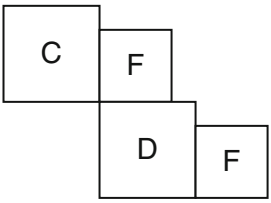


Fig. 2.7. Illustration of Rule 3

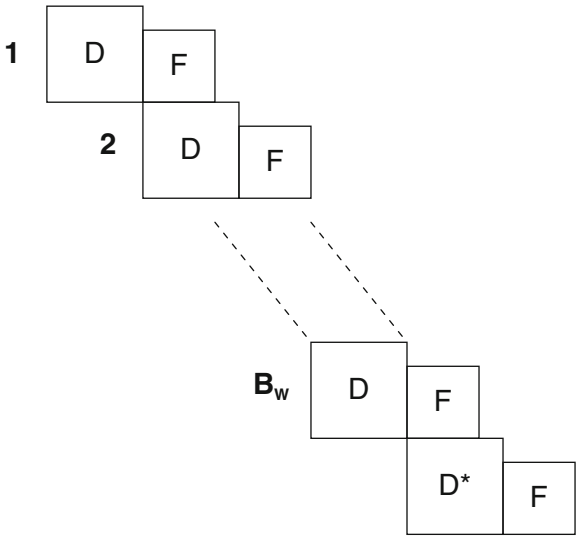


Fig. 2.8. Illustration of Rule 4

RULE 3. C is a square matrix of order $m_Y^{B_{W+1},\ldots,B_K}$. D is generated from C by subtracting μ_W from the first $\left(m_Y^{B_{W+1},\ldots,B_K} - m_{Y-1}^{B_{W+2},\ldots,B_K}\right)$ diagonal elements of C. D is positioned as in Figure 2.7. Also, F is copied onto F (see Figure 2.7). If $B = 0$ and $T = K$ then μ_1 is also added to the last $m_{Y-1}^{B_{W+2},\ldots,B_K}$ diagonal elements of C, i.e., C will be copied onto D^*.

Rule 4 is contained within a loop ('DO $Z = 1$ to B_W'), and it is executed B_W times. If $B_W = 0$, this rule is not used.

RULE 4. DO $Z = 1$ to B_W
Copy Sub-matrices D and F as described in Figure 2.8.
When $T = K$ and $Z = B_W$, μ_1 is added to the last $m_{Y-1}^{B_{W+2},\ldots,B_K}$ diagonal elements of D, i.e., sub-matrix D is copied onto D^*.
END DO Z

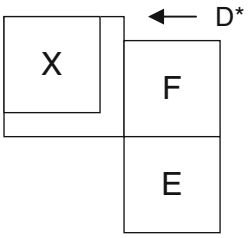


Fig. 2.9. Illustration of Rule 5

RULE 5. The top left of the square sub-matrix D^* of dimension $\left(m_Y^{B_{W+1},\ldots,B_K} - m_{Y-1}^{B_{W+2},\ldots,B_K}\right)$ is copied
on to E, see Figure 2.9 i.e., X is copied onto E.
The position of E is also illustrated in Figure 2.9.
If $T = K$, μ_1 is added to all the diagonal elements of E.
END DO T

Rule 6 below generates the non-zero elements of A_2 and is executed after exiting 'DO $T = 3$ to K'.

RULE 6. DO $I = 1$ to $\left(m_K^{B_2,\ldots,B_K} - m_{K-1}^{B_3,\ldots,B_K}\right)$
Place μ_1 in row $\left(m_{K-1}^{B_3,\ldots,B_K} + I\right)$ and column I.
END DO I

Application of the Algorithm

Here, the explicit derivation of the conservative matrix A for $K = 3, B_2 = 1, B_3 = 0$, with exponentially distributed processing times with mean values $\frac{1}{\mu_i}$, $i = 1,2,3$, is developed by applying the algorithm described above.

Rule 1

Applying Rule 1, matrix (2.31) is obtained. Since $K > 2$, this is matrix C for $K = 3$.

$$C = \begin{vmatrix} -\lambda - \mu_1 & \mu_3 & 0 \\ 0 & -\lambda - \mu_1 - \mu_3 & \mu_3 \\ 0 & 0 & -\lambda - \mu_1 - \mu_3 \end{vmatrix}. \tag{2.31}$$

Rule 1(i) generated the first row of C. Rule 1(ii) generated the next $2 = (B_3 + 2)$ row(s): note that in the last row generated by Rule 1(ii), μ_3 is not placed in the column next to the diagonal.

Rules 2, 3, 4, 5 are all contained within a loop which is executed $(K - 3 + 1)$ times. Therefore, for the example illustrated here, only one iteration of the loop is performed, with $T = 3, X = 1, Y = 2$ and $W = 2$.

Rule 2

Rule 2 will generate a square matrix (sub-matrix F) of order $2 = \left(m_2^0 - m_1\right) = \left(m_Y^{B_{W+1},\ldots,B_K} - m_{Y-1}^{B_{W+2},\ldots,B_K}\right)$ with $\mu_2 = \mu_W$ in the diagonal elements. The top left element of F is positioned in the $2^{nd} = (m_1 + 1) = (m_{Y-1}^{B_{W+2},\ldots,B_K} + 1)$ row and the $4^{th} = (m_2^0 + 1) = (m_Y^{B_{W+1},\ldots,B_K} + 1)$ column of A. The first 3 rows of matrix A are given in matrix (2.32).

$$\begin{vmatrix} -\mu_1 & \mu_3 & 0 & 0 & 0 & 0\,0\,0\,0\,0 \\ 0 & -\mu_1-\mu_3 & \mu_3 & \mu_2 & 0 & 0\,0\,0\,0\,0 \\ 0 & 0 & -\mu_1-\mu_3 & 0 & \mu_2 & 0\,0\,0\,0\,0 \end{vmatrix}. \tag{2.32}$$

Rule 3

Rule 3 generates sub-matrix D from sub-matrix C. D is generated from C by subtracting $\mu_2 = \mu_W$ from the first $2 = (m_2^0 - m_1) = (m_Y^{B_{W+1},\ldots,B_K} - m_{Y-1}^{B_{W+2},\ldots,B_K})$ diagonal elements of C. Rule 3 also copies sub-matrix F. The positions of sub-matrices D and F are illustrated in Figure 2.7. Excluding the non-diagonal μ_1 elements that are generated by Rule 6, matrix (2.33) gives the first 6 rows of A for $K = 3, B_2 = 1, B_3 = 0$

$$\begin{vmatrix} -\mu_1 & \mu_3 & 0 & 0 & 0 & 0 & 0\,0\,0\,0 \\ 0 & -\mu_1-\mu_3 & \mu_3 & \mu_2 & 0 & 0 & 0\,0\,0\,0 \\ 0 & 0 & -\mu_1-\mu_3 & 0 & \mu_2 & 0 & 0\,0\,0\,0 \\ 0 & 0 & 0 & -\mu_1-\mu_2 & \mu_3 & 0 & 0\,0\,0\,0 \\ 0 & 0 & 0 & 0 & -\mu_1-\mu_2-\mu_3 & \mu_3 & \mu_2\,0\,0\,0 \\ 0 & 0 & 0 & 0 & 0 & -\mu_1-\mu_3 & 0\,\mu_2\,0\,0\,0 \end{vmatrix} \tag{2.33}$$

Rule 4

Rule 4 copies sub-matrix D $B_W = 1$ times. Because $T = K$ and $Z = B_W$, μ_1 is added to the last $m_1 = 1$ diagonal elements of D, i.e., sub-matrix D is copied onto D^* (see Figure 2.8). Matrix (2.34) gives the first nine rows of A (excluding the non-diagonal elements generated by Rule 6), which have been generated by rules 1, 2, 3 and 4.

$$\begin{vmatrix} -\mu_1 & \mu_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0\,0\,0 \\ 0 & -\mu_1-\mu_3 & \mu_3 & \mu_2 & 0 & 0 & 0 & 0 & 0\,0\,0 \\ 0 & 0 & -\mu_1-\mu_3 & 0 & \mu_2 & 0 & 0 & 0 & 0\,0\,0 \\ 0 & 0 & 0 & -\mu_1-\mu_2 & \mu_3 & 0 & 0 & 0 & 0\,0\,0 \\ 0 & 0 & 0 & 0 & -\sum_{i=1}^{3}\mu_i & \mu_3 & \mu_2 & 0 & 0\,0\,0 \\ 0 & 0 & 0 & 0 & 0 & -\mu_1-\mu_3 & 0 & \mu_2 & 0\,0\,0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\mu_1-\mu_2 & \mu_3 & 0\,0\,0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\sum_{i=1}^{3}\mu_i & \mu_3\,\mu_2\,0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu_3\,0\,\mu_2 \end{vmatrix} \tag{2.34}$$

Rule 5

E is a square matrix of order $2 = (m_2^0 - m_1) = (m_Y^{B_{W+1},\ldots,B_K} - m_{Y-1}^{B_{W+2},\ldots,B_K})$ equal to the top left of sub-matrix D^* of the said dimension (see Figure 2.9). Since $T = K$, μ_1 is added to all the diagonal elements of E. The position of E is illustrated in Figure 2.9, the top left element of E is positioned on the 10^{th} row and the 10^{th} column of A (see matrix (2.35)).

Rule 6

The following loop generates the elements of A_2 (the non-diagonal μ_1 elements of A), $8 = (m_3^{1,0} - m_2^0) = (m_K^{B_2,\ldots,B_K} - m_{K-1}^{B_3,\ldots,B_K})$ and $3 = m_2^0 = m_{K-1}^{B_3,\ldots,B_K}$.

DO $I = 1$ to 8
    Place μ_1 in row $(3+I)$ and column I.
END DO I

The required matrix A for $K = 3, B_2 = 1, B_3 = 0$ is given in matrix (2.35).

$$A = \begin{vmatrix}
-\mu_1 & \mu_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\mu_1-\mu_3 & \mu_3 & \mu_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -\mu_1-\mu_3 & 0 & \mu_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mu_1 & 0 & 0 & -\mu_1-\mu_2 & \mu_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mu_1 & 0 & 0 & -\sum_{i=1}^{3}\mu_i & \mu_3 & \mu_2 & 0 & 0 & 0 & 0 \\
0 & 0 & \mu_1 & 0 & 0 & -\mu_1-\mu_3 & 0 & \mu_2 & 0 & 0 & 0 \\
0 & 0 & 0 & \mu_1 & 0 & 0 & -\mu_1-\mu_2 & \mu_3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mu_1 & 0 & 0 & -\sum_{i=1}^{3}\mu_i & \mu_3 & \mu_2 & 0 \\
0 & 0 & 0 & 0 & 0 & \mu_1 & 0 & 0 & -\mu_3 & 0 & \mu_2 \\
0 & 0 & 0 & 0 & 0 & 0 & \mu_1 & 0 & 0 & -\mu_2 & \mu_3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_1 & 0 & 0 & -\mu_2-\mu_3
\end{vmatrix} \tag{2.35}$$

2.1.3 A simple non-linear flow model

Non-linear flow models have the characteristic that parts may be returned to upstream stations or skip stations or meet other parts at particular stations for assembly or two or more parts emerge from a disassembly station. Thus, non-linearity implies some lack of strict successive continuity of a distinct product going from one station to the succeeding station in a production line.

In non-linear flow models consideration is given to assembly/disassembly and merge operations in production lines. These models may also take account of quality inspection stations and allow for the possibility of rework where a product is returned to earlier stations. Clearly, the topology of non-linear flow is more complicated than linear flow models.

Here, consideration is given to the non-linear flow model shown in Figure 2.10.

The merge phenomenon is indicated in Figure 2.10. Two machines upstream from the buffer perform the same operation and feed the buffer in such a way that one machine has priority over the other when the buffer is full. The third machine

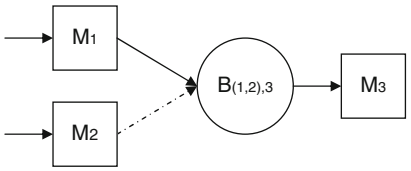


Fig. 2.10. A merge non-linear flow model

removes material from the buffer. The circle indicates a buffer of finite capacity and the squares indicate the machines. Thus the priority-one buffer is always selected first unless it is empty, where the priority-two buffer is chosen. The machines may break down.

Among the major assumptions of the model investigated here, are that the two upstream machines are never starved, the third machine is never blocked, all machines have equal and constant processing times, and geometrically distributed repair times and times to failures and machines can only fail while processing. The phenomenon of partial and full blocking of the second upstream machine is taken into account.

In order to analyze the model of Figure 2.10, the following three methodological steps are required:

(i) Derivation of the transition equations of all states of the system (internal, lower boundary and upper boundary).
(ii) Development of a recursive algorithm for generating the transition matrix for any value C of the extended storage level of buffer $B_{(1,2),3}$ (that is, the capacity of the original buffer plus 3).
(iii) Numerical computation of the transition probabilities and then of the various performance measures of the system.

A formula for the number of states, m, for any value $C > 4$ of the extended storage level of buffer $B_{(1,2),3}$ is given by

$$m = (8 \times C) - 4. \tag{2.36}$$

The expected in-process inventory (average buffer level), $\overline{WIP}$, of the system of Figure 2.10 may be written as follows:

$$\overline{WIP} = \sum_{c=0}^{C} \sum_{\alpha_1=0}^{1} \sum_{\alpha_2=0}^{1} \sum_{\alpha_3=0}^{1} cp[c, \alpha_1, \alpha_2, \alpha_3] \tag{2.37}$$

where $p[c, \alpha_1, \alpha_2, \alpha_3]$ denotes the steady-state probability of the system being in state $[c, \alpha_1, \alpha_2, \alpha_3]$. The level of the extended buffer is denoted by c. $\alpha_i, i = 1, 2, 3$, denotes the status of machine M_i, which may be up ($\alpha_i = 1$) or down ($\alpha_i = 0$).

The blocking probabilities of machines M_1 and M_2, denoted by p_1^{bl}, p_2^{bl} and the starvation probability of machine M_3, denoted by p_3^{st} are

$$p_1^{bl} = p[C-1,1,0,0] + p[C-1,1,0,1] + p[C,1,1,0] + p[C,1,1,1], \tag{2.38}$$

$$p_2^{bl} = p[C-1,0,1,0] + p[C-1,0,1,1] + p[C-1,1,1,0] + p[C-1,1,1,1] + p[C,1,1,0] + p[C,1,1,1], \tag{2.39}$$

$$p_3^{st} = p[0,0,0,1] + p[0,0,1,1] + p[0,1,0,1] + p[0,1,1,1]. \tag{2.40}$$

The mean production rates related to each one of the three machines can be readily determined. If β_i and r_i are the mean rates of failure and repair, respectively, of machine M_i, then $e_i = r_i/(r_i+\beta_i)$, $i = 1,2,3$ represents the fraction of time that machine M_i is operational. Since all processing times are identical and are taken as the time unit, it is obvious that e_i, $i = 1,2,3$, is the isolated mean production rate of machine M_i, i.e., the mean production rate of machine M_i, if it were working alone. Since machines M_i , $i = 1,2,3$ are part of the system, blocking and starvation probabilities should be taken into account. Therefore the mean production rates (throughputs) related to each one of these three machines are

$$X_1 = (1-p_1^{bl})e_1 \tag{2.41}$$

$$X_2 = (1-p_2^{bl})e_2 \tag{2.42}$$

$$X_3 = (1-p_3^{st})e_3. \tag{2.43}$$

In order to determine the throughput of the system shown in Figure 2.10, the throughput of the third machine should be computed. The throughput, X, of the system is simply given by X_3.

$$X = X_3. \tag{2.44}$$

In Diamantidis, Papadopoulos and Vidalis (2004), a process for the generation of the transition matrix was developed and an algorithm to evaluate the performance parameters, including the average buffer level and the throughput of the system, was presented.

2.2 Decomposition Approach

Queueing networks are a natural way of analyzing production lines. Although there is a very rich literature in queueing networks, difficulties in analysis arise when finite buffers are considered because of the associated starving and blocking phenomena. Classically, queueing networks tended to be investigated through a process of decomposition into a set of single-server systems. Such an approach is totally valid in the case of infinite buffers. With finite buffers, exact classical decomposition is inappropriate. However, many researchers have developed efficient decomposition methods for the approximate evaluation of tandem queues which are suitable for the analysis of production lines. As noted above, numerical techniques based on exact Markovian analysis are space and computer time consuming for solutions to the exact queueing problems and are generally only applicable in practice to small production lines. Considerable effort has been expended on the development of approximate solutions to large-scale production lines.

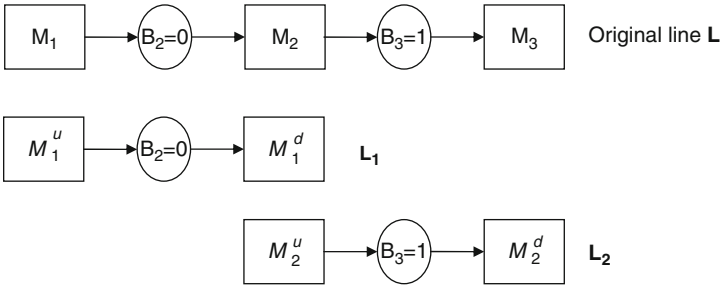


Fig. 2.11. A three-station line, L, decomposed into two sub-lines, L_1 and L_2

Essentially, the decomposition approach as applied to a K-station line consists of decomposing the original line into a set of $K-1$ sub-lines. Each sub-line normally consists of two stations and an intermediate buffer which corresponds to a buffer of the original line. The original method proposed by Gershwin (1987) was initially used in the analysis of serial production lines. Consider the following example.

Example: Consider a balanced three-station production line where each work-station consists of only one machine (this is the original line L). All machines are assumed to be perfectly reliable with exponentially distributed service times and identical mean service rates, $\mu_1 = \mu_2 = \mu_3 = 1$. There is no intermediate buffer between the first two stations, whereas there is an intermediate buffer of size 1 between the second and third station, viz., $B_2 = 0$ and $B_3 = 1$. The original line, L, is decomposed into two sub-lines, L_1 and L_2, as shown in Figure 2.11. X_K denotes the throughput of the K-station line, whereas X_1 and X_2 denote the throughput of sub-line L_1 and sub-line L_2, respectively.

Following the approach of Gershwin (1994):

The *boundary conditions* of the decomposition method are

$$\mu_1^u = \mu_1, \tag{2.45}$$

$$\mu_{K-1}^d = \mu_K. \tag{2.46}$$

The general steps of the decomposition method are the following:

Step 1: Initialization

$$\mu_i^u = \mu_i, \quad i = 1,2,\ldots,K-1$$
$$\mu_i^d = \mu_{i+1}, \quad i = 1,2,\ldots,K-1.$$

Step 2: Iteration

Perform the following steps 2.1 and 2.2 alternately until the termination condition is satisfied.

Step 2.1: Evaluate quantities

$$\mu_i^u = \frac{1}{\frac{1}{X_{i-1}} + \frac{1}{\mu_i} - \frac{1}{\mu_{i-1}^d}}, \quad i = 2,3,\ldots,K-1$$

Step 2.2: Evaluate quantities

$$\mu_i^d = \frac{1}{\frac{1}{X_{i+1}} + \frac{1}{\mu_{i+1}} - \frac{1}{\mu_{i+1}^u}}, \quad i = K-2, K-3, \ldots, 1$$

Step 3: Termination condition

The algorithm is terminated when

$$|X_i - X_1| < \varepsilon, \quad i = 2, 3, \ldots, K-1,$$

where ε is a pre-determined very small positive real number.

Application of the above decomposition algorithm to the example three-station balanced production line, L:

INITIALIZATION: $\mu_1^u = \mu_1,\ \mu_2^u = \mu_2,\ \mu_1^d = \mu_2,\ \mu_2^d = \mu_3$.

FIRST ITERATION (I1)

Step 2.1: From the boundary condition: $\mu_1^u = \mu_1 = 1$ and

$$\begin{aligned}\mu_2^u &= \frac{1}{\frac{1}{X_1} + \frac{1}{\mu_1} - \frac{1}{\mu_1^d}} \\ &\overset{\mu_1^d = \mu_2}{=} \frac{1}{\frac{1}{X_1} + \frac{1}{1} - \frac{1}{\mu_2}} \\ &= \frac{1}{\frac{1}{0.666667} + \frac{1}{\mu_1} - \frac{1}{1}} = 0.666667,\end{aligned}$$

where X_1 is the throughput of sub-line L_1 which is calculated from the Markovian algorithm with the parameter values $\mu_1^u = \mu_1 = 1$, $\mu_1^d = \mu_2 = 1$ and $B_2 = 0$ and gives the value $X_1 = 0.666667$.

Step 2.2: From the boundary condition: $\mu_2^d = \mu_3 = 1$ and

$$\begin{aligned}\mu_1^d &= \frac{1}{\frac{1}{X_2} + \frac{1}{\mu_2} - \frac{1}{\mu_2^u}} \\ &\overset{Step 2.1}{=} \frac{1}{\frac{1}{X_2} + \frac{1}{1} - \frac{1}{0.666667}} \\ &= \frac{1}{\frac{1}{0.584573} + \frac{1}{\mu_1} - \frac{1}{0.666667}} = 0.826001,\end{aligned}$$

where X_2 is the throughput of sub-line L_2 which is calculated from the Markovian algorithm with the parameter values $\mu_2^u = 0.666667$, $\mu_2^d = \mu_3 = 1$ and $B_3 = 1$ and gives the value $X_2 = 0.584573$.

TERMINATION CONDITION: X_1 is the throughput of sub-line L_1 which is calculated from the Markovian algorithm with the parameter values $\mu_1^u = \mu_1 = 1$, $\mu_1^d = 0.826001$ and $B_2 = 0$ and gives the values $X_1 = 0.601320$ and $X_2 = 0.584573$

calculated in step 2.2 above. Thus, $|X_2 - X_1| > \varepsilon = 0.0001$ and the two-step iteration continues.

SECOND ITERATION (I2)

Step 2.1: From the boundary condition: $\mu_1^u = \mu_1 = 1$ and

$$\mu_2^u = \frac{1}{\frac{1}{X_1} + \frac{1}{\mu_1} - \frac{1}{\mu_1^d}} \overset{Step2.2(I1)}{=} \frac{1}{\frac{1}{X_1} + \frac{1}{1} - \frac{1}{0.826001}} = \frac{1}{\frac{1}{0.601320} + 1 - \frac{1}{0.826001}} = 0.688536.$$

Step 2.2: From the boundary condition: $\mu_2^d = \mu_3 = 1$ and

$$\mu_1^d = \frac{1}{\frac{1}{X_2} + \frac{1}{\mu_2} - \frac{1}{\mu_2^u}} \overset{Step2.1(I2)}{=} \frac{1}{\frac{1}{X_2} + \frac{1}{1} - \frac{1}{0.688536}} = \frac{1}{\frac{1}{0.598217} + 1 - \frac{1}{0.688536}} = 0.820157,$$

where X_2 is the throughput of sub-line L_2 which is calculated from the Markovian algorithm with the parameter values $\mu_2^u = 0.688536$, $\mu_2^d = \mu_3 = 1$ and $B_3 = 1$ and gives the value $X_2 = 0.598217$.

TERMINATION CONDITION: X_1 is the throughput of sub-line L_1 which is calculated from the Markovian algorithm with the parameter values $\mu_1^u = \mu_1 = 1$, $\mu_1^d = 0.820157$ and $B_2 = 0$ and gives the values $X_1 = 0.598823$ and $X_2 = 0.598217$ calculated in step 2.2 above. Again $|X_2 - X_1| > \varepsilon = 0.0001$ and the two-step iteration continues.

THIRD ITERATION (I3)

Step 2.1: From the boundary condition: $\mu_1^u = \mu_1 = 1$ and

$$\mu_2^u = \frac{1}{\frac{1}{X_1} + \frac{1}{\mu_1} - \frac{1}{\mu_1^d}} \overset{Step2.2(I2)}{=} \frac{1}{\frac{1}{X_1} + 1 - \frac{1}{0.820157}} = \frac{1}{\frac{1}{0.598823} + 1 - \frac{1}{0.820157}} = 0.689339.$$

Step 2.2: From the boundary condition: $\mu_2^d = \mu_3 = 1$ and

$$\mu_1^d = \frac{1}{\frac{1}{X_2} + \frac{1}{\mu_2} - \frac{1}{\mu_2^u}} \overset{Step2.1(I3)}{=} \frac{1}{\frac{1}{X_2} + \frac{1}{1} - \frac{1}{0.689339}} = \frac{1}{\frac{1}{0.598707} + 1 - \frac{1}{0.689339}} = 0.819940,$$

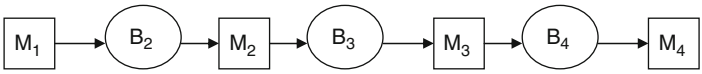
where X_2 is the throughput of sub-line L_2 which is calculated from the Markovian algorithm with the parameter values $\mu_2^u = 0.689339$, $\mu_2^d = \mu_3 = 1$ and $B_3 = 1$ and gives the value $X_2 = 0.598707$.

TERMINATION CONDITION: X_1 is the throughput of sub-line L_1 which is calculated from the Markovian algorithm with the parameter values $\mu_1^u = \mu_1 = 1$, $\mu_1^d = 0.819940$ and $B_2 = 0$ and gives the values $X_1 = 0.598738$ and $X_2 = 0.598707$ calculated in step 2.2 above. Now, $|X_2 - X_1| = 0.000031 < \varepsilon = 0.0001$ and the algorithm terminates giving a throughput value, $X_{\text{DECO}} = 0.5987$. This value may be compared against the throughput value calculated from the Markovian algorithm, $X_{\text{MARK}} = 0.613333$. The reader may note that if the value of ε was smaller, say $\varepsilon = 0.00001$, then more iterations would be needed for the algorithm to terminate.

A coded version of the original Gershwin's decomposition algorithm is not at the website associated with this book. However, the algorithm developed by Dr. Diamantidis (Diamantidis, Papadopoulos and Heavey, 2006) for parallel perfectly reliable machine production lines (given in Section 2.5 and at the website associated with this text with abbreviated name DECO-2) may be used by setting the number of parallel machines at each station equal to 1 as an alternative to the Gershwin algorithm for perfectly reliable single-machine stations. The authors have checked that the equations derived from Diamantidis et al. (2006) work, and setting the number of servers at each station equal to 1 showed them to be identical to those developed by Gershwin for the single-machine perfectly reliable production lines.

Dallery and his group undertook considerable work in the area of decomposition modeling. Available at the website associated with this book with abbreviated name DECO-1 is a coded version of an algorithm given in Dallery and Frein (1993) for the analysis of reliable production lines with single machines at each station. To illustrate the development of Dallery and Frein's algorithm, consider the four-station production line with finite inter-station buffers, B_2, B_3 and B_4, with capacities B_2, B_3 and B_4, respectively, shown in Figure 2.12. For simplicity, work-stations are denoted by $M_i,\ 1 = 1,2,3,4$ instead of $WS_i,\ i = 1,\ldots,4$. Figure 2.13 depicts the decomposition of this four-station line into three sub-lines, L_1, L_2, and L_3, each consisting of two stations and one intermediate buffer. All single-machine work-stations are assumed to be perfectly reliable and the service times at each machine are exponentially distributed with mean service rates, $\mu_i,\ i = 1,\ldots,4$.

The application of queueing network theory to production lines involved the use of either the open model or the saturated model. In the saturated model, the assumption is that the first station is never starved, whereas in the open model, the first queue



M_i : work-station i, i = 1,2,3,4

B_i : buffer i, i = 2,3,4

Fig. 2.12. Production line, L, with $K = 4$ work-stations and 3 intermediate buffers

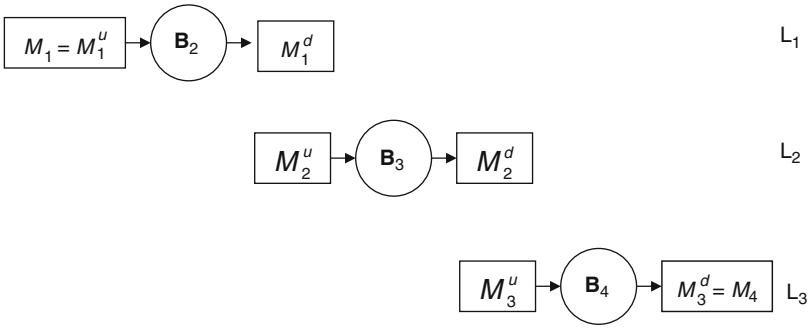


Fig. 2.13. Decomposition of the original line, L, into three sub-lines each with two stations and one buffer

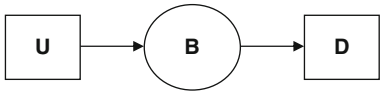


Fig. 2.14. Sub-line L_i

is finite and the first station may be starved. The model considered here is a saturated model. Actually, all models examined in this book are saturated models except for the model solved via the expansion method (Section 2.3).

Some formal results from queueing theory are required in the development of the decomposition equations and these are given immediately below.

Queueing theory analysis of the sub-lines

Consider the sub-line shown in Figure 2.14.

There are $B+2$ states in the system of Figure 2.14, where B is the buffer size of the intermediate buffer B. Some of the performance parameters of a two-station system, called decomposition block, are required for the development of the decomposition equations. The states of the system are characterized by $\nu = 0, 1, \ldots, B+1$, where ν denotes the number of jobs waiting for service at station 2. $\nu = B+1$ occurs when station U is blocked. μ_U and μ_D, respectively, denote the mean service rate of machines U and D. Let $p(\nu)$ be the steady-state probability of the system being in

state v and let $p_U(v)$ be the probability that there are v jobs in the buffer when a job is completed at station U. Likewise, let $p_D(v)$ be the probability that on completion of service at station D, there are v jobs left in buffer B. The throughput of the system is denoted by X. Using well-known results, the following relationships apply:

$$p_U(v) = \frac{p(v)}{1-p(B+1)}, \quad v = 0,\ldots,B \tag{2.47}$$

$$p_D(v) = \frac{p(v+1)}{1-p(0)}, \quad v = 0,\ldots,B \tag{2.48}$$

$$X = \mu_U(1-p(B+1)) \tag{2.49}$$

$$X = \mu_D(1-p(0)). \tag{2.50}$$

Two further probabilities are of interest, these being p_U^{bl}, the probability that station U is blocked, and p_D^{st}, the probability that station D is starved. Clearly these are given by

$$p_U^{bl} = p_U(B) \tag{2.51}$$

$$p_D^{st} = p_D(0). \tag{2.52}$$

Returning to the decomposition process, in Figure 2.13, sub-line L_i, $i = 1,2,3$ $(= K-1)$ approximates the flow of jobs in buffer B_i, $i = 2,3,4(= K)$ of the original line. In sub-lines L_i, $i = 1,\ldots,K-1$, stations M_i^u, $i = 1,\ldots,K-1$ and M_i^d, $i = 2,\ldots,K-1$ represent the part of the original line, L, upstream and downstream of buffer B_{i+1}, $i = 1,\ldots,K-1$, respectively. The concept is that whereas the buffer B_i is in all respects identical to the buffer preceding station i in the original line L, the two stations M_i^u and M_i^d are not identical to stations $i-1$ and i, except that station M_1^u is identical to station M_1 and station M_3^d is identical to station M_4. The characteristics of the remaining stations are so chosen that in effect they represent the impact of the upstream and downstream parts of the production line L on the buffer $B_i, i = 2,3,4$. μ_i, $i = 1,\ldots,K$ denotes the mean service rate of station i, $i = 1,\ldots,K$ in the original line, L. Similarly, μ_i^u, $i = 1,\ldots,K-1$ and μ_i^d, $i = 1,\ldots,K-1$ denote the mean service rate of stations $M_i^u, i = 1,\ldots,K-1$ and M_i^d, $i = 1,\ldots,K-1$, respectively, in the sub-lines, L_i, $i = 1,\ldots,K-1$. All service times are assumed to be exponentially distributed with the respective mean service rates given above.

The development of the sets of the decomposition equations starts out by considering the sub-lines L_1 and L_3. This gives the boundary conditions already stated above that $\mu_1^u = \mu_1$ and $\mu_3^d = \mu_4 = \mu_K$.

In total, there are in general $2(K-1)$ unknowns, and there is a need to obtain another $2(K-2)$ independent equations.

$w_i = 1/\mu_i$, $i = 1,\ldots,K$ denotes the mean service time (average work-load) at station i, $i = 1,\ldots,K$ of the original line, L. The mean service time of the downstream station $M_i^d, i = 1,2,\ldots,K-2$ $(K-2 = 4-2 = 2$ for the example line of Figure 2.12 and the sub-lines of Figure 2.13), denoted by $w_i^d = 1/\mu_i^d$, is the sum of the service time at station i in the original line and the possible blocking time of station M_i in the

original line L, which is equivalent to the blocking time of station M_{i+1}^u in sub-line L_{i+1} due to the fact that buffer B_{i+2} is full and on the assumption that the station is perfectly reliable. This gives rise, in general, to the following set of equations for the reliable exponential production lines:

$$w_i^d = w_i + p_{i+1}^{bl} w_{i+1}^d, \quad i = 1,2,\ldots,K-2, \tag{2.53}$$

where, p_{i+1}^{bl} denotes the blocking probability of station M_{i+1}^u.

A similar set of equations may be developed for the upstream stations. More specifically, the mean service time of the upstream station $M_i^u, i = 2,\ldots,K-1$ ($K-1 = 4-1 = 3$ for the example line of Figure 2.12 and the sub-lines of Figure 2.13), denoted by $w_i^u = 1/\mu_i^u$, is the sum of the service time of station $i-1$ in the original line and the possible starvation time of station $i-1$. The latter event in the original line is equivalent to the starvation of station M_{i-1}^d in sub-line L_{i-1}. This gives rise, in general, to the following set of equations for the reliable exponential production lines:

$$w_i^u = w_{i-1} + p_{i-1}^{st} w_{i-1}^u, \quad i = 2,3,\ldots,K-1, \tag{2.54}$$

where p_{i-1}^{st} denotes the starvation probability of station M_{i-1}^d.

The third set of equations is related to the conservation of flow, i.e., the throughput of all stations in the line is the same and consequently the throughput of the sub-lines must satisfy the following flow equations:

$$X_1 = X_2 = \cdots = X_{K-1}, \tag{2.55}$$

where X_i denotes the throughput of sub-line $L_i,\ i = 1,\ldots,K-1$.

As may be noted from the above, there are two sets of $K-2$ equations plus two boundary conditions, so it is not necessary to use all the equations to solve for the unknowns. This leads to the utilization of the following sub-set of the above equations:

$$w_i^u = w_{i-1} + p_{i-1}^{st} w_{i-1}^u, \quad i = 2,3,\ldots,K-1, \tag{2.56}$$

$$w_i^d = w_i + p_{i+1}^{bl} w_{i+1}^d, \quad i = 1,2,\ldots,K-2, \tag{2.57}$$

$$w_1^u = w_1, \quad \text{and} \quad w_{K-1}^d = w_K \tag{2.58}$$

Dallery and Frein (1993) proved that the above set of equations satisfies the conservation of flow criterion. They also proved the existence and uniqueness of the solution derived from this set of equations and that this symmetrical set of equations is equivalent to each of the following sets of equations:

$$w_i^d = w_i + p_{i+1}^{bl} w_{i+1}^d, \quad i = 1,2,\ldots,K-2, \tag{2.59}$$

$$X_1 = X_2 = \cdots = X_{K-1}, \tag{2.60}$$

$$w_1^u = w_1, \quad \text{and} \quad w_{K-1}^d = w_K. \tag{2.61}$$

$$w_i^u = w_{i-1} + p_{i-1}^{st} w_{i-1}^u, \quad i = 2,3,\ldots,K-1, \tag{2.62}$$

$$X_1 = X_2 = \cdots = X_{K-1}, \tag{2.63}$$

$$w_1^u = w_1, \quad \text{and} \quad w_{K-1}^d = w_K. \tag{2.64}$$

Iterative procedures for solving the above three sets of equations have been proposed by Dallery and Frein (1993) and form the basis of the decomposition algorithm available at the website associated with this book.

The numerical processes involved in the algorithm are relatively straightforward. The two boundary conditions on the mean service rates of the first and last stations of the line are set and then the mean service rate of each of the other downstream stations are set equal to the values of the original line. The starvation and blocking probabilities are then calculated and values of the upstream and changed values of the downstream stations mean service rates are developed. This process continues until satisfactory convergence is achieved. Finally, the throughput of the line may be determined. The numerical decomposition process is outlined in flow diagram form in Figure 2.15.

In general, the decomposition method as applied to production lines consists essentially of three steps as follows:

1. The specification of the sub-lines
2. The determination of a set of equations used to evaluate the unknown parameters of each sub-line in such a way that the flow of material through the sub-lines resembles the corresponding flow of material in the original line. More specifically, the following conditions have to be satisfied as explicitly given in Gershwin (1994):
   - The rate of flow into and out of buffer B_i in sub-line L_i approximates that of buffer B_i in the original line L.
   - The probability of the buffer of sub-line L_i being empty or full is close to that of B_i in the original line L being empty or full.
   - The probability of resumption of flow into and out of the buffer in sub-line L_i in a time interval after a period during which it was interrupted is close to the probability of the corresponding event in the original line L.
   - The average level of material in buffer B_i in sub-line L_i approximates the corresponding material level in buffer B_i in the original line L.

   These conditions lead in the case of perfectly reliable production lines to the conservation of flow equations and in the case of unreliable production lines to the addition of the resumption of flow equations and interruption of flow equations.
3. The development of an appropriate procedure for solving the set of equations.

2.3 The Expansion Method

The expansion method is an approximation technique developed by Kerbache (1984), published also in Kerbache and MacGregor Smith (1987) and extended by Jain and MacGregor Smith (1994). This method is characterized as a combination of repeated trials and node-by-node decomposition solution procedures. Methodologies for computing performance measures for a finite queuing network use the following two kinds of blocking:

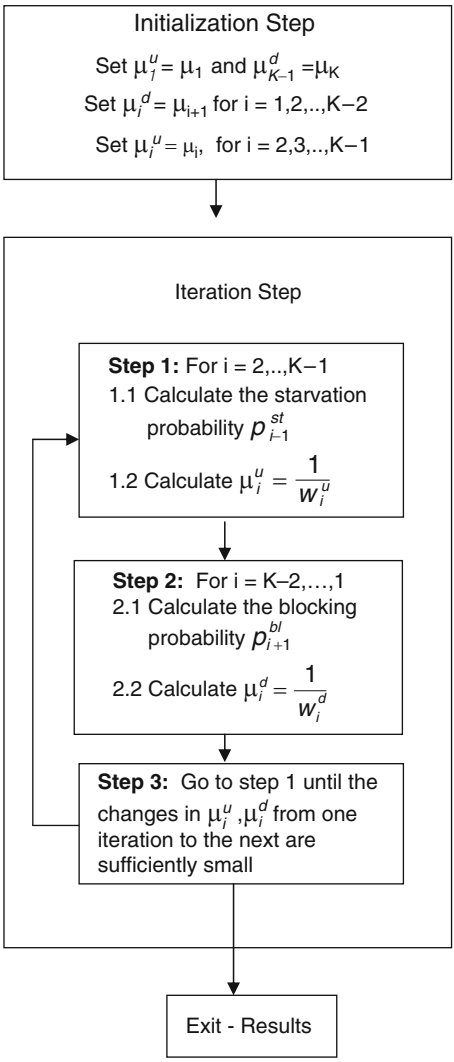


Fig. 2.15. Flow chart for decomposition method

Type I: The upstream node *i* gets blocked if the service on a unit is completed but it cannot move downstream due to the queue at the downstream node *j* being full. This is referred to as blocking after service (BAS) (Onvural and Perros, 1986, Perros, 1994).

Type II: The upstream node is blocked when the downstream node becomes saturated and service must be suspended on the upstream unit regardless of whether service is completed or not. This is referred to as blocking before service (BBS) (Onvural and Perros, 1986, Perros, 1994).

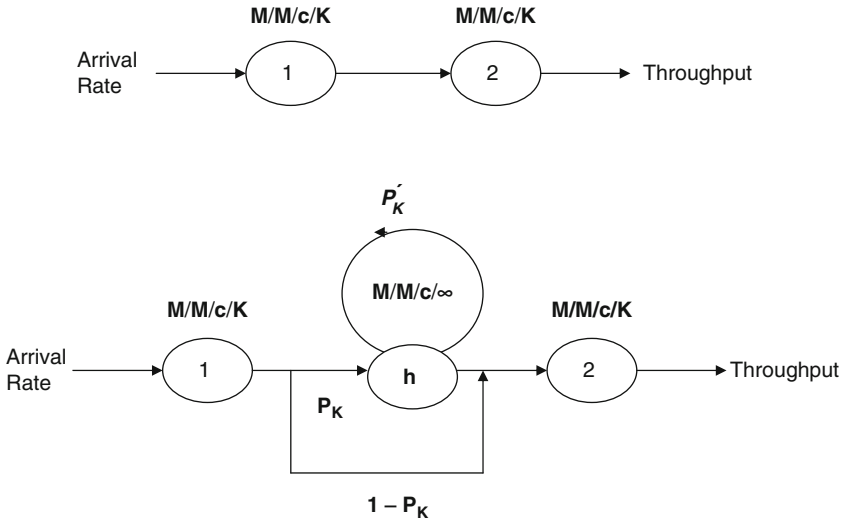


Fig. 2.16. Expansion of a finite queue $M/M/c/K$

The Expansion Method uses *Type I* blocking, which is prevalent in most production, manufacturing, transportation and other similar systems.

Consider a single node with finite capacity K (including service). This node essentially oscillates between two states—the saturated phase and the unsaturated phase. In the unsaturated phase, node j has at most $K-1$ units (in service or in the queue). On the other hand, when the node is saturated no more units can join the queue. Refer to Figure 2.16 for a graphical representation of the expansion of a finite queue $M/M/c/K$. The reader may note that this model is the only open model considered in this book. All the other models are saturated models.

The Expansion Method consists of the following three stages:

- Stage I: Network reconfiguration.
- Stage II: Parameter estimation.
- Stage III: Feedback elimination.

The following notation defined by Kerbache and MacGregor Smith (1987) and Jain and MacGregor Smith (1994) will be used in further discussion regarding this methodology:

$h :=$ The holding node established in the expansion method.
$\Lambda :=$ External Poisson arrival rate to the network.
$\lambda_j :=$ Poisson arrival rate to node j.
$\tilde{\lambda}_j :=$ Effective arrival rate to node j.
$\mu_j :=$ Exponential mean service rate at node j.
$\tilde{\mu}_j :=$ Effective service rate at node j due to blocking.
$p_K :=$ Blocking probability of finite queue of size K.
$p'_K :=$ Feedback blocking probability in the expansion method.

$p_0^j :=$ Unconditional probability that there is no unit in the service channel at node j (either being served or being held after service).
$X :=$ Mean production rate (throughput).

Stage I: Network reconfiguration

Using the concept of two phases at node j, an artificial node h is added for each finite node in the network to register blocked units. Figure 2.16 shows the additional delay, caused to units trying to join the queue at node j when it is full, with probability p_K. The units successfully join queue j with a probability $(1 - p_K)$. Introduction of an artificial node also dictates the addition of new arcs with p_K and $(1 - p_K)$ as the routing probabilities.

The blocked unit proceeds to the finite queue with probability $(1 - p'_K)$ once again after incurring a delay at the artificial node. If the queue is still full, it is re-routed with probability p'_K to the artificial node where it incurs another delay. This process continues until it finds a space in the finite queue. A feedback arc is used to model the repeated delays. The artificial node is modeled as an $M/M/\infty$ queue. The infinite number of servers is used simply to serve the blocked unit a delay time without queuing.

Stage II: Parameter estimation

This stage essentially estimates the parameters p_K, p'_K and μ_h utilizing known results for the $M/M/c/K$ model.

- p_K: Analytical results from the $M/M/c/K$ model provide the following expression for p_K:

$$p_K = \frac{1}{c^{K-c}c!}\left(\frac{\lambda}{\mu}\right)^K p_0 \tag{2.65}$$

where for $(\lambda/c\mu \neq 1)$

$$p_0 = \left[\sum_{n=0}^{c-1}\frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n + \frac{(\lambda/\mu)^c}{c!}\frac{\left[1-(\lambda/c\mu)^{K-c+1}\right]}{(1-\lambda/c\mu)}\right]^{-1} \tag{2.66}$$

and for $(\lambda/c\mu = 1)$,

$$p_0 = \left[\sum_{n=0}^{c-1}\frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n + \frac{(\lambda/\mu)^c}{c!}(K-c+1)\right]^{-1}. \tag{2.67}$$

- p'_K: Since there is no closed form solution for this quantity, an approximation obtained by Labetoulle and Pujolle (1980), using diffusion techniques, is used:

$$p'_K = \left[\frac{\mu_j + \mu_h}{\mu_h} - \frac{\lambda\left[(x_2^K - x_1^K) - (x_2^{K-1} - x_1^{K-1})\right]}{\mu_h\left[(x_2^{K+1} - x_1^{K+1}) - (x_2^K - x_1^K)\right]}\right]^{-1} \tag{2.68}$$

where x_1 and x_2 are the roots to the polynomial:

$$\lambda - (\lambda + \mu_h + \mu_j)x + \mu_h x^2 = 0 \tag{2.69}$$

where $\lambda = \lambda_j - \lambda_h(1 - p'_K)$ and λ_j and λ_h are the actual arrival rates to the finite and artificial holding nodes respectively.

In fact, λ_j the arrival rate to the finite node is given by:

$$\lambda_j = \tilde{\lambda}_i(1 - p_K) = \tilde{\lambda}_i - \lambda_h . \tag{2.70}$$

If an arriving unit is blocked, the queue is full and thus a unit is being serviced, so the arriving unit to the holding node has to remain in service at the artificial holding node for the remaining service time interval of the unit in service. The delay distribution of a blocked unit at the holding node has the same distribution as the remaining service time of the unit being serviced at the node doing the blocking. Using renewal theory, one can show that the remaining service time distribution has the following rate μ_h:

$$\mu_h = \frac{2\mu_j}{1 + \sigma^2 \mu_j^2} \tag{2.71}$$

where σ^2 is the service time variance given by Kleinrock (1975). Notice that if the service time distribution at the finite queue doing the blocking is exponential with rate μ_j, then:

$$\mu_h = \mu_j$$

the service time at the artificial node is also exponentially distributed with rate μ_j.

Stage III: Feedback elimination

Due to the feedback loop around the holding node, there are strong dependencies in the arrival processes. Elimination of these dependencies requires reconfiguration of the holding node which is accomplished by recomputing the service time at the node and removing the feedback arc. The new service rate is given by:

$$\mu'_h = (1 - p'_K)\mu_h . \tag{2.72}$$

The probabilities of being in any of the two phases (saturated or unsaturated) are p_K and $(1 - p_K)$, respectively. The mean service time at a node i, preceding the finite node is μ_i^{-1} if in the unsaturated phase and $(\mu_i^{-1} + \mu_h'^{-1})$ in the saturated phase. Thus, on average, the mean service time at the node i preceding a finite node, is given by:

$$\tilde{\mu_i^{-1}} = \mu_i^{-1} + p_K \mu_h'^{-1} . \tag{2.73}$$

Similar equations can be established with respect to each of the finite nodes. Ultimately, a set of simultaneous non-linear equations in variables p_K, p'_K, μ_h^{-1} along with auxiliary variables such as μ_j and $\tilde{\lambda}_i$ is developed. Solvingthese equations

simultaneously, all the performance measures of the network can be computed:

$$\lambda = \lambda_j - \lambda_h(1 - p'_K) \tag{2.74}$$

$$\lambda_j = \tilde{\lambda}_i(1 - p_K) \tag{2.75}$$

$$\lambda_j = \tilde{\lambda}_i - \lambda_h \tag{2.76}$$

$$p'_K = \left[\frac{\mu_j + \mu_h}{\mu_h} - \frac{\lambda[(x_2^K - x_1^K) - (x_2^{K-1} - x_1^{K-1})]}{\mu_h[(x_2^{K+1} - x_1^{K+1}) - (x_2^K - x_1^K)]}\right]^{-1} \tag{2.77}$$

$$z = (\lambda + 2\mu_h)^2 - 4\lambda\mu_h \tag{2.78}$$

$$x_1 = \frac{[(\lambda + 2\mu_h) - z^{\frac{1}{2}}]}{2\mu_h} \tag{2.79}$$

$$x_2 = \frac{[(\lambda + 2\mu_h) + z^{\frac{1}{2}}]}{2\mu_h} \tag{2.80}$$

$$p_K = \frac{1}{c^{K-c}c!}\left(\frac{\lambda}{\mu}\right)^K p_0. \tag{2.81}$$

Equations (2.74) to (2.77) are related to the arrivals and feedback in the *holding* node. Equations (2.78) to (2.80) are used for solving equation (2.77) with z used as a dummy parameter for simplicity of the solution. Finally, equation (2.81) gives the approximation to the blocking probability derived from the exact model for the $M/M/c/K$ queue. Hence, essentially there are five equations to solve, *viz.* (2.74) to (2.77) and (2.81).

To recapitulate, first the network is expanded with an artificial holding node; this stage is followed by the approximation of the routing probabilities, due to blocking, and the service delay in the holding node; and, finally, the feedback arc at the holding node is eliminated. Once these three stages are completed, an expanded network has been developed which can then be used to compute the performance measures for the original network. As a decomposition technique, this approach allows the successive addition of a holding node for every finite node, estimation of the parameters and subsequent elimination of the holding node.

The expansion algorithm is available on the website associated with this text with the abbreviated name EXPAN. Not many practitioners are aware of the expansion method and there is little guidance in the published literature as to the accuracy achieved using the method in the analysis of realistic systems of interest to the designers of production lines. However, it must be recognized that in a historical context, the expansion method was used as a first serious attempt to computationally solve systems with parallel machines at each station.

2.4 The Aggregation Method

Lim, Meerkov and Top (1990) published an approximation approach used in the analysis of transfer lines which has come to be known as the aggregation method. This very powerful method begins by combining the first two machines of the transfer

line into a new combined machine. This aggregated combined machine is then combined with the third machine and this forward aggregation process is continued until the last machine is reached. A backward aggregation process is then applied. The algorithm which is available at the website associated with this book stops when the results of both aggregations (forward and backward) coincide.

Assumptions of the model

A serial transfer line is considered consisting of K machines and $K-1$ intermediate buffers. Machines are assumed to have identical cycle time and the time axis is slotted with the slot/period duration equal to the cycle time. It is assumed that the first machine is never starved and the last machine is never blocked. It is further assumed that a certain machine i, $i = 1,\ldots,K$ produces a part during any time slot/period with probability q_i and fails to do so with probability $1-q_i$, provided that machine i is neither blocked nor starved. Mathematically, q_i is defined as follows:

$$q_i = 1 - \varepsilon\Lambda_i,$$

where $0 < \varepsilon << 1$, which characterizes the asymptotically reliable line, and Λ_i, $i = 1,\ldots,K$ is independent of ε. The Λ_i's were defined by Lim, Meerkov and Top (1990) as the *loss parameters*. Lim et al. (1990) also defined the following function:

$$Q(\alpha,\nu) = \frac{1-\alpha}{1-\alpha^{\nu}}, \quad \alpha \in \mathscr{R}^{+}, \quad \nu \in [1,\infty). \tag{2.82}$$

The two-machine, one-buffer transfer line

A two-machine, one-buffer transfer line in steady state is equivalent to a single aggregated machine characterized by

$$q_{\text{aggregation}} = 1 - \left[\Lambda_2 + \Lambda_1 Q\left(\frac{\Lambda_2}{\Lambda_1},\nu\right)\right]\varepsilon \tag{2.83}$$

Thus, the loss parameter of the equivalent aggregated machine is

$$\Lambda_{\text{aggregation}} = \Lambda_2 + \Lambda_1 Q\left(\frac{\Lambda_2}{\Lambda_1},\nu\right), \tag{2.84}$$

where Λ_1 and Λ_2 are, respectively, the loss parameters of the first and second machine. The mean production rate, X_2, of the two-machine, one-buffer system is given by

$$\begin{aligned} X_2 &= 1 - \left[\Lambda_2 + \Lambda_1 Q\left(\frac{\Lambda_2}{\Lambda_1},\nu\right)\right]\varepsilon + O(\varepsilon^2) \\ &= 1 - \left[\Lambda_1 + \Lambda_2 Q\left(\frac{\Lambda_1}{\Lambda_2},\nu\right)\right]\varepsilon + O(\varepsilon^2). \end{aligned} \tag{2.85}$$

It is obvious that

$$\Lambda_{\text{aggregation}} = \Lambda_1 + \Lambda_2 Q\left(\frac{\Lambda_1}{\Lambda_2}, \nu\right). \tag{2.86}$$

Equations (2.84) and (2.86) show that

$$\Lambda_{\text{aggregation}} = \Lambda_1 + \Lambda_2 Q\left(\frac{\Lambda_1}{\Lambda_2}, \nu\right) = \Lambda_2 + \Lambda_1 Q\left(\frac{\Lambda_2}{\Lambda_1}, \nu\right).$$

Longer lines

The above process can be generalized to the case of homogeneous asymptotically reliable serial transfer lines consisting of K machines and $K-1$ intermediate buffers. The first two machines are combined into an aggregated machine with the loss parameter, Λ_2^f, defined by (2.84), i.e.,

$$\Lambda_2^f = \Lambda_2 + \Lambda_1 Q\left(\frac{\Lambda_2}{\Lambda_1}, \nu_1\right).$$

Superscript 'f' indicates that during the aggregation, one moves forward (from the first to the last machine). The aggregated machine, characterized by Λ_2^f, is now combined with the third machine, defined by the loss parameter Λ_3. The new aggregated machine is characterized by the loss parameter:

$$\Lambda_3^f = \Lambda_3 + \Lambda_2^f Q\left(\frac{\Lambda_3}{\Lambda_2^f}, \nu_2\right).$$

At the ith step of this multi-stage aggregation process, one may obtain:

$$\Lambda_i^f = \Lambda_i + \Lambda_{i-1}^f Q\left(\frac{\Lambda_i}{\Lambda_{i-1}^f}, \nu_{i-1}\right) \tag{2.87}$$

and at the final step:

$$\Lambda_K^f = \Lambda_K + \Lambda_{K-1}^f Q\left(\frac{\Lambda_K}{\Lambda_{K-1}^f}, \nu_{K-1}\right).$$

The estimate of the mean production rate (throughput) obtained as a result of this aggregation is:

$$X_K^f = 1 - \left[\Lambda_K + \Lambda_{K-1}^f Q\left(\frac{\Lambda_K}{\Lambda_{K-1}^f}, \nu_{K-1}\right)\right] \varepsilon. \tag{2.88}$$

Because there is no proof that X_K^f is close to the real throughput of the production line with K machines in series and $K-1$ intermediate buffers, another set of equations should be supplemented, but this time directed backwards instead of forwards. This

scheme is called backward aggregation and aggregates the line moving from the last machine to the first machine. The respective loss paramaters are:

$$\Lambda^b_{K-1} = \Lambda_{K-1} + \Lambda_K\, Q\left(\frac{\Lambda^f_{K-1}}{\Lambda_K}, \nu_{K-1}\right)$$

$$\Lambda^b_{K-2} = \Lambda_{K-2} + \Lambda^b_{K-1}\, Q\left(\frac{\Lambda^f_{K-2}}{\Lambda^b_{K-1}}, \nu_{K-2}\right)$$

$$\Lambda^b_j = \Lambda_j + \Lambda^b_{j+1}\, Q\left(\frac{\Lambda^f_j}{\Lambda^b_{j+1}}, \nu_j\right)$$

$$\Lambda^b_1 = \Lambda_1 + \Lambda^b_2\, Q\left(\frac{\Lambda_1}{\Lambda^b_2}, \nu_1\right).$$

By repeating the process and constructing a new forward aggregation based on the backward aggregation, the following iterative algorithm is obtained:

$$\begin{aligned}
&\Lambda^f_i(s+1) = \Lambda_i + \Lambda^f_{i-1}(s+1)\, Q\left(\frac{\Lambda^b_i(s)}{\Lambda^f_{i-1}(s+1)}, \nu_{i-1}\right), \quad i = 2,\ldots,K \\
&\qquad s = 0,1,\ldots,\ \ \Lambda^b_i(0) = \Lambda_i,\ \ \Lambda^f_1(s) = \Lambda_1,\ \ \Lambda^b_K(s) = \Lambda_K,\ \ \forall s. \qquad (2.89) \\
&\Lambda^b_j(s+1) = \Lambda_j + \Lambda^b_{j+1}(s+1)\, Q\left(\frac{\Lambda^f_j(s+1)}{\Lambda^b_{j+1}(s+1)}, \nu_j\right), \quad j = 1,\ldots,K-1.
\end{aligned}$$

Procedure (2.89) generates the following two sequences of throughput estimates:

$$\begin{aligned}
X^f_K(s) &= 1 - \Lambda^f_K(s)\,\varepsilon \\
X^b_K(s) &= 1 - \Lambda^b_1(s)\,\varepsilon.
\end{aligned} \qquad (2.90)$$

The properties of these sequences are described in Lim, Meerkov and Top (1990). The aggregation algorithm is available at the website associated with this text with abbreviated name AGGRE.

2.5 Modeling of Production Lines with Parallel Reliable Machines at Each Station

The throughput of production lines may be increased by adding extra machines at stations. It should be understood that all machines at the stations are used provided there is work available. Here, attention is confined to lines with reliable machines and with exponential processing times. A K-work-station line with $S_i, i = 1,2,\ldots,K$ parallel machines at work-station i, denoted by WS_i, and with intermediate buffers $B_j, j = 2,\ldots,K$ of capacities B_j is depicted in Figure 2.17.

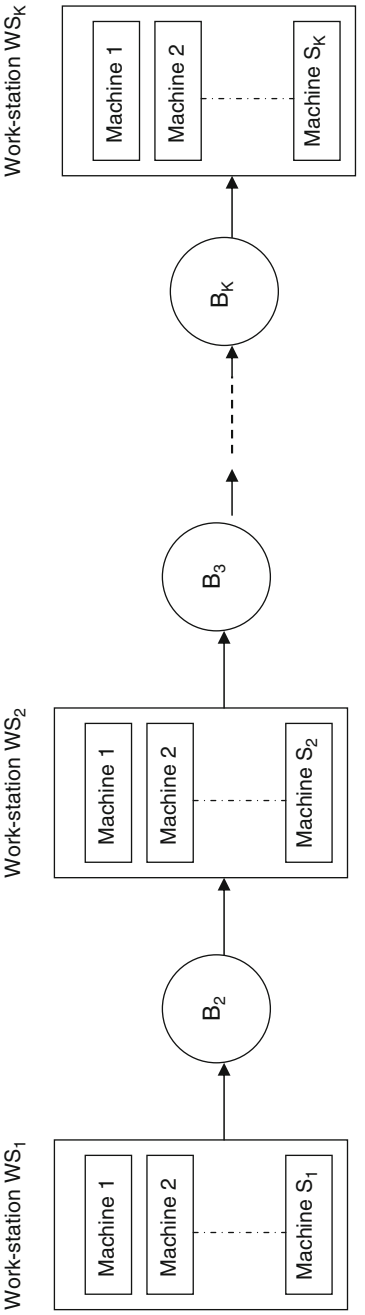


Fig. 2.17. A K-work-station production line with S_i parallel machines at each work-station WS_i, $i = 1, 2, \ldots, K$

In sub-section 2.5.1, the exact solution to a two-station line with multiple machines at each station is presented. It might be noted that it is possible to develop the conservative matrix A of these systems with a view of developing exact numerical solutions along the lines already explained in Section 2.1. Interested readers might refer to Vidalis and Papadopoulos (2001). However, computational complexities considerably reduce the value of developing the conservative matrix A of such systems. In sub-section 2.5.2, an alternative exact solution to the the two-station line with parallel machines at each station is presented as given in Diamantidis, Papadopoulos and Heavey (2006). This solution is used as a building block for a decomposition analysis of larger production lines with parallel machines at each station. Details of the latter analysis are given in sub-section 2.5.3.

2.5.1 Exact solution to a two-station production line with parallel machines at each station

Consider a system consisting of two stations with S_1 and S_2 parallel machines at station 1 and station 2, respectively. It is assumed that the first station is always busy, i.e., it is saturated and the intermediate buffer is of capacity B_2 which includes the number of parallel machines at station 2, i.e., $B_2 \geq S_2$. The processing times at each station are exponentially distributed with mean rates $\mu_i, i = 1,2$. Buzacott and Shanthikumar (1993) (pages 205–206) and Perros (1994) (pages 64–65), among others, considered this problem. By forming the Markovian chain of this system, the random variable of interest is $N(t)$, the number of jobs which have been processed by the first station at time t and have not finished their processing at station 2 (at time t). $N(t), t \geq 0$ is a birth-death process with state space $s = \{0,1,\ldots,S_2,S_2+1,\ldots,B_2,B_2+1,\ldots,B_2+S_1\}$. The birth rate is $\mu_1(\min S_1, B_2+S_1-\nu)$, whereas the death rate is $\mu_2(\min S_2,\nu)$, where, $\nu = 0,1,\ldots,S_2,S_2+1,\ldots,B_2,B_2+1,\ldots,B_2+S_1$. Let $p(\nu)$ be the probability that there are ν jobs in the second station including the jobs in the first station that are blocked. The steady-state (flow balance) equations associated with $p(\nu)$ are (see Perros, 1994, pp. 64–65):

$$
\begin{aligned}
S_1\mu_1 p(0) &= \mu_2 p(1) \\
(S_1\mu_1 + \nu\mu_2)p(\nu) &= (\nu+1)\mu_2 p(\nu+1) + S_1\mu_1 p(\nu-1), \\
&\quad \text{for } \nu = 1,2,\ldots,S_2-1, \\
(S_1\mu_1 + S_2\mu_2)p(\nu) &= S_2\mu_2 p(\nu+1) + S_1\mu_1 p(\nu-1), \\
&\quad \text{for } \nu = S_2,\ldots,B_2, \\
[(S_1+B_2-\nu)(\mu_1+S_2\mu_2)]\,p(\nu) &= S_2\mu_2 p(\nu+1) \\
&\quad +[S_1+B_2-(\nu-1)]\mu_1 p(\nu-1), \\
&\quad \text{for } \nu = B_2+1,\ldots,B_2+S_1-1, \\
S_2\mu_2 p(B_2+S_1) &= \mu_1 p(B_2+S_1-1).
\end{aligned}
\tag{2.91}
$$

In steady state, the throughput of this system may be shown to be:

$$X = \sum_{v=0}^{S_2-1} v\mu_2 p(v) + S_2\mu_2 \sum_{v=S_2}^{B_2+S_1} p(v) \tag{2.92}$$

$$X = \sum_{v=0}^{B_2} S_1\mu_1 p(v) + \sum_{v=B_2+1}^{B_2+S_1} (B_2+S_1-v)\mu_1 p(v) \tag{2.93}$$

where $p(v),\ v = 0,1,\ldots,S_2,S_2+1,\ldots,B_2,B_2+1,\ldots,B_2+S_1$ are obtained from equations (2.91), above, by iteration:

$$p(v) = \begin{cases} \frac{S_1^v}{v!}\left(\frac{\mu_1}{\mu_2}\right)^v p(0), & v = 0,\ldots,S_2 \\ \frac{S_1^v}{S_2!S_2^{v-S_2}}\left(\frac{\mu_1}{\mu_2}\right)^v p(0), & v = S_2+1,\ldots,B_2 \\ \frac{S_1^{B_2}S_1!}{S_2!S_2^{v-S_2}(B_2+S_1-v)!}\left(\frac{\mu_1}{\mu_2}\right)^v p(0), & v = B_2+1,\ldots,B_2+S_1 \end{cases} \tag{2.94}$$

where probability $p(0)$ is obtained from the normalizing condition that the sum of all the steady-state probabilities is equal to 1.

As part of the development of a decomposition method (sub-section 2.5.3), Diamantidis, Papadopoulos and Heavey (2006) also solved the above problem exactly and the algorithm formulated by them is given below in sub-section 2.5.2.

This algorithm is available at the website associated with this book as special case of the 1184 2 (with the abbreviated name DECO-2) for $K = 2$ and in this case it gives the exact solution.

2.5.2 Alternative exact Markovian analysis of a two-station line with parallel machines at each station

The motivation for the development of this solution to the two-station multiple server line was to have available a building block for use in a decomposition approach to the solution of larger lines.

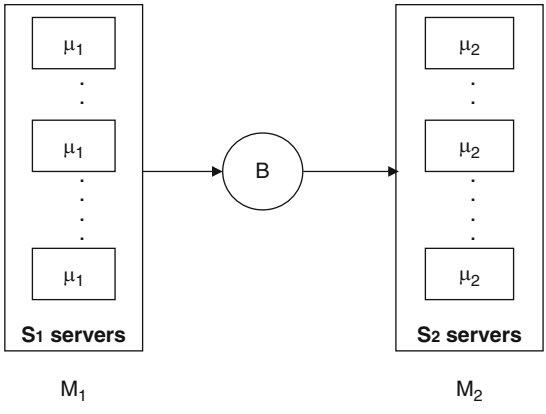


Fig. 2.18. A two-station, one-buffer production line with parallel machines at each station

Consider a system with two work-stations, which, for simplicity, are denoted by M_1 and M_2 (instead of WS_1 and WS_2, respectively) as shown in Figure 2.18 consisting of S_1 and S_2 parallel machines, respectively. This system is used as the decomposition block in the decomposition approach given in sub-section 2.5.2. It is assumed that an inexhaustive supply of workpieces is available upstream of work-station 1, and an unlimited storage area is present downstream of work-station 2, viz., work-station 1 is never starved and work-station 2 is never blocked. Work-station i, $i = 1,2$ consists of S_i reliable and identical machines, arranged in parallel and S_1 need not equal S_2. Each parallel server has an exponentially distributed service time with mean $1/\mu_i$. The size or capacity of the intermediate buffer is denoted by B. The total storage capacity of the system is the physical storage of buffer B as well as the service positions at both work-stations 1 and 2. Therefore, the total storage capacity of the system, C, is $C = S_1 + S_2 + B$. Thus work-station 1 can be either *partially* or *fully blocked*. More specifically, if the current inventory of parts of the system (including those on the machines) equals $S_2 + B + 1$, then only one machine at work-station 1 is blocked and the remaining $S_1 - 1$ machines are not blocked. In this case, work-station 1 is partially blocked. If the storage of the system equals $S_1 + S_2 + B$, then all S_1 machines of the first work-station are blocked and, therefore, this work-station is fully blocked.

Because of the exponentially distributed service times, during the time interval $[t, t + dt]$ it is assumed only one event can occur at each work-station. Thus during the time interval $[t, t + dt]$ only one machine among the S_1 machines of work-station 1 can produce a part, or only one machine among the S_2 machines of work-station 2 can remove a part from buffer B. The total number of units in the system varies from 0 to $S_1 + S_2 + B$. It is straightforward that the total number of states is $S_1 + S_2 + B + 1$. Let $y = (c)$ denote the state of the system, where $c = 0, \ldots, C$.

To solve this two-station system using exact Markovian analysis, the transition matrix must be derived. The following sub-section gives the transition equations. Then in sub-section 2.5.2 an algorithm for generating the transition matrix for any value C is presented.

Transition equations

The system states can be divided with respect to the storage level, c, into three sets: (i) lower boundary states; (ii) internal states; (iii) upper boundary states. It is further assumed that $S_1 \geq 1$, $S_2 \geq 1$ and $B \geq 0$.

Lower boundary state equation

The transition equation for state y with $c = 0$ (referred to as lower boundary state) has the following structure:

$$p_0 = (1 - S_1\mu_1)p_0 + \mu_2 p_1. \tag{2.95}$$

Internal state equations

The transition equations for states $y = (c)$ with $0 < c < C$ can be sub-classified as follows:

Case 1: If $c > 0$ and $c < S_2$, then:

$$p_c = S_1\mu_1 p_{c-1} + (1 - S_1\mu_1 - c\mu_2)p_c + (c+1)\mu_2 p_{c+1}. \tag{2.96}$$

Case 2: If $c > S_2 - 1$ and $c < S_2 + B + 1$, then:

$$p_c = S_1\mu_1 p_{c-1} + (1 - S_1\mu_1 - S_2\mu_2)p_c + S_2\mu_2 p_{c+1}. \tag{2.97}$$

Case 3: If $c > S_2 + B$ and $c < C$, then:

$$\begin{aligned} p_c = {} & S_2\mu_2 p_{c+1} + (C+1-c)\mu_1 p_{c-1} \\ & + (1 - (C-c)\mu_1 - S_2\mu_2)p_c. \end{aligned} \tag{2.98}$$

Upper boundary state equation

The state with storage level $c = C$ is called an upper boundary state. It holds that:

$$p_C = \mu_1 p_{C-1} + (1 - S_2\mu_2)p_C. \tag{2.99}$$

The algorithm for generating the transition matrix

Based on the above classification of the steady-state equations, an algorithm has been developed to generate the transition matrix of the two-station system (called decomposition block in the context of the decomposition approach, given below, in sub-section 2.5.2). Let $P_{ij},\ i, j = 0, \ldots, C$ be the element that is located in the i^{th} row and j^{th} column of the transition matrix P. The algorithm generates the transition probabilities in three stages: (i) transition probabilities of the lower boundary states (see Figure 2.19); (ii) transition probabilities of the internal states (Figure 2.20); (iii) transition probability of the upper boundary state (see Figure 2.19).

The transition matrix for the decomposition block can be generated using the algorithms presented in Figures 2.19 and 2.20. The Gaussian elimination method implemented in C++ is used to solve for the steady-state probabilities. The mean production rate (throughput) (X) of the decomposition block is calculated by using either of the following two formulas:

$$X = S_2\mu_2 \sum_{c=0}^{c=C} p_c - \mu_2 \sum_{c=0}^{c=S_2-1} (S_2 - c)p_c \tag{2.100}$$

or

$$X = S_1\mu_1 \sum_{c=0}^{c=C} p_c - \mu_1 \sum_{c=S_2+B+1}^{c=C} (c - S_2 - B)p_c. \tag{2.101}$$

{**Lower boundary states**}

$P_{0,0} = 1 - S_1\mu_1$
$P_{0,1} = S_1\mu_1$
for $c = 2$ **to** C **do**
  $P_{0,c} = 0.0$
end for

{**Upper boundary states**}

$P_{C,C-1} = S_2\mu_2$
$P_{C,C} = 1 - S_2\mu_2$
for $c = 0$ to $C - 2$ **do**
  $P_{C,c} = 0.0$
end for

Fig. 2.19. Algorithm for generation of lower and upper boundary state transition probabilities

The expected in-process inventory (average storage level), $\overline{WIP}$, of the system can be calculated as follows (the reader is referred to Gershwin, 1994 and Helber, 1999)

$$\overline{WIP} = \sum_{c=0}^{c=C} c\, p_c. \tag{2.102}$$

The method for solving the decomposition block was validated using simulation. Sample results are given Table 2.11. For comparison purposes, a simulation model was developed in Arena V3.0 and the simulation results were found to be close enough to those obtained from the analytical model. Ninety-five percent confidence intervals were computed for any value B. The length of the simulation time is identical for all cases and equals 1100 time units.

For the experiments presented in Table 2.11, the processing times at both work-stations are assumed to be exponentially distributed with mean service rates, $\mu_1 = \mu_2 = 1$. In Table 2.11, the first column gives the number of parallel machines at the first work-station (S_1), the number of parallel machines at work-station 2 (S_2) and the buffer size, B. All these three values are represented by a vector (S_1, S_2, B). The second column gives the throughput obtained by the numerical solution of the exact analytical algorithm proposed by Diamantidis, Papadopoulos and Heavey (2006), described above, while $X_{\text{algorithm}}$ and the third column gives the estimated 95% confidence intervals for the simulated mean production rates.

2.5.3 Approximate methods for large lines

Using the solution of the two-station line as a building block, the decomposition approach was applied by Diamantidis, Papadopoulos and Heavey (2006) to solve large-scale production lines consisting of K parallel-machine work-stations as those shown in Figure 2.21. Each work-station i, denoted for simplicity by M_i in the rest of

for $i = 1$ **to** $C-1$ **do**
  for $j = 0$ **to** $j = C$ **do**
    if $i > j$ **and** $i - j = 1$ **and** $i < S_2$ **then**
      $P_{i,j} = i\mu_2$
    end if
    if $i > j$ **and** $i - j = 1$ **and** $i \geq S_2$ **then**
      $P_{i,j} = S_2\mu_2$
    end if
    if $i = j$ **and** $j < S_2$ **and** $i < S_2 + B + 1$ **then**
      $P_{i,j} = 1 - S_1\mu_1 - j\mu_2$
    end if
    if $i = j$ **and** $j \geq S_2$ **and** $i < S_2 + B + 1$ **then**
      $P_{i,j} = 1 - S_1\mu_1 - S_2\mu_2$
    end if
    if $i = j$ **and** $j \geq S_2$ **and** $i \geq S_2 + B + 1$ **then**
      $K = C - i$
      $P_{i,j} = 1 - K\mu_1 - S_2\mu_2$
    end if
    if $j > i$ **and** $j - i = 1$ **and** $i < S_2 + B + 1$ **then**
      $P_{i,j} = S_1\mu_1$
    end if
    if $j > i$ **and** $j - i = 1$ **and** $i \geq S_2 + B + 1$ **then**
      $m = C - i$
      $P_{i,j} = m\mu_1$
    end if
    if $i > j$ **and** $i - j > 1$ **then**
      $P_{i,j} = 0.0$
    end if
    if $j > i$ **and** $j - i > 1$ **then**
      $P_{i,j} = 0.0$
    end if
  end for
end for

Fig. 2.20. Algorithm for generation of internal state transition probabilities

Table 2.11. Throughput of a two-work-station system with parallel machines

(S_1, S_2, B)	$X_{\text{algorithm}}$	95% CI for Simulated Throughput
(4,4,3)	3.52593	(3.48, 3.57)
(5,5,5)	4.54631	(4.47, 4.60)
(10,15,7)	9.99439	(9.80, 10.15)
(15,20,15)	14.99740	(14.75, 15.14)
(10,10,10)	9.45416	(9.25, 9.56)

this sub-section, consists of multiple identical reliable parallel machines with service rates $\mu_i, i = 1, \ldots, K$ and intermediate buffers $B_i, i = 2, \ldots, K$. The number of parallel machines at station i is $S_i, i = 1, \ldots, K$, with each S_i an integer. Service times are exponentially distributed with mean $1/\mu_i$. It is also assumed that when any one of

the S_i parallel machines at work-station M_i completes a part, that part is placed in the buffer B_{i+1} downstream of the work-station immediately, provided the buffer is not full.

Markovian analysis of flow lines with moderate to large sized K is computationally expensive or impossible due to the enormous resulting state space (see Vidalis and Papadopoulos, 2001). Approximate methods are required to solve large systems. The work reported here uses the decomposition algorithm developed by Diamantidis, Papadopoulos and Heavey (2006). This decomposition method actually extends the work by Gershwin (1987) to solve flow lines with parallel servers at each work-station.

The solution approach for solving large lines with parallel machines at each work-station is as follows:

Following the derivation of the transition equations of the two-station system using exact Markovian analysis, an algorithm for generating the transition matrix for any two-station parallel system is developed. Thereafter, decomposition equations are derived using the well-known two-step methodology of obtaining the conservation flow equations and the flow rate idle time equations. Finally, a decomposition algorithm as outlined in Figure 2.22 was developed.

In the sequence, first, the decomposition equations are derived and then the decomposition algorithm is presented.

2.5.4 Derivation of the decomposition equations

In general, the decomposition method makes use of the four sets of equations (see Gershwin, 1994 where the decomposition method is described in great detail): (i) the conversation of flow equations; (ii) the flow rate idle equations; (iii) the resumption of flow equations; (iv) the interruption of flow equations. As the system analyzed here is reliable, only the first two sets of equations are used.

Conservation of flow equations

Let X_i^L be the mean production rate of the two-work-station, one-buffer sub-line L_i and X_i^u (X_i^d) be the mean production rate of the virtual upstream (downstream) pseudo work-station M_i^u (M_i^d), $i = 1,\ldots,K-1$. The mean production rate of each work-station M_i in the original line is denoted by X_i. The conservation of flow equations states that the production rates of all the two-work-station, one-buffer sub-systems L_i are the same.

Because the flow is conserved, it holds:

$$X_i^L = X_i^u = X_i^d = X_i, \quad i = 1,\ldots,K-1. \tag{2.103}$$

The flow rate idle time equations

Each virtual upstream pseudo work-station M_i^u of sub-line L_i, $i = 1,\ldots,K-1$, consists of S_i parallel machines, while each virtual downstream pseudo work-station M_i^d of line L_i consists of S_{i+1} parallel machines. The service times of the S_i parallel

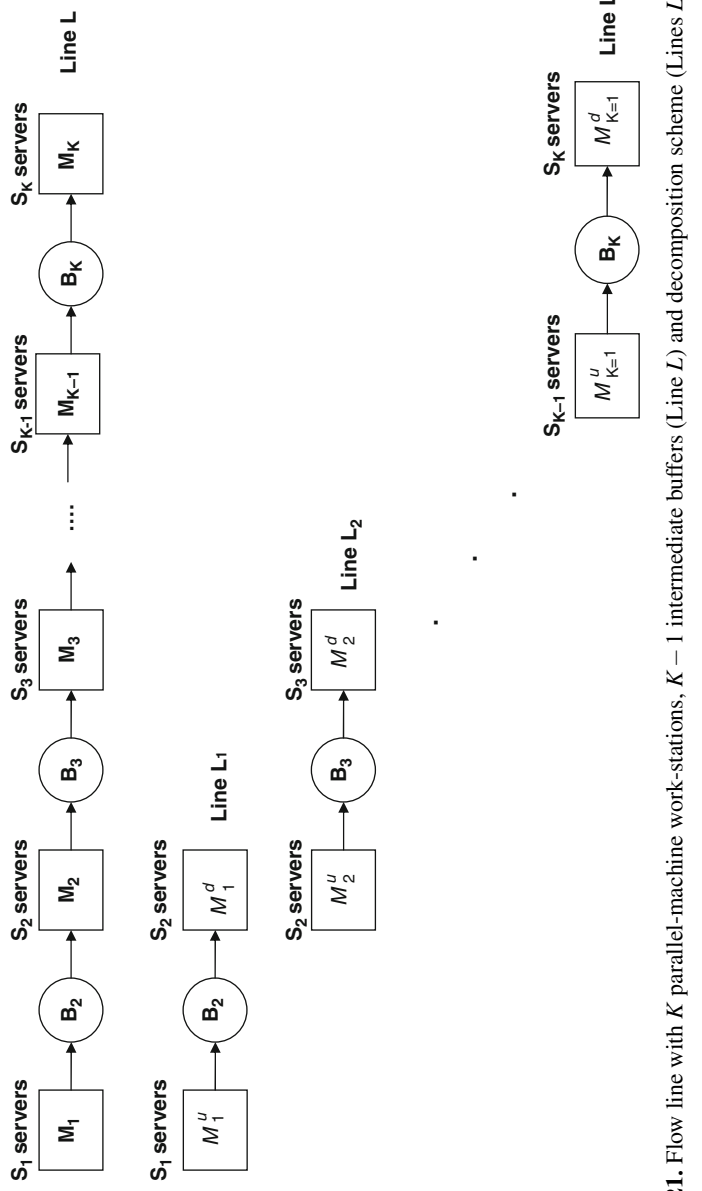


Fig. 2.21. Flow line with K parallel-machine work-stations, $K-1$ intermediate buffers (Line L) and decomposition scheme (Lines $L_1, \ldots, L_{K-1}$)

{**Step 1: Initialization**}

for $i = 1$ **to** $K-1$ **do**
  $\mu_i^u = \mu_i$
  $\mu_i^d = \mu_{i+1}$
  ε = small positive number for terminating condition
end for

{**Step 2: Calculate** μ_i^u **and** μ_j^d}

for $i = 2$ to $K-1$ **do**
  Calculate μ_i^u using equation (2.116)
  Evaluate the two-work-station, one buffer sub-line L_{i-1}, using the most recent values of μ_{i-1}^u and μ_{i-1}^d in the algorithm presented in sub-section 2.5.1.
end for
for $i = 2$ to $K-1$ **do**
  $j = K - i$
  Calculate μ_j^d using equation (2.117)
  Evaluate the two-work-station, one buffer sub-line L_{i+1}, using the most recent values of μ_{i+1}^u and μ_{i+1}^d in the algorithm presented in sub-section 2.5.1.
end for

{**Step 3: Terminating Conditions**}

if $\left|X_i^L - X_1^L\right| < \varepsilon,\ i = 2,\ldots,K-1$ **then**
  GOTO Step 4
else
  GOTO Step 2
end if

{**Step 4: Output Results**}

$X = X_i^L, i = 1,\ldots,K-1$

Fig. 2.22. Decomposition algorithm

machines of pseudo work-station M_i^u are exponentially distributed with mean $\frac{1}{\mu_i^u}$, while the service times of the S_{i+1} parallel machines of pseudo work-station M_i^d are also exponentially distributed with mean $\frac{1}{\mu_i^d}$, $i = 1,\ldots,K-1$. Also, define p_c^i to be the steady-state probability of state $y = (c)$ for sub-line L_i, where $c = 0,\ldots,C_i$. Defining $C_i = S_i + S_{i+1} + B_i$ and taking into account equations (2.100) and (2.101), the mean production rate of work-stations M_{i-1}^d and M_i^u is given by the following formulae:

$$X_{i-1}^d = S_i\mu_{i-1}^d \sum_{c=0}^{c=C_{i-1}} p_c^{i-1} - \mu_{i-1}^d \sum_{c=0}^{c=S_i-1} (S_i - c)p_c^{i-1},\ \ i = 2,\ldots,K \tag{2.104}$$

$$X_i^u = S_i\mu_i^u \sum_{c=0}^{c=C_i} p_c^i - \mu_i^u \sum_{c=S_{i+1}+B_i+1}^{c=C_i} (c - S_{i+1} - B_i)p_c^i,\ \ i = 1,\ldots,K-1. \tag{2.105}$$

Rewriting equations (2.104) and (2.105):

$$X_{i-1}^{d} = \mu_{i-1}^{d} \left(S_i \sum_{c=0}^{c=C_{i-1}} p_c^{i-1} - \sum_{c=0}^{c=S_i-1} (S_i - c) p_c^{i-1} \right) \tag{2.106}$$

and

$$X_i^{u} = \mu_i^{u} \left(S_i \sum_{c=0}^{c=C_i} p_c^{i} \quad - \sum_{c=S_{i+1}+B_i+1}^{c=C_i} (c - S_{i+1} - B_i) p_c^{i} \right). \tag{2.107}$$

It is straightforward to show that:

$$\sum_{c=0}^{c=C_i} p_c^{i} = \sum_{c=0}^{c=C_{i-1}} p_c^{i-1} = 1. \tag{2.108}$$

The blocking probability p_i^{bl} of each virtual two-work-station, one-buffer sub-line L_i is given by (derivation is omitted and the reader is addressed to Diamantidis, Papadopoulos and Heavey, 2006):

$$p_i^{bl} = \sum_{c=S_{i+1}+B_i+1}^{c=C_i} (c - S_{i+1} - B_i) p_c^{i}, \;\; i = 1, \ldots, K-1 \tag{2.109}$$

whereas the starvation probability p_{i-1}^{st} of each virtual two-work-station, one-buffer sub-line L_{i-1} is given by (derivation is omitted and the reader is addressed to Diamantidis, Papadopoulos and Heavey, 2006):

$$p_{i-1}^{st} = \sum_{c=0}^{c=S_i-1} (S_i - c) p_c^{i-1}, \;\; i = 2, \ldots, K. \tag{2.110}$$

Substituting equations (2.108) and (2.109) into equation (2.107), the mean production rate of the upstream work-station M_i^u is:

$$X_i^{u} = \mu_i^{u} (S_i - p_i^{bl}). \tag{2.111}$$

Similarly, substituting equations (2.108) and (2.110) into equation (2.106), the mean production rate of the downstream work-station M_{i-1}^d is:

$$X_{i-1}^{d} = \mu_{i-1}^{d} (S_i - p_{i-1}^{st}). \tag{2.112}$$

The mean production rate of work-station i, X_i, of the original production line L is given by:

$$X_i = \mu_i (S_i - p_{i-1}^{st} - p_i^{bl}). \tag{2.113}$$

Calculating probabilities p_i^{bl} and p_{i-1}^{s} from equations (2.111) and (2.112), respectively, one obtains:

$$p_i^{bl} = S_i - \frac{X_i^{u}}{\mu_i^{u}} \tag{2.114}$$

and

$$p_{i-1}^{st} = S_i - \frac{X_{i-1}^d}{\mu_{i-1}^d}. \tag{2.115}$$

Substituting equations (2.114) and (2.115) into equation (2.113) and taking into account conservation of flow in equation (2.103), the following two equations for calculating μ_i^u and μ_i^d can be derived:

$$\mu_i^u = \frac{1}{\frac{1}{\mu_i} + \frac{S_i}{X_{i-1}} - \frac{1}{\mu_{i-1}^d}}, \quad i = 2, ..., K-1 \tag{2.116}$$

$$\mu_i^d = \frac{1}{\frac{1}{\mu_{i+1}} + \frac{S_{i+1}}{X_{i+1}} - \frac{1}{\mu_{i+1}^u}}, \quad i = K-2, ..., 1. \tag{2.117}$$

Finally, because the virtual work-station M_1^u corresponds to the input work-station M_1 and the virtual work-station M_{K-1}^d corresponds to the output machine M_K of the original line L, the following boundary conditions are used:

$$\mu_1^u = \mu_1 \text{ and } \mu_{K-1}^d = \mu_K. \tag{2.118}$$

2.5.5 The decomposition algorithm

Using the above derived equations, a decomposition algorithm shown in Figure 2.22 was developed. The ε value used in all the numerical examples given here was 0.00001.

2.5.6 Numerical results

In order to evaluate the performance and the accuracy of the proposed decomposition algorithm, several numerical experiments have been conducted by Diamantidis, Papadopoulos and Heavey (2006) for various configurations of production lines with parallel machines at each work-station. Here, a few representative sample numerical results are given. First, results for short lines of up to 7 stations are presented and compared to published results. Then, to illustrate the efficiency of the solution method, sample results for long production lines are presented.

Comparison with published exact results—Short lines

In Diamantidis, Papadopoulos and Heavey (2006), results for short lines with up to 7 work-stations were compared against those reported in Hillier and So (1989, 1995, 1996). Hillier and So applied exact Markovian analysis to calculate the throughput of small production lines with up to 7 work-stations in series. Here, in Table 2.12 and Table 2.13 sample results are given for lines with 5 stations, unbalanced lines (processing rates of machines at different stations are not the same but the processing rates of machines at any station are the same) and 3, 5 and 7 stations, balanced lines (all machines in all stations have the same processing rate) with different

Table 2.12. Comparison of results with Hillier and So (1996) – 5 work-stations

s	μ	X_{DECO}	X_{HS96}	% Error	Time
(1,1,1,1,4)	(2.0876, 2.7624, 3.0120, 3.4013, 0.2831)	1.0192	1.0240	0.469	0.01
(4,1,1,1,1)	(0.2831, 3.4013, 3.0120, 2.7624, 2.0876)	1.0192	1.0210	0.176	0.01
(1,1,1,4,1)	(2.0920, 2.8248, 3.2786, 0.2894, 2.4509)	0.9965	1.0120	1.532	0.01
(1,1,4,1,1)	(2.1739, 3.1347, 0.2905, 3.1347, 2,1739)	0.9913	1.0100	1.851	0.01
(2,1,1,1,3)	(0.6877, 2.9585, 2.9673, 3.1250, 0.3920)	0.9890	0.9790	1.021	0.01
(1,2,1,1,3)	(2.0533, 0.7710, 3.0581, 3.1250, 0.3892)	0.9745	0.9730	0.154	0.01
(1,1,2,3,1)	(1.9569, 2.6881, 0.7987, 3.2467, 0.3910)	0.9667	0.9690	0.237	0.01
(1,2,1,3,1)	(2.0408, 0.7686, 3.2154, 0.4103, 2.1691)	0.9514	0.9610	0.999	0.01
(1,1,1,1,6)	(3.0211, 4.0000, 4.4444, 5.1020, 0.2501)	1.4149	1.4130	0.134	0.01
(6,1,1,1,1)	(0.2501, 5.1020, 4.4444, 4.0000, 3.0211)	1.4149	1.4090	0.419	0.01

Table 2.13. Comparison of results with Hillier (1995) – 3, 5 and 7 work-stations

K	**s**	X_{DECO}	X_{HS95}	%Error	Time
3	(15,16,16)	13.5500	13.5400	0.074	0.05
3	(1,2,2)	0.8846	0.8873	0.304	0.01
3	(30,32,30)	27.7800	27.6800	0.361	0.27
5	(1,1,2,1,1)	0.5541	0.5638	1.720	0.02
5	(1,2,1,2,1)	0.6677	0.6637	0.603	0.02
5	(1,2,2,2,2)	0.8716	0.8752	0.411	0.02
7	(1,1,2,1,2,1,1,)	0.5575	0.5613	0.677	0.05
7	(1,2,1,2,1,2,1)	0.6334	0.6320	0.222	0.05

servers allocation per station and zero buffer levels for all the intermediate buffers, respectively.

In all tables, columns labeled by vectors $\mathbf{s} = (S_1, \ldots, S_K)$, and $\mu = (\mu_1, \ldots, \mu_K)$ denote, respectively, the server allocation and the mean service rate allocation at the respective work-stations of a production line with N work-stations. The column labeled X_{DECO} gives the estimated mean production rate using the proposed decomposition algorithm by Diamantidis, Papadopoulos and Heavey (2006), while the column labeled X_{HSXX} ($XX = 95, 96$) is the published results given in Hillier and So (1995, 1996). The percentage error between the results obtained from decomposition and those reported in Hillier (1995, 1996) is computed using the following formula:

$$\% \text{ Error} = \frac{|X_{\text{DECO}} - X_{\text{HSXX}}|}{X_{\text{HSXX}}} \times 100\%, \tag{2.119}$$

where XX denotes results reported in year 19XX.

Table 2.12 presents numerical results obtained for a production line consisting of 5 work-stations with all the buffer capacities equal to zero. Column 4 presents the mean production rate reported in Hillier (1996), (X_{HS96}). Column 5 (% Error) gives the percentage error between the results obtained from decomposition and those reported in Hillier (1996). Column 6 gives the time (in seconds) taken by

decomposition to estimate the throughput. The decomposition algorithm was run on a Pentium III at 450MHz with 256MB RAM.

Table 2.13 presents numerical results for production lines where the number of work-stations K are 3, 5 and 7. It is also assumed that the processing rates of all machines at each work-station are equal to 1 and that the buffer level for all the intermediate buffers is 0. The first column (K) gives the number of work-stations, whereas the fourth column gives the estimated throughput reported in Hillier (1995), (X_{HS95}). The fifth and sixth columns present the percentage error and the required time by decomposition to obtain the results, respectively.

Comparison with simulation results—Long lines

The decomposition algorithm proposed in Diamantidis, Papadopoulos and Heavey (2006) has also been used to estimate the throughput of large balanced and unbalanced production lines (up to 1000 and even more work-stations). This algorithm was developed and coded by Dr. Alexandros Diamantidis in C++. To our knowledge, there is no exact analytical method that can estimate the throughput of such large lines. Diamantidis, Papadopoulos and Heavey (2006) applied their algorithm and solved a large system with 1000 work-stations, each with 3 servers and 999 intermediate buffers each of buffer size equal to 10 buffer slots in approximately 50 minutes on a Pentium IV computer. For comparison purposes, Diamantidis, Papadopoulos and Heavey (2006) developed a simulation model in eM-Plant (`http://www.ugs.com/products/tecnomatix/plant_design/em_plant.shtml`) in order to compare the results obtained from the decomposition algorithm for large production lines. Recall that this decomposition algorithm for $K = 2$ stations specifies the two-station line, with parallel machines at each work-station and an intermediate buffer,which is solved exactly and the throughput of the system is calculated.

S_i, and $\mu_i, i = 1,\ldots,K$, denote the number of parallel machines at work-station i and the service rate of each one of the S_i parallel machines at work-station i, respectively. $B_i, i = 2,\ldots,K$, denote the storage capacity of the buffer located in front of work-station i, $i = 2,\ldots,K$.

Table 2.14 presents configurations for 12 sample production lines varying from 10 to 120 stations in steps of 10. The results given for each example in Table 2.15 are: (i) throughput of the system calculated using simulation (X_{SIM}); (ii) throughput of the system calculated using the decomposition algorithm (X_{DECO}); (iii) average inventory for the system calculated using simulation ($\bar{c}_{\text{SIM}}$); (iv) average inventory for the system calculated using the decomposition algorithm ($\bar{c}_{\text{DECO}}$). Table 2.15 also presents 95% confidence intervals calculated for X_{SIM} and $\bar{c}_{\text{SIM}}$. The % Error for X was calculated using equation (2.119) with the % Error for $\bar{c}$ calculated similarly. The computing time, in seconds, to execute the decomposition algorithm is given in the two columns labeled "X-Time" and "$\bar{c}$-Time," respectively. These columns give the time to calculate the throughput and the average inventory, respectively. The computing time, in seconds, to execute the simulation experiments is given in the last column. The simulation model was run for 20,000 units/customers before statistics

Table 2.14. Sample configurations for long lines

Production Line 1	Production Line 2	Production Line 3
$K = 10$ $S_i = 2, i = 1,\ldots,4$ $S_5 = S_6 = 3$ $S_i = 2, i = 7,\ldots,10$ $B_i = 3, i = 1,\ldots,9$ $\mu_i = 1, i = 1,\ldots,10$	$K = 20$ $S_i = 3, i = 1,\ldots,7$ $S_j = 4, j = 8,\ldots,15$ $S_k = 3, k = 16,\ldots,20$ $B_i = 4, i = 1,\ldots,19,$ $\mu_i = 1, i = 1,\ldots,20$	$K = 30,$ $S_i = 3, i = 1,\ldots,10$ $S_j = 4, j = 11,\ldots,20$ $S_k = 3, k = 21,\ldots,30$ $B_i = 4, i = 1,\ldots,29$ $\mu_i =$ $1, i = 1,\ldots,30$
Production Line 4	**Production Line 5**	**Production Line 6**
$K = 40$ $S_i = 2, i = 1,\ldots,15$ $S_j = 3, j = 16,\ldots,25$ $S_k = 2, k = 26,\ldots,40$ $B_i = 3, i = 1,\ldots,49$ $\mu_i = 1, i = 1,\ldots,50$	$K = 50$ $S_i = 2, i = 1,\ldots,10$ $S_j = 1, j = 11,\ldots,40$ $S_k = 2, k = 41,\ldots,50$ $B_i = 2, i = 1,\ldots,49$ $\mu_i = 1, i = 1,\ldots,50$	$K = 60$ $S_i = 4, i = 1,\ldots,20$ $S_j = 5, j = 21,\ldots,40$ $S_k = 4, k = 41,\ldots,60$ $B_i = 4, i = 1,\ldots,59$ $\mu_i = 1, i = 1,\ldots,60$
Production Line 7	**Production Line 8**	**Production Line 9**
$K = 70$ $S_i = 3, i = 1,\ldots,25$ $S_j = 2, j = 26,\ldots,45$ $S_k = 3, k = 46,\ldots,70$ $B_i = 2, i = 1,\ldots,69$ $\mu_i = 1, i = 1,\ldots,70$	$K = 80$ $S_i = 2, i = 1,\ldots,30$ $S_j = 3, j = 31,\ldots,50$ $S_k = 2, k = 51,\ldots,80$ $B_i = 2, i = 1,\ldots,79$ $\mu_i = 1, i = 1,\ldots,80$	$K = 90$ $S_i = 2, i = 1,\ldots,30$ $S_j = 3, j = 31,\ldots,60$ $S_k = 2, k = 61,\ldots,90$ $B_i = 2, i = 1,\ldots,89$ $\mu_i = 1, i = 1,\ldots,90$
Production Line 10	**Production Line 11**	**Production Line 12**
$K = 100$ $S_i = 2, i = 1,\ldots,40$ $S_k = 3, k = 41,\ldots,60$ $S_m = 2, m = 61,...,100$ $B_i = 2, i = 1,\ldots,99$ $\mu_i = 1, i = 1,\ldots100$	$K = 110$ $S_i = 5, i = 1,\ldots,40$ $S_j = 6, j = 41,\ldots,70$ $S_k = 5, i = 71,\ldots,110$ $B_i = 4, i = 1,\ldots,110$ $\mu_i = 1, i = 1,\ldots,110$	$K = 120$ $S_i = 3, i = 1,\ldots,50$ $S_j = 4, i = 51,\ldots,70$ $S_k = 3, i = 71,\ldots,120$ $B_i = 3, i = 1,\ldots,120$ $\mu_i = 1, i = 1,\ldots,120$

were collected. The batch means method was used to collect 30 independent samples within a single run. A batch size of 5000 units/customers was used.

From examination of Table 2.15 it can be observed that the maximum error for the throughput is 1.72%. The accuracy of the decomposition algorithm for average inventory is not as good with a maximum error of 16.27% observable in Table 2.15. The convergence of the algorthm was found, for the majority of cases, to be very fast. However, the convergence speed can vary considerably and is system dependent, as can be observed in Table 2.15 where Line # 9 and 10 took approximately 50% of the time it took to obtain results using simulation. For all the results of the decomposition algorithm, a Pentium III at 450MHz with 256MB RAM was used. The simulation experiments were carried out on a Pentium IV at 2992MHz with 1000MB of RAM.

Table 2.15. Sample numerical results for long lines

Line #	Throughput				Average Inventory				Decomposition Time		Simulation
	X_{SIM}	X_{DECO}	% Error	95% CI	$\bar{c}_{SIM}$	$\bar{c}_{DECO}$	% Error	95% CI	X-Time	$\bar{c}$-Time	Time
1	1.5242	1.5423	1.1739	1.5198−1.5285	32.95679	33.4985	1.6171	30.7799−35.1336	0.09	0.39	87.46
2	2.3884	2.3808	0.3183	2.3805−2.3961	87.02922	77.5526	12.2196	83.4917−90.5666	0.61	0.66	136.83
3	2.3315	2.3533	0.9269	2.3402−2.3506	143.4842	139.6802	2.7234	138.9421−148.0263	0.22	0.22	183.85
4	1.4387	1.4447	0.4184	1.4353−1.4419	135.6619	130.001	4.3545	131.2453−140.0785	54.65	56.35	252.26
5	0.6092	0.6036	0.9335	0.6081−0.6102	101.4165	116.9982	13.3179	97.5978−105.2352	0.98	0.99	303.57
6	3.1274	3.1662	1.2250	3.1204−3.1343	347.8995	321.6019	8.1771	340.8268−354.9722	1.43	1.26	366.95
7	1.3440	1.3412	0.2071	1.3407−1.3472	214.3427	255.9992	16.2721	208.7912−219.8942	13.46	1.27	437.96
8	1.3311	1.3389	0.5814	1.3283−1.3339	239.463	215.59	11.0733	233.5951−245.3308	42.24	62.47	477.1
9	1.3319	1.339	0.5311	1.329 −1.3347	262.4053	230.581	13.8018	256.2628−268.5477	244.69	249.64	562.71
10	1.3241	1.3366	0.9315	1.3212−1.327	279.8781	274.1059	2.1058	273.5344−286.2217	260.68	266.6	616.71
11	3.9448	4.0066	1.5418	3.9357−3.9538	768.0851	701.1589	9.5451	757.5761−778.5941	2.64	1.27	704.81
12	2.1926	2.2309	1.7175	2.1883−2.1968	529.9708	503.4035	5.2775	521.2414−538.7001	2.97	3.02	752.33

Table 2.16. Configurations for longer lines

Production Line 13	Production Line 14	Production Line 15
$K = 200$	$K = 300$	$K = 400$
$S_i = 3, i = 1,\ldots,50$	$S_i = 3, i = 1,\ldots,50$	$S_i = 2, i = 1,\ldots,100$
$S_i = 2, i = 51,\ldots,100$	$S_i = 1, i = 51,\ldots,100$	$S_i = 3, i = 101,\ldots,200$
$S_i = 4, i = 101,\ldots,150$	$S_i = 4, i = 101,\ldots,150$	$S_i = 4, i = 201,\ldots,300$
$S_i = 1, i = 151,\ldots,200$	$S_i = 2, i = 151,\ldots,200$	$S_i = 1, i = 301,\ldots,400$
$B_i = 2,\ i = 1,\ldots,199$	$S_i = 3, i = 201,\ldots,300$	$B_i = 2,\ i = 1,\ldots,399$
$\mu_i = 1, i = 1,\ldots,200$	$B_i = 3,\ i = 1,\ldots,299$	$\mu_i = 1, i = 1,\ldots,400$
	$\mu_i = 1, i = 1,\ldots,300$	
Production Line 16	**Production Line 17**	**Production Line 18**
$K = 500$	$K = 600$	$K = 700$
$S_i = 3, i = 1,\ldots,150$	$S_i = 2, i = 1,\ldots,200$	$S_i = 4, i = 1,\ldots,250$
$S_i = 4, i = 151,\ldots,300$	$S_i = 4, i = 201,\ldots,400$	$S_i = 2, i = 251,\ldots,450$
$S_i = 2, i = 301,\ldots,500$	$S_i = 3, i = 401,\ldots,600$	$S_i = 4, i = 451,\ldots,700$
$B_i = 3,\ i = 1,\ldots,499$	$B_i = 2,\ i = 1,\ldots,599$	$B_i = 3,\ i = 1,\ldots,699$
$\mu_i = 1, i = 1,\ldots,500$	$\mu_i = 1, i = 1,\ldots,600$	$\mu_i = 1, i = 1,\ldots,700$
Production Line 19	**Production Line 20**	**Production Line 21**
$K = 800$	$K = 900$	$K = 1000$
$S_i = 4, i = 1,\ldots,200$	$S_i = 4, i = 1,\ldots,400$	$S_i = 4, i = 1,\ldots,400$
$S_i = 3, i = 201,\ldots,400$	$S_i = 3, i = 401,\ldots,500$	$S_i = 3, i = 401,\ldots,600$
$S_i = 2, i = 401,\ldots,600,$	$S_i = 4, i = 501,\ldots,900$	$S_i = 4, i = 601,\ldots,1000$
$S_i = 3, i = 601,\ldots,800$	$B_i = 3,\ i = 1,\ldots,899$	$B_i = 3,\ i = 1,\ldots,999$
$B_i = 3,\ i = 1,\ldots,799$	$\mu_i = 1, i = 1,\ldots,900$	$\mu_i = 1, i = 1,\ldots,1000$
$\mu_i = 1, i = 1,\ldots,800$		

In Table 2.17 throughput results are given for the configurations shown in Table 2.16 which include even longer production lines (with $K = 200(100)1000$ workstations). As it can be seen from Table 2.17, the maximum error was found to be 2.5%. The parameters **s**, number of parallel stations, **n**, the buffer sizes, and μ, the mean service times, vary arbitrarily, so as to illustrate the versatility of the algorithm. Run times, in seconds, for the decomposition algorithm and the simulation model are given in the last two columns. From Table 2.17 it can be noted that X_{DECO} lies outside the 95% CI for Line # 16–21. In general it was found that X_{DECO} was outside the 95% CI for configurations with % Error greater than 1.00.

The numerical results presented in Table 2.15 and in Table 2.17 indicate that the decomposition algorithm is very accurate. The average percentage error of the throughput obtained from the proposed decomposition algorithm and simulation for lines with up to 100 stations is less than 1%, whereas the results presented for lines with up to 1000 stations indicate that the percentage error is less than 2.5%. The convergence of the algorithm is very fast and reliable. Diamantidis, Papadopoulos and Heavey (2006) claim that they have not found a case in which the algorithm does not converge.

Table 2.17. Numerical results for longer lines

Line #	X_{SIM}	X_{DECO}	% Error	95% CI	Decomposition Time	Simulation Time
13	0.6047	0.6014	0.5444	0.6036–0.6058	7.58	1340.72
14	0.6620	0.6678	0.8688	0.6610–0.6631	15.05	1478.35
15	0.6001	0.6004	0.0546	0.5994–0.6008	131.77	3104.08
16	1.3984	1.4287	2.1651	1.3963–1.4012	369.16	3673.55
17	1.3089	1.3335	1.8763	1.3073–1.3105	347.08	3188.14
18	1.3988	1.4287	2.1411	1.3956–1.4012	361.09	4618.73
19	1.3982	1.4287	2.1790	1.3960–1.4003	571.55	5626.19
20	2.1802	2.2292	2.2470	2.1766–2.1838	239.97	5887.03
21	2.1759	2.2287	2.4276	2.1722–2.1796	804.60	7162.45

The above algorithm is available at the website associated with this book, with abreviated name DECO-2. As noted above if K, the number of stations is equal to 2, the exact numerical solution to the two-station production line with identical perfectly reliable parallel machines at each station may be obtained. In addition, if the number of machines at each station is set equal to 1, the authors have shown that the results obtained for large production lines with single machine stations, exponential service times, perfectly reliable machines and intermediate buffers of finite capacity replicate the decomposition equations originally given by Gershwin (1994).

2.6 Simulation Modeling

Simulation of production lines is a powerful tool in obtaining the performance measures where analytical methods are either difficult or impossible to use. In the past, simulation was often considered to be an expensive and time consuming approach to the solution of system type problems. However, with the increase of computer power and the availability of special-purpose simulation languages, such constraints are less severe. Usually, in simulation studies of production lines what is technically involved is Monte Carlo simulation because of the inherent stochastic variability of these systems. The combination of simulation studies with analytical studies is probably the way of the future in the design of production lines.

Ideally, the production line analyst and designer requires a discrete event simulation package with Monte Carlo simulation capabilities, graphical and other output reporting facilities together with a relatively easy method for the statistical assessment of results. In simulation modeling, the modeler must specify very carefully how the production line is meant to operate and the various disciplines and rules which are involved. The basis of discrete event simulation is that the system state at any time t is stored and that the state only changes when a particular event occurs. The specification of the state of the system depends on the detail required by the modeler in respect to the performance characteristics of the system. In all simulation studies there is a need to consider the time horizon of the process under investigation.

Short term system performance analysis requires that data be taken from the simulation during a short time horizon. On the other hand, steady-state simulation models are appropriate for the analysis of systems which in theory could run indefinitely. Usually, in production line modeling the modeler is interested in steady-state behavior of the system by which time the precise initial state of the system has little impact. It is normal in these cases to have a "warm-up" period before recording data for the calculation of the steady-state behavior. Graphical output of the performance parameters of the system can be extremely useful in determining when the warm-up period is ended. In some simulation packages it is possible to specify in advance the time to allow the system to settle down. The appropriateness of this time may in fact be checked from the results of the simulation. Finally, it is usual to place confidence limits on the values of the permormance parameters of the system, based on certain assumptions, and such limits may usually be incorporated into modern simulation models.

As an illustration of the power of simulation, Arena, a simulation software package available to the authors, has been used to model a system of the type depicted in Figure 2.23, with the following characteristics:

- The line consists of $K = 4$ work-stations with identical parallel machines at each work-station. The number of machines at station $i, i = 1, \ldots, 4$ are 3, 2, 2, 3, respectively. Service or processing times are exponentially distributed and the mean service rates of the identical machines at each work-station are $\mu_i = 1$, $i = 1, \ldots, 4$. Thus, the probability that a service is completed on a machine at station i in a time interval Δt is $\mu_i \Delta t$.
- All machines are assumed to be perfectly reliable.
- The interstation buffers $B_i, i = 2, 3, 4$ have capacities of 4, 2, 4, respectively.
- All products produced conform to specifications.
- Transfer times between stations and buffers and between buffers and stations are considered negligible.
- Any particular machine may be blocked after service due to the finite capacities of the buffers excluding the last set of parallel machines.
- Arrangements are made to ensure that the first station is always busy, i.e., never starved or it is saturated; any machine at any other station may be starved.
- No machine of any of the group of machines at any particular station is given priority in relation to being unblocked when unblocking occurs. Likewise in relation to the resumption of production at a work-station following the removal of starvation.

The following performance measures of the line were determined:

- Throughput (jobs exiting from the production line per unit time).
- Utilization of each work-station (the limit of the time average of the number of busy machines over time divided by the total number of machines in the station).
- Average buffer level for each intermediate buffer.
- Average work-in-process, $\overline{WIP}$, excluding the buffer before the first station.
- Average job waiting time at each of the intermediate buffers.

Fig. 2.23. A production line with four stations with parallel reliable machines at each station and three intermediate buffers

To ensure saturation of the first station of the line, the capacity of the queue in front of the first station was set at twenty (20) units and batches arrive at sizes of five (5), and the arrival rate into the system, λ, was taken to be greater than $3\mu_1$. Finally, a warm-up period of 100 minutes was specified to ensure that steady-state conditions were obtained. (More details of this Arena simulation model and numerical results are given in Appendix E.)

It may be of interest to discuss briefly possible extensions of the relatively simple simulation model developed above. Clearly, the number of stations and the number of parallel machines at each station could be modified. Buffer capacities could be changed, and the restriction of identical machines at each work-station could be removed which would impact on the scheduling rules. Unreliable machines could be incorporated. Processing times, repair times, times to failures and transport times from stations to buffers and from buffers to stations may be modeled using the Monte Carlo simulation. There is no need to confine the distributions to be exponential as Erlang, phase-type, normal, uniform or deterministic distributions may be modeled using Monte Carlo simulation. The production of defective items at any station could be incorporated into the model via a feedback mechanism for rework if necessary to earlier stations.

2.7 General Comment

The reader will note that the Markovian method gives an exact evaluation of the state probabilities from which an exact evaluation of the throughput may be obtained, whereas the other three methods (aggregation, decomposition and expansion) give approximate results for the throughput. Clearly, the limitation in using Markovian methods is the smaller number of states which may be handled effectively in contrast to the approximate methods at the expense of accuracy. The existence of parallel-machine stations introduced complications on two fronts, viz., significantly increasing the number of states and the difficulties of such elements being incorporated in the early classical decomposition methods. The expansion method gave early hope that some of the problems with the modeling of parallel machines would be reduced, but issues relating to accuracy of results still remain. It appears that the results reported in Section 2.5 overcome a number of the problems associated with earlier attempts to use decomposition methods in systems with parallel structures. In principle, it is possible to derive the states for Markovian type of solution for such systems, but unfortunately this would be computationally very inefficient for any realistic production lines with parallel-machine stations. All methods presented are, of course, in principle applicable to unreliable as well as prefectly reliable machines.

2.8 Related Bibliography

Researchers and practitioners alike should appreciate that there is a very rich literature applicable to the general area of the performance evaluation of production lines. Here an attempt is made to highlight the main strands of thought in this area. It should

be noted that work intitiated for application areas quite diverse from manufacturing has been found to be fruitful when applied to the analysis of production lines. Cases in point are analyses originally oriented toward computer performance modeling and communication networks have subsequently given insights into problems germane to the analysis of production lines. Basically, the mathematical underlined theory of production line analysis is queueing theory, in particular, queueing networks with blocking. An exceptional reference in this area is the book by Perros (1994). However, care must be taken to ensure the validity of the model of the production system in that for example blocking in a communication system tends to occur before the service starts, whereas in a production line blocking occurs after the service has been completed. What is offered below is a classification by the authors of what they believe to be significant and somewhat distinct areas of research of value in the analysis of the performance of production lines. As far as Markovian analysis approaches are concerned, some five areas of work have been identified and are described below. It should be noted that there is nothing unique about this categorization and indeed some authorities might well question the relative influence accorded to the work of particular researchers.

The exact solution of small production lines was initiated by Hunt (1956) followed by Buzacott (1972), Gershwin and Berman (1981) and Gershwin and Schick (1983), among others. Solutions were obtained for two/three stations with limited inter-station buffers, and methods of solution used included matrix recursive and matrix geometric methods applied to the underlying Markov chains. Initially, exponentially distributed processing times were only considered, but the work of Buzacott and Kostelski (1987) extended the distribution of processing times to phase-type distributions.

Altiok (1997) in a seminal work summarized and developed the earlier work by Altiok and Ranjan (1989), Buzacott and Kostelski (1987) and Perros (1994), among others, and brought phase-type modeling to its present position. Exact analysis of small-scale production lines with any type of processing and repair time distributions may be undertaken. Arising out of Altiok's work it is possible to perform approximate analysis of larger systems with any general distributions of processing and repair times by the approximation of these distributions by phase-type distributions by matching their first two or three moments.

When faced with the analysis of relatively large production lines, there is a need for efficient computational procedures due essentially to the large number of associated states of the underlying Markov chain of such systems. Hillier and Boling (1967) developed a numerical approach for solving reliable exponential and Erlang production lines. Papadopoulos and O'Kelly (1989), Papadopoulos, Heavey and O'Kelly (1989, 1990) and Heavey, Papadopoulos and Browne (1993) further developed this work by producing efficient numerical algorithms for generating the transition matrices for reliable and unreliable production lines with exponential and Erlang processing and repair time distributions and efficient solution methods. Further extensions in this area are included in the book of Altiok (1997), as noted above, by using the mixed generalized exponential distributions (phase-type distributions). The algorithm included at the website associated with this book with abbreviated

name MARKOV for the generation of the transition matrix and the solution of the associated steady-state Markov equations is based on the work of Papadopoulos, O'Kelly and Heavey.

In contrast to continuous parameter discrete state Markov process analysis of production lines, Muth (1984) introduced the concept of the holding time model where the focus is on the three possible states of each station, viz., the station is idle, busy or blocked. Alkaff and Muth (1987) extended Muth's model to solve K-station production lines with an arbitrary number of stations. A major advantage of the holding time model is that the number of separate non-linear equations that have to be solved is significantly reduced in comparison to the Markovian situation. The price paid for this reduction is the need to solve non-linear equations that are of the form of a fixed point problem. Holding time models can accommodate Erlang and phase-type distributions more readily than can Markovian methods again because of the reduction in the number of states. However, the holding time model cannot accommodate intermediate buffers of non-zero capacity.

It is of great assistance to designers to have simple closed form formulae to determine the throughput of production lines. A number of such formulae have been developed based on insights from general queueing theory, considerations and sometimes curve-fitting. Hunt (1956) was an early developer of such a closed form expression. Makino (1964), Muth (1984) and Muth (1987) offered formulas for the exponential and two-phase Erlang and distribution-free cases with no intermediate buffers between successive stations. Freeman (1968) and Anderson and Moodie (1969) obtained empirical formulas for utilization of the production line, based on regression analysis of various sets of simulation data. Knott (1970) offered a formula based on theoretical and intuitive reasoning. Blumenfeld (1990) extended Muth's formula for throughput of a production line with variable processing times and buffers of finite capacities. Haydon (1973) dealt with approximations in his Ph.D. dissertation and he provided approximate throughput formulae that perform quite satisfactorily. Papadopoulos (1996) using Muth's holding time model developed an analytical formula for the throughput of a K-station production line with no intermediate buffers and exponential processing times which may be different at the various stations of the line. A particular simpler formula was developed for the balanced line.

The limitations of seeking exact solutions to production line problems are related to problems arising from the number of states of such systems and the difficulties associated with a numerical approach. Thus, there has been considerable interest in developing approximate methods of analysis. Most approximate methods are based on decomposition and an esssential element of this approach is that the sub-lines used have exact solutions. Decomposition methods are approximations as the sub-lines used are simpler than the original line and the equations used to develop the parameters may also be approximated to facilitate the numerical analysis. Earlier work on decomposition methods include Zimmern (1956), Sevast'yanov (1962) and Hillier and Boling (1967). Queueing networks with blocking were decomposed by Caseau and Pujolle (1979), Takahashi et al. (1980), and Boxma and Konheim (1981). Altiok and Perros and their teams have made significant contributions by working on decomposition to solve large systems with exponential and phase-type distributed

processing times. This work is reported in papers including Altiok (1982), Perros and Altiok (1986), Jun and Perros (1987), Brandwajn and Jow (1988), Altiok (1989), Altiok and Ranjan (1989) and Gun and Makowski (1989). Excellent expositions of this work are given in the book of Altiok (1997). Gershwin (1987) in a well-known article offered an efficient decomposition method for the approximate evaluation of tandem queues with finite intermediate buffers and blocking. Dallery, David and Xie (1988) improved the convergence of Gershwin's algorithm. An excellent review of flow line models is given in Dallery and Gershwin (1992), and the decomposition approaches are treated in detail in the book of Gershwin (1994). Decomposition models of various types of manufacturing systems are also included in the seminal work of Buzacott and Shanthikumar (1993). Dallery and Frein (1993) classified the various decomposition methods for solving production lines into one of three classes according to the sets of decomposition equations used by the various authors.

Many papers concerned with the analysis of production lines have reported results on simulation studies. It is virtually impossible to give an adequate review of such papers from the perspective of the use of simulation method in the determination of production line performance. Nevertheless, there are a number of books and research papers which are certainly worth further detailed study and investigation by analysts specifically interested in simulation. These include the books by Altiok and Melamed (2001), Kouikoglou and Phillis (2001), Law and Kelton (2000), Guide to Arena Standard Edition by Systems Modeling Corporation (1999), Banks et al. (1999), Kelton et al. (1998), Benson (1996), Khoshnevis (1994), Papadopoulos et al. (1993), Brateley et al. (1987), Pritsker (1986) and Fishman (1973), among others.

Decomposition techniques have also been applied not only to manufacturing systems but also to computer systems (see Perros, 1994 and many references therein), among others, and to more general manufacturing systems. For example, Tempelmeier and Burger (2001a) examined non-homogeneous asynchronous flow production systems and presented an analytical approximation for the performance of such systems. They assumed generally distributed stochastic processing times as well as breakdowns and imperfect production. The proposed approximation was based on the decomposition of an K-station-line into $(K-1)$ two-station-lines that were analyzed using a $GI/G/1/N_{\max}$ queueing model. They also presented numerical comparisons with exact and simulation results which indicated that the procedure provides accurate results. In Kuhn (2003) an analytical approach was given for performance evaluation of an automated flow line system which considers the dependency between the production and the repair system. The proposed model and solution approach may be used in the initial design phase as well as during a redesign process in order to evaluate various configurations of the production and repair systems.

Tolio and Matta (1998) presented an elegant decomposition approach for the performance evaluation of automated flow lines with multiple failure modes. The decomposition block that was used in their analysis was solved exactly by a method that is independent of the buffer size. An extension of the decomposition approach for the performance evaluation of a flow line with linear flow of material and two part types was presented by Nemec (1999). A different efficient decomposition analysis

for serial flow lines with two part types, deterministic identical processing times and multiple failure modes was proposed by Colledani, Matta and Tolio (2003). Flow lines with single machine work-stations and non-linear flow of material are examined in Helber (1999), where a detailed analysis of flow lines with split and merge operations is presented, Gershwin et al. (2001), Helber and Mehrtens (2003), Tan (2001) and Helber and Jusic (2004).

Tolio, Matta and Gershwin (2002) presented an analytical method for the performance evaluation of production lines with two unreliable machines and one intermediate buffer of finite capacity. Each machine can fail in more than one way.

Levantesi, Matta and Tolio (1999a, b) developed an efficient decomposition method for the performance evaluation of production lines with exponential processing times, multiple failure modes and finite buffer capacities. The different types of failures are distributed according to different exponential distributions as are the times to repair.

Levantesi, Matta and Tolio (2003) provided an approximate analytical method for the performance evaluation of asynchronous production lines with deterministic processing times, multiple failure modes and finite buffer capacity. In their analysis, the authors approximated the discrete flow of parts by a continuous flow of material.

Literature is relatively scarce on the analysis of flow lines with multiple identical parallel-machine work-stations. Friedman (1965) presented a reduction method that reduces a queueing system with parallel-machine work-stations to corresponding problems for a system of fewer stages. It was also assumed that for any sequence of customer arrival times, the time spent in the system was independent of the order of stages. Forestier (1980) examined automated flow lines where each station consists of two parallel machines. Dubois and Forestier (1982) considered similar systems using Markovian analysis. Iyama and Ito (1987) considered a flow line where some work-stations have different numbers of parallel machines and unequal service rates. They presented the effects of server allocation on the maximum average production rate by using a Markovian model.

The exact solution of the two-station production line with the first station saturated is based on queueing theory and a good exposition of this analysis may be found in the book by Perros (1994), in the book by Buzacott and Shanthikumar (1993) and in the book by Neuts (1981), among others. Details of the generation of the associated conservative matrix, A, and a method for the calculation of the throughput of such systems based on the elements of A are given in the paper by Vidalis and Papadopoulos (2001). In addition, the recursive relationship for the number of states of a general production line with $K \geq 2$ parallel stations is derived in this paper. With respect to the approximate solutions of larger systems there are a few research studies of interest. These include the book by Buzacott and Shanthikumar (1993), where an iterative procedure is applied to calculate the throughput of the long line using the solution to the two-station line described above. In the paper by Jain and MacGregor Smith (1994), the expansion method was used to approximate the performance measures of each parallel station of the production line. In this paper, apart from the series system, merge and splitting topologies were also analyzed.

Another paper of major interest is that by Patchong and Willaeys (2001), where each set of parallel machines is replaced by an equivalent single machine at each station of the production line. Then, existing methods may be used to derive the performance measures of the original system. A similar approximation method was applied by Jeong and Kim (1999) for performance analysis of assembly/disassembly systems with parallel machine stations. Earlier, Caseau and Pujolle (1979) derived the throughput of some specialized telecommunication models using repeated trials methods.

In van Dijk and van der Wal (1989) computationally attractive lower and upper bounds for finite multi-server exponential tandem queues were presented. A proof of the bounds and related monotonicity results were also presented, which were based on Markov reward theory. Gosavi and MacGregor Smith (1995) developed computationally efficient bounds and approximations for the performance measures of series parallel queueing networks. They approximated analytically the throughput of a system with two tandem exponential queues and extended their analysis to elementary merge and split queuing networks.

Ancelin and Semery (1987) described a method that replaces each parallel-machine work-station by an equivalent single machine work-station. The processing rate of the equivalent work-station equals the sum of the processing rates of all parallel machines in the work-station. The failure rate and repair rate of the equivalent work-station are given by a formula which incorporates the failure and repair parameters of the parallel machines in the work-station.

Burman (1995) applied a similar method that replaces each parallel server work-station by a single equivalent work-station for the case of continuous flow of material. The author assumed that the equivalent work-station has a maximum processing rate which equals the sum of the processing rates of the parallel machines. The failure and repair parameters of the equivalent work-station are calculated by using the assumption that all parallel machines at a specific work-station operate independently.

Cheah and MacGregor Smith (1994) showed how a M/G/C/C state dependent queuing model is embodied into the modeling of large-scale facilities where the blocking phenomenon can be or cannot be controlled. They also presented an approximation technique based on the expansion method to incorporate the M/G/C/C queuing models into series, merge and splitting topologies of production lines. Jain and MacGregor Smith (1994) presented an analytical technique to calculate system performance measures of M/M/C/K queuing networks. They analyzed series, merge and splitting topologies and in addition they explored the optimal order of the M/M/C/K servers in such systems.

In Diamantidis, Papadopoulos and Heavey (2006), a flow line with parallel machines at each work-station is analyzed via the decomposition method which was presented in Section 2.5.2. The proposed approach differs from those of Ancelin and Semery (1987), Burman (1995), Jeong and Kim (1999) and Patchong and Willaeys (2001), in that each parallel-machine work-station is not replaced by an equivalent work-station. That is, the decomposition approach is applied directly to each one of the parallel machines for each work-station without using replacement techniques.

It is expected that this direct approach will provide more accurate results than do the replacement techniques.

Regarding the non-linear flow lines, the material in the text is based on the paper by Diamantidis, Papadopoulos and Vidalis (2004). An excellent exposition of this area is given in the book by Helber (1999) in which various non-linear flow models are analyzed. Models using continuous variables are given by Tan (2001) and by Helber and Mehrtens (2003), in which times to failure and repair are exponentially distributed. Other relevant papers include Gershwin (1991), Jeong and Kim (1998), Yu and Bricker (1993), Ammar and Gershwin (1989), Dallery, Liu and Towsley (1994), Di Mascolo et al. (1991), Frein et al. (1996), Helber (1998), among others.

References

1. Alkaff, A. and Muth, E.J. (1987), The throughput rate of multistation production lines with stochastic servers, *Probability in the Engineering and Informational Sciences*, Vol. 1, pp. 309–326.
2. Altiok, T. (1982), Approximate analysis of exponential tandem queues with blocking, *European Journal of Operational Research*, Vol. 11, pp. 390–398.
3. Altiok, T. (1989), Approximate analysis of queues in series with phase-type service times and blocking, *Operations Research*, Vol. 37, pp. 601–610.
4. Altiok, T. (1997), *Performance Analysis of Manufacturing Systems*, Springer.
5. Altiok, T. and Ranjan, R. (1989), Analysis of production lines with general service times and finite buffers: A two-node decomposition approach, *Engineering Costs and Production Economics*, Vol. 17, pp. 155–165.
6. Altiok, T. and Melamed, B. (2001), *Simulation Modeling and Analysis with Arena*, Cyber Research, Inc. and Enterprise Technology Solutions, Inc.
7. Ammar, M. and Gershwin, S.B. (1989), Equivalence relations in queueing models of fork/join queueing networks with blocking, *Performance Evaluation*, Vol. 10, pp. 233–245.
8. Ancelin, B. and Semery, A. (1987), Calcul de la productivite d'une ligne integree de fabrication: CALIF, une methode analytique industrielle, *RAIRO, APII*, Vol. 21, No. 3, pp. 209–238.
9. Anderson, D.R. and Moodie, C.L. (1969), Optimal buffer storage capacity in production line systems, *International Journal of Production Research*, Vol. 7, pp. 233–240.
10. Banks, J., Carson, J.S., and Nelson, B.L. (1999), *Discrete-Event System Simulation*, 2nd Edition, Prentice Hall.
11. Benson, D. (1996), *Simulation Modeling and Optimization using PROMODEL*, PROMODEL Corporation, Orem, Utah.
12. Blumenfeld, D.E. (1990), A simple formula for estimating throughput of serial production lines with variable processing times and limited buffer capacity, *International Journal of Production Research*, Vol. 28, No. 6, pp. 1163–1182.
13. Boxma, O.J. and Konheim, A.G. (1981), Approximate analysis of exponential queueing systems with blocking, *Acta Informatica*, Vol. 15, pp. 19–66.
14. Brandwajn, A. and Jow, Y.L. (1988), An approximation method for tandem queues with blocking, *Operations Research*, Vol. 36, pp. 73–83.
15. Brateley, P., Fox, B.L., and Schrage, L.E. (1987), *A Guide to Simulation*, Springer-Verlag.

16. Burman, M.H. (1995), *New Results in Flow Line Analysis*, Ph.D. Thesis, Massachusetts Institute of Technology (M.I.T.).
17. Buzacott, J.A. (1972), The effect of station breakdowns and random processing times on the capacity of flow lines with in-process storage, *AIIE Transactions*, Vol. 4, No. 4, pp. 308–313.
18. Buzacott, J.A. and Kostelski, D. (1987), Matrix-geometric and recursive algorithm solution of a two-stage unreliable flow line, *IIE Transactions*, Vol. 19, pp. 429–438.
19. Buzacott, J.A. and Shanthikumar, J.G. (1993), *Stochastic Models of Manufacturing Systems*, Prentice Hall.
20. Caseau, P. and Pujolle, G. (1979), Throughput capacity of a sequence of transfer lines with blocking due to finite waiting room, *IEEE Transactions Software Eng.,* SE-5, pp. 631–642.
21. Cheah, J.Y. and MacGregor Smith, J. (1994), Generalized $M/G/C/C$ state dependent queueing models and pedestrial traffic flows, *Queueing Systems*, Vol. 15, pp. 365–386.
22. Colledani, M., Matta, A., and Tolio, T. (2003), Performance evaluation of production lines with finite buffer capacity producing two different products, *In Proceedings of the Fourth Aegean International Conference on Analysis of Manufacturing Systems*, pp. 231–240, Samos Island, Greece.
23. Dallery, Y., David, R., and Xie, X.L. (1988), An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers, *IIE Transactions*, Vol. 20, pp. 280–283.
24. Dallery, Y. and Frein, Y. (1993), On decomposition methods for tandem queueing networks with blocking, *Operations Research*, Vol. 41, No. 2, pp. 386–399.
25. Dallery, Y. and Gershwin, S.B. (1992), Manufacturing flow line systems: A review of models and analytical results, *Queueing Systems Theory and Applications*, Vol. 12, pp. 3–94.
26. Dallery, Y., Liu, Z., and Towsley, D. (1994), Equivalence, reversibility, symmetry and concavity properties in fork-join queuing networks with blocking, *Journal of the Association for Computing Machinery*, Vol. 41, No. 5, pp. 903–942.
27. Diamantidis, A.C., Papadopoulos, C.T., and Heavey, C. (2006), Approximate analysis of serial flow lines with multiple parallel-machine stations, *IIE Transactions*, Vol. 39, No. 4, pp. 361–375.
28. Diamantidis, A.C., Papadopoulos, C.T., and Vidalis, M.I. (2004), Exact analysis of a discrete material three station one buffer merge system with unreliable machines, *International Journal of Production Research*, Vol. 42, No. 4, pp. 651–675.
29. Di Mascolo, M., David, R., and Dallery, Y. (1991), Modeling and analysis of assembly systems with unreliable machines and finite buffers, *IIE Transactions*, Vol. 23, No. 4, pp. 315–330.
30. Dubois, D. and Forestier, J.P. (1982), Productivité et en-cours moyens d'un ensemble de deux machines separées par une zone de stockage, *RAIRO Automatique*, Vol. 16, No. 2, pp. 105–132.
31. Fishman, G.S. (1973), *Concepts and Methods in Discrete Event Simulation*, Wiley.
32. Forestier, J.P. (1980), Modelisation stochastique et comportement asymptotique d'un systéme automatise de production, *RAIRO Automatique*, Vol. 14, No. 2, pp. 127–143.
33. Freeman, D.R. (1968), A general line balancing model, *Proceedings of the 19th Annual Conference, AIIE*, Tampa, Florida, Norcross, Georgia: American Institute of Industrial Engineers, pp. 230–235.
34. Frein, Y., Commault, C., and Dallery, Y. (1996), Modeling and analysis of closed-loop production lines with unreliable machines and finite buffers, *IIE Transactions*, Vol. 28, pp. 545–554.

35. Friedman, H.D. (1965), Reduction methods for tandem queueing systems, *Operations Research*, Vol. 13, pp. 121–131.
36. Gershwin, S.B. (1987), An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking, *Operations Research*, Vol. 35, pp. 291–305.
37. Gershwin, S.B. (1991), Assembly/disassembly systems: An efficient decomposition algorithm for tree structured networks, *IIE Transactions*, Vol. 23, No. 4, pp. 302–314.
38. Gershwin, S.B. (1994), *Manufacturing Systems Engineering*, Prentice Hall.
39. Gershwin, S.B. and Berman, O. (1981), Analysis of transfer lines consisting of two unreliable machines with random processing times and finite storage buffers, *AIIE Transactions*, Vol. 13, No. 1, pp. 2–11.
40. Gershwin, S.B., Maggio, N., Matta, A., Tolio, T., and Werner, L. (2001), Analysis of loop networks by decomposition, *In Proceedings of the Third Aegean International Conference on Analysis and Modeling of Manufacturing Systems*, pp. 239–248, Tinos Island, Greece.
41. Gershwin, S.B. and Schick, I.C. (1983), Modeling and analysis of three-stage transfer lines with unreliable machines and finite buffers, *Operations Research*, Vol. 31, No. 2, pp. 354–380.
42. Gosavi, H.D. and MacGregor Smith, J. (1995), Asymptotic bounds of throughput in series-parallel queueing networks, *Computers & Operations Research*, Vol. 22, No. 10, pp. 1057–1073.
43. Gun, L. and Makowski, A. (1989), An approximation method for general tandem queueing systems subject to blocking, *Proc. 1st Int'l Workshop on Queueing Networks with Blocking*, Raleigh, NC.
44. Haydon, B.J. (1973), The behaviour of systems of finite queues, *Ph.D. Thesis*, The University of New South Wales, Kensington, New South Wales, Australia.
45. Heavey, C., Papadopoulos, H.T., and Browne, J. (1993), The throughput rate of multistation unreliable production lines, *European Journal of Operational Research*, Vol. 68, pp. 69–89.
46. Helber, S. (1998), Decomposition of unreliable assembly/disassembly networks with limited buffer capacity and random processing times, *European Journal of Operational Research*, Vol. 109, No. 1, pp. 24–42.
47. Helber, S. (1999), *Performance Analysis of Flow Lines with Non-Linear Flow of Material*, In Volume 473 of Lecture Notes in Economics and Mathematical Systems, Springer-Verlag.
48. Helber, S. and Jusic, H. (2004), A new decomposition approach for non-cyclic continuous material flow lines with a merging flow of material, *Annals of Operations Research*, Vol. 124, No. 1–4, pp. 117–139.
49. Helber, S. and Mehrtens, N. (2003), Exact analysis of a continuous material merge system with limited buffer capacity and three stations, In S.B. Gershwin, Y. Dallery, C.T. Papadopoulos, and J. MacGregor Smith, Editors, *Analysis and Modeling of Manufacturing Systems*, pp. 85–121, Kluwer Academic Publishers.
50. Hillier, F.S. and Boling, R.W. (1967), Finite queues in series with exponential or Erlang service times – A numerical approach, *Operations Research*, Vol. 15, pp. 286–303.
51. Hillier, F.S. and So, K.C. (1989), The assignment of extra servers to stations in tandem queueing systems with small or no buffer, *Performance Evaluation*, Vol. 10, pp. 219–231.
52. Hillier, F.S. and So, K.C. (1995), On the optimal design of tandem queueing systems with finite buffers, *Queueing Systems*, Vol. 21, pp. 245–266.

53. Hillier, F.S. and So, K.C. (1996), On the simultaneous optimization of server and work allocations in production line systems with variable processing times, *Operations Research*, Vol. 44, No. 3, pp. 435–443.
54. Hunt, G.C. (1956), Sequential arrays of waiting lines, *Operations Research*, Vol. 4, pp. 674–683.
55. Iyama, T. and Ito, S. (1987), The maximum production rate for an unbalanced multi-server flow line system with finite buffer storage, *International Journal of Production Research*, Vol. 25, No. 8, pp. 1157–1170.
56. Jain, S. and Smith, J.M. (1994), Open finite queueing networks with $M/M/C/K$ parallel servers, *Computers & Operations Research*, Vol. 21, No. 3, pp. 297–317.
57. Jeong, K.-C. and Kim, Y.-D. (1998), Performance analysis of assembly/disassembly systems with unreliable machines and random processing times, *IIE Transactions*, Vol. 30, pp. 41–53.
58. Jeong, K.-C. and Kim, Y.-D. (1999), An approximation method for performance analysis of assembly/disassembly systems with parallel-machine stations, *IIE Transactions*, Vol. 31, pp. 391–394.
59. Jun, K. and Perros, H.G. (1987), An approximate analysis of open tandem queueing networks with blocking and general service times, *Computer Science*, NCSU, Raleigh, NC.
60. Kelton, W.D., Sadowski, R.P., and Sadowski, D.A. (1998), *Simulation with Arena*, McGraw-Hill.
61. Kerbache, L. (1984), Analysis of open finite queueing networks, Ph.D. thesis, Department of Industrial Engineering and Operations Research, University of Massachusetts, Amherst, MA.
62. Kerbache, L. and MacGregor Smith, J. (1987), The generalized expansion method for open finite queueing networks, *European Journal of Operational Research*, Vol. 32, pp. 448–461.
63. Khoshnevis, B. (1994), *Discrete Systems Simulation*, McGraw-Hill.
64. Kleinrock, L. (1975), *Queueing Systems, Vol. I*, Wiley.
65. Knott, A.D. (1970), The inefficiency of a series of work-stations – a simple formula, *International Journal of Production Research*, Vol. 8, pp. 109–119.
66. Kouikoglou, V.S. and Phillis, Y.A. (2001), *Hybrid Simulation Models of Production Networks*, Kluwer Academic Publishers.
67. Kuhn, H. (2003), Analysis of automated flow line systems with repair crew interference, In S.B. Gershwin, Y. Dallery, C.T. Papadopoulos, and J. MacGregor Smith, Editors, *Analysis and Modeling of Manufacturing Systems*, pp. 155–179, Kluwer Academic Publishers.
68. Labetoulle, J. and Pujolle, G. (1980), Isolation method in a network of queues, *IEEE Transact. Software Eng.*, Vol. SE-6, No. 4.
69. Law, A.M. and Kelton, W.D. (2000), *Simulation Modeling and Analysis*, McGraw-Hill.
70. Levantesi, R., Matta, A., and Tolio, T. (1999a), Exponential two machine lines with multiple failure modes and finite buffer capacity, *In Proceedings of the Second Aegean International Conference on Analysis and Modeling of Manufacturing Systems*, pp. 103–112, Tinos Island, Greece.
71. Levantesi, R., Matta, A., and Tolio, T. (1999b), A decomposition method for the performance evaluation of production lines with random processing times, multiple failure modes and finite buffer capacity, *In Proceedings of the Second Aegean International Conference on Analysis and Modeling of Manufacturing Systems*, pp. 113–122, Tinos Island, Greece.

72. Levantesi, R., Matta, A., and Tolio, T. (2003), Performance evaluation of continuous production lines with machines having different processing times and multiple failure modes, *Performance Evaluation*, Vol. 51, pp. 247–268.
73. Lim, J.-T., Meerkov, S.M., and Top, F. (1990), Homogeneous, asymptotically reliable serial production lines: Theory and a case study, *IEEE Transactions on Automatic Control*, Vol. 35, No. 5, pp. 524–534.
74. Magazine, M.J. and Stecke, K.E. (1996), Throughput for production lines with serial work-stations and parallel service facilities, *Performance Evaluation*, Vol. 25, No. 3, pp. 211–232.
75. Makino, T. (1964), On the mean passage time concerning some queueing problems of the tandem type, *J. Opns Res. Soc. Japan*, Vol. 7, pp. 17–47.
76. Muth, E.J. (1984), Stochastic processes and their network representations associated with a production line queueing model, *European Journal of Operational Research*, Vol. 15, pp. 63–83.
77. Muth, E.J. (1987), An update on analytical models of serial transfer lines, *Research Report*, No. 87-15, Gainesville, FL: Department of Industrial and Systems Engineering, University of Florida.
78. Nemec, J.E. (1999), *Diffusion and Decomposition Approximations of Stochastic Models of Multiclass Processing Networks*, Ph.D. Thesis, Massachusetts Institute of Technology (M.I.T.).
79. Neuts, M.F. (1981), *Matrix-Geometric Solutions in Stochastic Models – An Algorithmic Approach*, The Johns Hopkins University Press.
80. Onvural, R.O. and Perros, H.G. (1986), On equivalencies of blocking mechanisms in queueing networks with blocking, *Operations Research Letters*, Vol. 5, No. 6, pp. 293–298.
81. Papadopoulos, H.T. (1989), Mathematical modelling of reliable production lines, Ph.D. Thesis, University College Galway, Ireland.
82. Papadopoulos, H.T. (1996), An analytic formula for the mean throughput of K-station production lines with no intermediate buffers, *European Journal of Operational Research*, Vol. 91, pp. 481–494.
83. Papadopoulos, H.T., Heavey, C., and Browne, J. (1993), *Queueing Theory in Manufacturing Systems Analysis and Design*, Chapman & Hall.
84. Papadopoulos, H.T., Heavey, C., and O'Kelly, M.E.J. (1989) Throughput rate of multistation reliable production lines with inter-station buffers (I) Exponential Case, *Computers in Industry*, Vol. 13, No. 3, pp. 229–244.
85. Papadopoulos, H.T., Heavey, C., and O'Kelly, M.E.J. (1990), Throughput rate of multistation reliable production lines with inter station buffers: II Erlang case, *Computers in Industry*, Vol. 13, No. 4, pp. 317–335.
86. Papadopoulos, H.T. and O'Kelly, M.E.J. (1989) A recursive algorithm for generating the transition matrices of multistation series production lines, *Computers in Industry*, Vol. 12, pp. 227–240.
87. Patchong, A. and Willaeys, D. (2001), Modeling and analysis of an unreliable flow line composed of parallel-machine stages, *IIE Transactions*, Vol. 33, pp. 559–568.
88. Perros, H. (1994), *Queueing Networks with Blocking - Exact and Approximate Solutions*, Oxford University Press.
89. Perros, H.G. and Altiok, T. (1986), Approximate analysis of open networks of queues with blocking, tandem configurations, *IEEE Trans. Soft. Eng.*, SE-12, pp. 450–461.
90. Pritsker, A.A.B. (1986), *Introduction to Simulation and Slam II*, 3rd Edition, Halsted.

91. Sevast'yanov, B.A. (1962) Influence of stage bin capacity on the average standstill time of a production line, *Theory of Probability and its Applications*, Vol. 7, No. 4, pp. 429–438.
92. Systems Modeling Corporation (1999), *Guide to Arena Standard Edition*, Sewickley.
93. Takahashi, Y., Miyahara, H., and Hasegawa, T. (1980), An approximation method for open restricted queueing network, *Operations Research*, Vol. 28, pp. 594–602.
94. Tan, B. (2001), A three-station merge system with unreliable stations and a shared buffer, *Mathematical and Computer Modeling*, Vol. 33, pp. 1011–1026.
95. Tempelmeier, H. and Burger, M. (2001), Performance evaluation of unbalanced flow lines with general distributed processing times, failures and imperfect production, *IIE Transactions*, Vol. 33, No. 4, pp. 293–302.
96. Tolio, T. and Matta, A. (1998), A method for performance evaluation of automated flow lines, *Annals of the CIRP*, Vol. 47, No. 1, pp. 373–376.
97. Tolio, T., Matta, A., and Gershwin, S.B. (2002), Analysis of two-machine lines with multiple failure modes, *IIE Transactions*, Vol. 34, pp. 51–62.
98. Van Dijk, N. and van der Wal, J. (1989), Simple bounds and monotonicity results for finite multi-server exponential tandem queues, *Queueing Systems Theory and Applications*, Vol. 4, pp. 1–16.
99. Vidalis, M.I. and Papadopoulos, H.T. (2001), A recursive algorithm for generating the transition matrices of multistation multiserver exponential queueing networks, *Computers & Operations Research*, Vol. 28, No. 9, pp. 853–883.
100. Yu, K.-Y.C. and Bricker, D.L. (1993), Analysis of a Markov chain model of a multistage manufacturing system with inspection, rejection, and rework, *IIE Transactions*, Vol. 25, No. 1, pp. 109–112.
101. Zimmern, B. (1956), Etudes de la propagation des arrets aleatoires dans les chaines de production, *Review Statistical Applications*, Vol. 4, pp. 85–104.

3

The Design of Production Lines

3.1 Introduction

This chapter is essentially a prelude to the rest of the text and its objective is to assist the reader to understand the main initial design problems that arise with production lines. It is important for the reader to clarify the context of any design problem related to any production line, e.g., is it a green fields situation, a modification of an existing production line to enhance performance or the adaptation of an existing line to produce products not produced already?

Once the strategic decision to use a production line to manufacture the products has been made, the design of the line must be undertaken. To remind the reader of the complexities involved, in Figure 3.1, an example of a relatively complex production line, adapted from Li (2003), is shown.

In Figure 3.1, the rectangles represent machines and the circles represent buffers. Although it is traditional in analysis to indicate machines by rectangles, it must be remembered that associated with many such machines are human operators and that human operators may in fact form a separate work-station without any machines. Indeed, it is these human operators that add variability to the production line in that many machine processes are essentially deterministic in practice. Here, it is assumed that the ergonomic design of the systems is undertaken by relevant specialists while the physical requirements of the system are being fully specified by others. All these specialists are of course in a position to contribute to an understanding of the variability involved in production lines on an ongoing basis during the design process of the production line.

Quality is a major performance characteristic of modern manufacturing and in particular there are inspection and test stations embedded in production lines. The precise arrangement for handling rework of defective material is generally dependent on the materials handling arrangements. Sometimes feedback is possible resulting in the reuse of the inspection and testing facilities whereas in other situations rework and further inspection are effectively performed off the main line. Either of these cases may be handled in most models.

C. T. Papadopoulos et al., *Analysis and Design of Discrete Part Production Lines*, Springer Optimization and Its Applications,
DOI: 10.1007/978-0-387-89494-2_3,

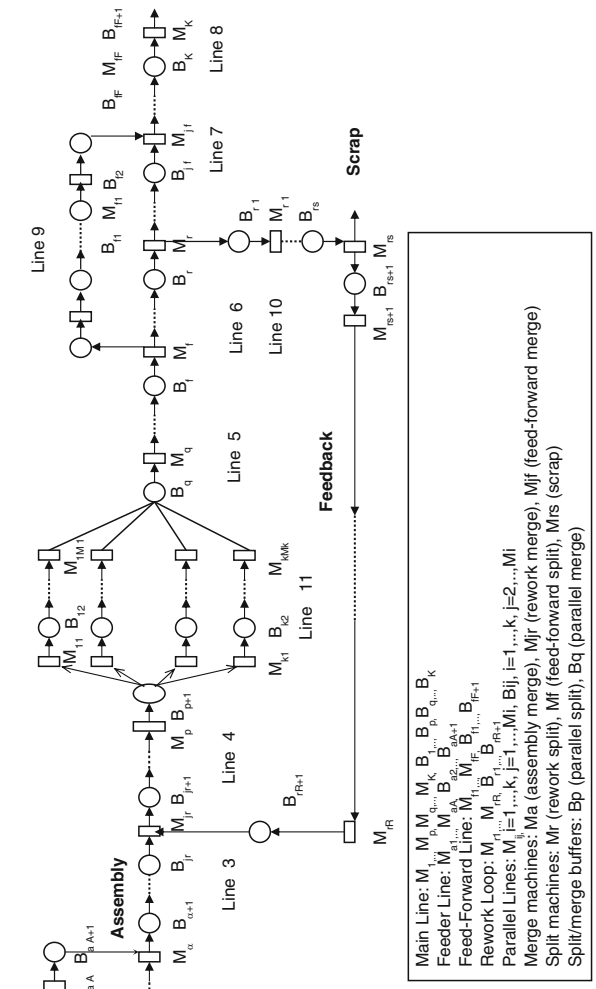


Fig. 3.1. A typical structure of a complex production line

As shown in Figure 3.1, there is one main production line (Line 1, Line 3, Line 4, Line 12, Line 5, Line 6, Line 7 and Line 8), with Line 12 being a parallel-machine line consisting of κ sub-lines, a feeder line (Line 2), a feed-forward line (Line 9) as well as a rework line/loop (Line 10 and Line 11). Machine M_a is an assembly merge machine, machine M_{jr} is a rework merge machine and machine M_{jf} is a feed-forward merge machine, while machine M_r is a rework split machine, machine M_f is a feed-forward split machine and machine M_{rs} is a scrap split machine. There are split/merge buffers associated with the parallel line (Line 12).

By design is meant the specification of some of the parameters (structure of the production system) to achieve a specific objective. The approach is quite different to the use of methods to evaluate the performance of a specified system which has been already discussed in Chapter 2.

In this chapter, it is assumed that the production processes at each machine are specified. To arrive at this situation may have involved considerable engineering work. In addition, the sequencing of the machines/layout of the production line has been determined. For the purposes of this chapter, the details of the transportation system between the machine stations are assumed to be given and the information and control systems are not of specific interest. Essentially, what is being said is that a flow diagram of type Figure 3.1 has been developed in outline form where the production rate of each individual machine, the details of the buffer sizes and the number of parallel machines have yet to be determined. Further details of the considerations involved may be found in Buzacott and Shanthikumar (1993), Altiok (1997) and Groover (2001), among others.

In general, there are three methods of increasing the throughput of an individual work-station: (a) increasing the production rate of an individual machine, (b) using machines in parallel, or (c) a combination of both. These involve technological and managerial choices. The design of production lines as understood here is confined to the following issues:

1. Work-load at each station: There are well-known design guidelines, discussed below in Chapter 4, which result in increased throughput of the line (units produced per unit time over the entire line). The application of these guidelines will specify the mean production rates of each of the work-stations. These design problems are referred to as *work-load allocation problems, WAP*. In such problems it is normal to assume fixed specified buffer sizes and single-machine work-stations.

   Readers will be aware that research results of interest to manufacturing systems designers may arise in work not specifically oriented towards manufacturing systems. This is particularly true in relation to the work allocation problem where a series of papers have developed quite strong results mainly using mathematical analysis. Interested readers are referred to the papers listed in Chapter 4.
2. Determination of the number of machines at each work-station: The use of parallel systems will affect the throughput of the line. The associated design problem is referred to as *the server allocation problem, SAP*, and is treated also in

Chapter 4. Normally, in such design problems it is assumed that there are fixed station specific buffer sizes between the parallel machine stations.

3. Specification of the sizes of the buffers: It is more usual to have machine or station specific buffers but occasionally common buffers for more than one machine or station are sometimes used. Such designs are referred to as *the buffer allocation problem, BAP*, which is the subject of Chapter 5.

   The design problem from the point of view of the systems engineer is as follows Given:

   - Fixed number of work-stations (K). This number is determined by technological, precedence and economic considerations. Servers at these K work-stations may consist of machines only, of human operatives only or of a feasible and necessary combination of these two types of resources.
   - Number of servers $S(S \geq K)$.
   - Total work-load of the line, normalized to K (time units).
   - Total number of buffer spaces (N).

   The design problem in general is to do the following meet a specified objective, usally expressed in throughput, work-in-process or cost terms:

   (i) Allocate the number of servers S over the given K stations; clearly there must be at least one server at each station;

   (ii) Allocate the normalized work-load to each of the given fixed K stations;

   (iii) Allocate the total number of buffer spaces N over the $K-1$ buffer storage areas. Usually, the buffer in front of the first station is assumed to be of adequate size (theoretically infinite) to accommodate the flow of work and these buffer spaces are not included in the N buffer spaces which are considered as a parameter of the design problem. Likewise, the storage spaces after the last (Kth) station are excluded from consideration leaving just $K-1$ storage areas among the K stations.

Needless to say, it is possible to consider the design problem of maximizing the throughput of production lines in which none of the following are specified a priori: the production rate at each station, the inter-station buffer sizes and the number of parallel servers at each station. This leads to a very general design problem with considerable computational complexities. In practice however, it is more usual, initially, to consider simpler design problems with two of the three decisions listed above already made, and these simpler design problems may be considered to be "pure" allocation problems.

It should be noted that usually the word 'allocation' has a very specific meaning. In the pure work-load allocation problem, the objective is to allocate a total capacity of K time units over K work-stations so as to maximize throughput given the machine specific buffers in the system. In the pure buffer allocation problem, the objective is to maximize throughput by allocating an overall buffer space of size N among the $K-1$ buffer locations, where each station has a fixed production rate. Finally, in the pure server allocation problem the total number of servers in the system is fixed

and the objective is to maximize throughput of the system by allocating an integer number of servers to each station given fixed station specific buffers.

The words 'work-stations' and 'machines' are used interchangeably in production line design problems. However, it should be noted that here 'machines' is a generic term which includes the following meanings: physical machines alone, operators alone or a combination of these two resources or more generally, servers.

Usually, designers are concerned with maximizing throughput. There are a few other possible objective functions which may be of interest. These include the minimization of average work-in-process, $\overline{WIP}$, having in mind current operations philosophies of lean production. In such models, a threshold throughput, X_0, must be achieved and $\overline{WIP}$ is minimized in the context of this achievement while satisfying other constraints in relation to buffer allocation, server allocation and work-load allocation. Finally, a more specific cost/financial objective function may be developed to include machining cost and buffer space and inventory holding costs.

The two performance measures mentioned above, viz., throughput, X, and average work-in-process, $\overline{WIP}$, may be characterized as efficiency and effectiveness performance measures, respectively. Increasing the throughput of the line is normally associated with increasing average WIP and vice versa. Usually, other measures of performance such as mean flow or production time, utilization of individual stations, often a favorite of earlier generations of production engineers and managers, etc., may be easily obtained from the computer results.

In production lines, machines may be considered to be reliable or unreliable. Unreliable machines have an associated reliability or survival curve from which the mean time to failure (MTTF) may be determined. Failed machines may be repaired in accordance with a repair time distribution from which the mean time to repair (MTTR) may be determined.

The processing time at a machine may be assumed to be deterministic or stochastic. If stochastic, the mean service time and its coefficient of variation may be determined from the associated processing time distribution. Often the exponential distribution is used, resulting in a coefficient of variation of 1. In practice, it has been observed that the coefficient of variation is less than 1 and thus a strict exponential distribution of processing times is inappropriate. However, phase type distributions (e.g., Coxian distribution with two phases) can be used to accommodate situations where the coefficient of variation of service time is less than 1 while retaining the analytical benefits of the exponential distribution.

The reader might note that the justification of any particular design of a manufacturing system raises complex issues, particularly in the case of systems which have some inherent flexibility. In the past, finished designs tended to be costed and evaluated on the basis of either meeting or not meeting a specified interest (hurdle) rate in a discounted cash flow analysis (dcf). Many criticisms have been leveled at this approach (see for example Noble and Tanchoco, 1993). As research in this area has progressed, the methodology for the concurrent evaluation of design performance and economic evaluation has been developed. In production line design, the full realization of this approach to design evaluation can only be achieved through the holistic integration of the work of the detailed engineering designers specifying

the outline of the initial system and the work of the system specialists involved in performance analysis. Chapter 7 is concerned with the costing of various designs.

3.2 Role of the Design Engineer

The design of a production line essentially involves the allocation of the following resources after decisions about the number of separate work-stations and the sequence of such work-stations have been made:

- Servers (operators and/or machines).
- Buffer slots/space between stations.
- Work-load at each station (expected service time of individual parts at each station).

Clearly, decisions in relation to resources have cost implications which must be taken into account in addition to the performance measures of the production line. From a purely production engineering and operations management point of view, there is considerable attraction to the concept of a symmetrically balanced production line which would be characterized by the following features:

- Servers with identical mean service rates.
- Same number of servers at each work-station.
- Identical inter-station buffer capacities.
- Identical expected service time of each part at each station (balanced work-load).

The reader might note that in the technical literature, generally, balanced production lines merely implies that mean service rates (number of servers at each station by identical individual mean service rate) are equal over all the stations. In fact, there are a number of algorithms and heuristics in existence which assume that a balanced work-load leads to the maximum throughput of a production line. The symmetrical balanced line described above has considerable intuitive appeal but very simple examples will show that if there is variability in service, a balanced line will not lead to the maximum throughput. For example, consider a balanced production line with $K = 4$ stations with exponentially distributed service/processing times at each station with equal mean service rates: $\mu_i = 1$, for $i = 1,\ldots,4$, and buffer sizes: $B_j = 2$, for $j = 2,3,4$. The throughput of such a line is 0.7007 compared with the higher throughput 0.7051 obtained by an unbalanced line with $\mu_1 = 1/1.069$, $\mu_2 = 1/0.931$, $\mu_3 = 1/0.931$ and $\mu_4 = 1/1.069$ (the summation of $1/\mu_i$'s is equal to $K = 4$) and buffer sizes again all equal to 2, i.e., the percentage increase in throughput is 0.63%. Here, consideration is given to the performance measure throughput only. The engineering economist/operations manager should ask the question at what cost was this increase in throughput achieved? For instance: What about comparison between the mean work-in-process of the two systems? What was the cost of utilizing the servers with the specified service rates in each case? Are such servers in fact available? What utilization of each server was achieved? A conclusion from this simple case is that improvement in engineering/operations performance measures alone may not assist

in answering the real question which is how to achieve the minimum cost of production using a production line system of work organization. It is possible to debate that because many analytical studies have shown that the optimal throughput is not 'significantly' different from the throughput of a 'balanced' production line, it may not be worthwhile pursuing the optimal solution. However, it must be noted that generally production lines are developed for high volume and relatively long life and so a small improvement in throughput may have a significant economic advantage. Another criticism of analytical models is that they fail to capture the complexities of real-life systems. This is true. For example, variability in service times is often described by a phase-type distribution such as the exponential, Erlang or Coxian, whereas in practice the distribution of service times may follow a very different distribution. The same criticism can be made of simulation studies. In passing, it may be noted that in practice the coefficient of variation of service times has been found to be of the order 0.2 to 0.4. However, if an analytical representation of a proposed production line indicates that the performance of the system would improve if some imbalance was introduced, then the designer would be well advised to take this into account. It is unrealistic to expect at this stage of the development of analytical modeling, including simulation, that the designer can produce designs of production lines which are incapable of being improved. The viewpoint of this book is to give every possible assistance to the designer to investigate different design configurations and to arrive at a design that is feasible, economic and has an acceptable performance. After the implementation of the design, further improvement is generally possible by way of special studies, simulations and analytical work following actual experience in operation.

3.3 Improvability

A different design problem arises when modifications to an existing system are contemplated. A production line may for example not be achieving desired production levels due to a deterioration in service levels or to a changed product or product mix. Clearly, in such cases a total re-design and physical re-construction of the production line may not be justified. Using evaluative models it may be possible to determine the throughput with the parameters derived from measurements on the existing system and hopefully confirming its current performance. Such models might point to the existence of a bottleneck station through for instance the starving of a downstream station and/or the blocking of an upstream station and so the design effort could be concentrated on alleviating the bottleneck station. In other cases it may be possible to design for optimal throughput and to determine how far the existing system is from optimal in terms of such measures as work-load allocation, machine specific buffers and number of parallel machines at each work-station. Clearly, in all such cases there would be a concern to achieve maximum impact on the performance measure desired at minimum cost in re-designing the current system.

It should be noted that in practice, the concept of buffers has several meanings. For example, a buffer between two single-machine stations might be considered to

be in series or in shunt (parallel). The discipline for the series buffer would normally be First-In, First-Out (FIFO), whereas the discipline for the shunt buffer would be Last-In, First-Out (LIFO). Where a station with parallel machines is concerned, the buffer discipline can be quite complex, in that an idle machine may not be in a position to service a waiting unit due to the materials handling protocol. So, the usual assumption of queueing theory that an idle server would immediately serve a waiting job may be violated. Clearly, care should be taken by the analyst to ensure that the buffer protocols used in any modeling work are in accordance with the actual situation.

The reader might note that the design problems specified above are not the same as those problems faced by operations management in their quest for continuous improvement (KAIZAN). When issues of improving the performance of an existing system arise, the work of Meerkov and his colleagues is particularly relevant. Meerkov defines a production system to be improvable if the limited resources involved in its operation can be redistributed so that a performance measure is improved. It must be understood that in practice there may be constraints on the redistribution process and improvability as such may not reach the optimality achievable in a mathematical sense. Performance measures involved here could relate to throughput, work-in-process (WIP), workforce (WF) allocation and due-time performance. Details of improvement strategies may be found in Jacobs and Meerkov (1995a). The role of bottlenecks in production systems is well known and in the paper cited above, Jacobs and Meerkov, gave a very precise definition of a bottleneck machine or buffer, as follows.

Let $PI(\mu_1,\ldots,\mu_K,N_1,\ldots,N_K)$ be the performance index of interest, e.g., the throughput, the due-time performance, the workforce allocation, product quality, and so forth.

A production system is called improvable with respect to WIP if there exists a sequence $N_1^*,\ldots,N_K^*$ such that $\sum_{i=1}^K N_i^* = N$ and

$$PI(\mu_1,\ldots,\mu_K,N_1^*,\ldots,N_K^*) > PI(\mu_1,\ldots,\mu_K,N_1,\ldots,N_K),$$

where, $\sum_{i=1}^K N_i = N$.

A production system is called improvable with respect to workforce (WF) if there exists a sequence $\mu_1^*,\ldots,\mu_K^*$ such that $\prod_{i=1}^K \mu_i^* = \mu^*$ and

$$PI(\mu_1^*,\ldots,\mu_K^*,N_1,\ldots,N_K) > PI(\mu_1,\ldots,\mu_K,N_1,\ldots,N_K),$$

where, $\prod_{i=1}^K \mu_i = \mu^*$.

The reader will note that the second equation above is in product form and is a bound on the workforce (WF). The assignment of the workforce defines the production rate (machine operators) and the average up-time (repair personnel) of each machine. The available workforce can be assigned to the work-stations in accordance with the constraint given by the second equation. This constraint may be referred to as the machine efficiency constraint and changes in the allocation of resources within the production line are required to maintain this overall constraint. In contrast, the design problem in the earlier paragraphs of this chapter was formulated

using the work-load allocation, where the usual summation constraint was used, i.e., $\sum_{i=1}^{K} w_i = 1$.

A production system is called improvable with respect to WIP and WF simultaneously if there exist sequences $N_1^*,\ldots,N_K^*$ and $\mu_1^*,\ldots,\mu_K^*$ such that $\sum_{i=1}^{K} N_i^* = N$, $\prod_{i=1}^{K} \mu_i^* = \mu^*$ and

$$PI(\mu_1^*,\ldots,\mu_K^*,N_1^*,\ldots,N_K^*) > PI(\mu_1,\ldots,\mu_K,N_1,\ldots,N_K),$$

where, $\sum_{i=1}^{K} N_i = N$.

Machine i is the bottleneck machine if

$$\frac{\partial PI(\mu_1,\ldots,\mu_K,N_1,\ldots,N_K)}{\partial \mu_i} > \frac{\partial PI(\mu_1,\ldots,\mu_K,N_1,\ldots,N_K)}{\partial \mu_j}, \quad \forall j \neq i.$$

Buffer i is the bottleneck buffer if

$$PI(\mu_1,\ldots,\mu_K,N_1,\ldots,N_i+1,\ldots,N_K) > PI(\mu_1,\ldots,\mu_K,N_1,\ldots,N_j+1,\ldots,N_K), \quad \forall j \neq i.$$

Other definitions of bottlenecks exist, for example, the bottleneck machine is the machine with the lowest isolated production rate. A buffer is considered to be a bottleneck buffer if the expected size of the work-in-process (WIP) at that buffer is larger than the expected size of the work-in-process, $\overline{WIP}$, at the other buffers on the assumption that all buffer capacities are equal. More detailed examination of issues related to bottlenecks are covered in Goldratt and Cox (1986) in the context of the theory of constraints. As is clear from the precise definition given by Jacobs and Meerkov (1995), the bottleneck machine is not necessarily the machine with the lowest isolated mean production rate nor is the bottleneck buffer that buffer with the smallest capacity. The reader is referred to an interesting example given in the work by Jacobs and Meerkov (1995a). Further extensions in the general area on the topic of improvability are contained in the following papers: Jacobs and Meerkov (1995b) Kuo, Lim and Meerkov (1996), Chiang, Kuo and Meerkov (1998), Chiang, Kuo and Meerkov (2000), Li and Meerkov (2000), Li and Meerkov (2001), Chiang, Kuo, Lim and Meerkov (2000a) and Chiang, Kuo, Lim and Meerkov (2000b). Other references of interest include Enginarlar, Li and Meerkov (2003a) and Enginarlar, Li and Meerkov (2003b). Collectively, these papers contain a rich source of information to determine bottleneck stations and buffers and insightful design guidelines which would enhance the performance of existing systems and could also be used to check the appropriateness of systems design using the methods that are more germaine to the main stream of the methods proposed in this text.

In Chapters 4, 5, and 6, design problems of particular importance to production line designers are presented. The objectives of these chapters are to assist designers in the solution of practical problems using software available at the website associated with this text. Where possible, design guide rules are given with respect to specific situations. It must be understood that these guidelines were developed by researchers following, in most cases, extensive experimentation over a wide range of parameters. However, although useful, these guidelines must be treated with respect, particularly if applied to situations not covered by the original experimentation. In this regard, the

reader is advised to consult the original papers which are usually given in the relevant bibliography. Finally, the authors would urge the designer to carry out, using the software provided, a series of experiments, if at all possible, over the range of parameters of interest, so that the appropriateness of the set of the design guidelines may be tested. It should also be remembered that it is important to develop some experience of the relative accuracy of some of the algorithms being used by researchers generally in this area and that perhaps it is true to say that algorithms developed more recently tend to be more accurate and more efficient. Nevertheless, it is vital to be fully familiar with the assumptions of any particular model being used because although most models will give a result, the really important issue is how realistic is the result obtained when applied to the problem in hand.

References

1. Altiok, T. (1997), *Performance Analysis of Manufacturing Systems*, Springer-Verlag.
2. Buzacott, J.A. and Shanthikumar, J.G. (1993), *Stochastic Models of Manufacturing Systems*, Prentice Hall.
3. Chiang, S.-Y., Kuo, C.-T., Lim, J.-T., and Meerkov, S.M. (2000a), Improvability of assembly systems I: Problem formulation and performance evaluation, *Mathematical Problems in Engineering*, Vol. 6, pp. 321–357.
4. Chiang, S.-Y., Kuo, C.-T., Lim, J.-T., and Meerkov, S.M. (2000b), Improvability of assembly systems II: Improvability indicators and case study, *Mathematical Problems in Engineering*, Vol. 6, pp. 359–393.
5. Chiang, S.-Y., Kuo, C.-T., and Meerkov, S.M. (1998), Bottlenecks in Markovian production lines: A systems approach, *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 2, pp. 352–359.
6. Chiang, S.-Y., Kuo, C.-T., and Meerkov, S.M. (2000), DT-Bottlenecks in serial production lines: Theory and application, *IEEE Transactions on Robotics and Automation*, Vol. 16, No. 5, pp. 567–580.
7. Enginarlar, E., Li, J., and Meerkov, S.M. (2003a), How lean can lean be? The University of Michigan, Systems Science and Engineering Division, Department of Electrical Engineering and Computer Science, Report No. CGR-03-10, September 2003.
8. Enginarlar, E., Li, J., and Meerkov, S.M. (2003b), Lean buffering in serial production lines with non-exponential machines, The University of Michigan, Systems Science and Engineering Division, Department of Electrical Engineering and Computer Science, Control Group Report No. CGR-03-13, November 2003.
9. Goldratt, E., and Cox, J. (1986), *The Goal*, North Rivers Press.
10. Groover, M.P. (2001), *Automation, Production Systems, and Computer Integrated Manufacturing*, Second Edition, Prentice Hall.
11. Jacobs, D., and Meerkov, S.M. (1995a), Mathematical theory of improvability for production systems, *Mathematical Problems in Engineering*, Vol. 1, pp. 95–137.
12. Jacobs, D. and Meerkov, S.M. (1995b), A system-theoretic property of serial production lines: improvability, *International Journal of Systems Science*, Vol. 26, No. 4, pp. 755–785.
13. Kuo, C.-T., Lim, J.-T., and Meerkov, S.M. (1996), Bottlenecks in serial production lines: A system-theoretic approach, *Mathematical Problems in Engineering*, Vol. 2, pp. 233–276.

14. Li, J. (2003), Modeling and analysis of complex production systems, Published in the Proceedings of the *Fourth Aegean International Conference on "Analysis of Manufacturing Systems,"* pp. 203–212, Samos Island, Greece, July 1–4.
15. Li, J. and Meerkov, S.M. (2000), Bottlenecks with respect to due-time performance in pull serial production lines, *Mathematical Problems in Engineering*, Vol. 5, pp. 479–498.
16. Li, J. and Meerkov, S.M. (2001), Customer demand satisfaction in production systems: A due-time performance approach, *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 4, pp. 472–482.
17. Noble, J.S. and Tanchoco, J.M.A. (1993), Design justification of manufacturing systems – A review, *The International Journal of Flexible Manufacturing Systems*, Vol. 5, pp. 5–25.

4

Work-Load and Server Allocation Problems

In this chapter, two separate design problems are considered, viz., the work-load allocation problem and the server allocation problem in production lines. In a broad sense both design problems are related to the allocation of work from the point of view of the operators. Section 4.1 of the chapter describes what is classically known as the work-load allocation problem, i.e., the allocation of work to each station of the line so that all the required work is undertaken having in mind any precedence requirements. A well-known empirically observed phenomenon, namely the bowl phenomenon, is described. Some computational issues are then discussed. In Section 4.2, the server allocation problem is described. In Section 4.3, the simultaneous optimization of the work allocation and server allocation problems is considered. Associated with this double optimal problem is the so-called *L*-phenomenon.

4.1 The Work-Load Allocation Problem

The work-load allocation problem in production lines is analogous to the assembly line balancing problem in that one is assigning to each work-station a certain amount of the work, in terms of time, which has to be done. The assembly line balancing problem is concerned with how much work should be done at each station given the precedence requirements. In the work-load allocation problem for production lines, the overall constraint is that the sum of the expected service times is a fixed constant and the work-load allocation problem essentially is to allocate this total time among the stations so as to optimize a given objective function, usually, throughput, X_K, or average work-in-process, $\overline{WIP}$.

In mathematical terms, the work-load allocation problem, WAP, may be stated as follows:

$$\max X(\mathbf{w}),$$

subject to:

$$\sum_{i=1}^{K} w_i = W, \tag{4.1}$$

C. T. Papadopoulos et al., *Analysis and Design of Discrete Part Production Lines*, Springer Optimization and Its Applications,
DOI: 10.1007/978-0-387-89494-2_4, © Springer Science+Business Media, LLC 2009

where $\mathbf{w} = (w_1, w_2, \ldots, w_K)$ denotes the vector of w_i's, $i = 1, 2, \ldots, K$. $w_i > 0$, $i = 1, 2, \ldots, K$, in turn, denotes the mean service time of the i, $i = 1, \ldots, K$ stations and W is a fixed constant, which may be normalized and set equal to K, the number of work-stations of the line. This normalized work-load is to be divided among the K stations. $X(w)$ denotes the throughput of the production line as a function of the mean service times. The line is configured with buffers of finite capacities between any two successive stations. Throughput is also a function of other parameters such as the buffer sizes, the number of servers at each work-station as well as the moments of higher than the first order of the service time distributions at each station. The latter set of three parameters are not decision variables in this section and are assumed to be fixed as specified in the production line layout. Also, in place of maximizing throughput, other performance measures may be used such as minimizing average WIP or minimizing the average flow (production) time. It is quite practical to minimize average WIP given that a specified level of throughput is achieved. The essence of the work-load allocation problem is the assignment of service rates to the machine stations to meet the objectives required. In practice, this would involve the appropriate use of operators and associated machines to achieve the desired service rates.

The distribution of service times used in unpaced (asynchronous) production lines is a matter of practical importance. Many models have used the exponential distribution for which the associated coefficient of variation is one. Other studies have used the normal (generally truncated) distribution with a range of coefficients of variation. There is a tendency to use the same service time distribution at each of the stations but maybe with different parameters. Generally speaking, the perceived view is that a more appropriate practical service time distribution is skewed to the right (see Dudley, 1968, Knott and Sury, 1987, Murrell, 1962, Buzacott and Shanthikumar, 1993, among others). Also, it may be noted that there seems to be a consensus among relevant experts, using experimental data, that the coefficient of variation of practical service time distributions should be in the range 0.2 to 0.5 (see Slack, 1982, Knott and Sury, 1987, Pike and Martin, 1994, among others). Reliance is also placed on Muth's observation (Muth 1977) that for practical purposes the throughput of a production line is a function of only the first two moments of the underlying distributions of service time. In sophisticated models it is possible to arrange specified values for the mean service time and the coefficient of variation at each station.

As far back as 1977, Hillier and Boling (1977) initiated a theoretical investigation into the existence of the bowl phenomenon for production lines with service times distributed according to exponential and phase-type distributions. They postulated three conjectures based on three properties, namely, the *reversibility* property, the *symmetricity* property and the *monotonicity* property, which if they could be shown to hold for a particular system, a bowl phenomenon would apply which would lead to an optimal throughput solution. As stated, these conjectures would be a sufficient condition for the existence of an optimal solution having the symmetrical bowl phenomenon.

A production line with intermediate buffers of equal capacities is said to be *reversible* if the throughput of the line remains the same if the order of work assigned to the stations is reversed, i.e.,

$$X(w_1, w_2, \ldots, w_K) = X(w_K, w_{K-1}, \ldots, w_1).$$

A work-load allocation is said to be *symmetrical* if $w_j = w_{K+1-j}$ for $j = 1, \ldots, K$.

The *monotonicity* property implies that the work-load w_j satisfies the following condition: $w_j > w_{j+1}$ for $j \leq [(K-1)/2]$ and $w_j < w_{j+1}$ for $j > [K/2] + 1$, where $[x]$ is the largest integer not exceeding x, $x \in R$.

It may be noted that the work-load allocation problem could be considered in the context of a fixed configuration of server allocation (but not specifying work undertaken by any server) and buffer space allocation which classically has resulted in the discovery of the bowl phenomenon. Alternatively, the work-load allocation problem could be considered in the context of a fixed buffer space configuration but with the server configuration not explicitly given. This latter simultaneous work-load and server allocations has resulted in the discovery of the *L*-phenomenon. In a broader sense, the work-load allocation problem is concerned not only with the allocation of work to each station but also with the allocation of capacity to each station through the allocation of the servers when $S > K$.

4.1.1 The bowl phenomenon

One of the most interesting aspects of the design of production lines is the role of the bowl phenomenon. Originally observed in experimental work by Hillier and Boling (1966), most researchers would now agree that optimizing throughput in a production line requires that the work-load be not uniformly allocated (equally balanced) among the stations of the line. A typical illustration of the bowl phenomenon is shown in Figure 4.1. This particular bowl arose in maximizing the throughput in a five-station production line with equal inter-station buffers each of size 3 slots.

It may be noted that because of the size of the system, an exact solution is possible. As may be seen from Figure 4.1, the curve connecting the service times is concave and is representative of the cross section of a bowl from which the title comes. To date no satisfactory theoretical proof of the required existence of the bowl phenomenon has been presented. Readers should be aware that whereas Figure 4.1 gives a relatively dramatic illustration of the bowl phenomenon, in many cases in practice the deviation from a uniform balance may be small or non-existent. As the reader will realize, the concept of reducing the work-load on a station in effect implies a smaller mean time to service and requires a more powerful station and so the station in effect could be described in "colloquial terms" as being a quicker or a more powerful station where the best operators may be assigned. Despite the absence of a rigorous mathematical proof of the existence under given conditions (buffer size allocation, service time distributions, number of stations and number of parallel servers at each station) of the bowl phenomenon, most experimental work confirms its existence. Also, some theoretical work in serial production lines involving the

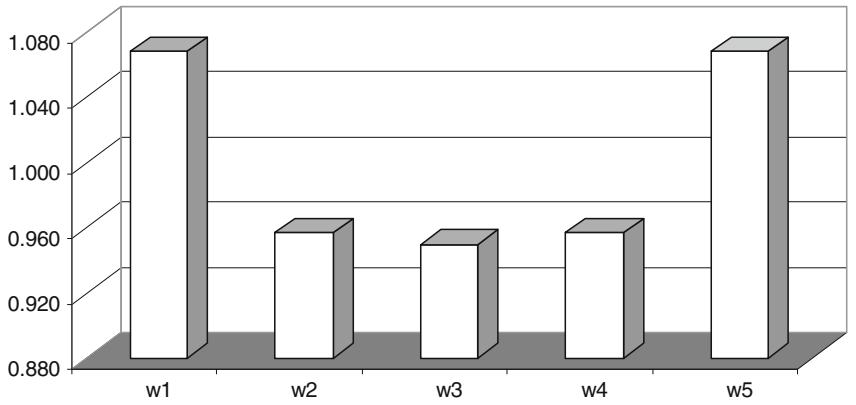


Fig. 4.1. The work-load allocation over five stations with inter-station buffer capacities of sizes $B_2 = B_3 = B_4 = B_5 = 3$ slots

appropriate ordering of machines of different service rates capabilities to achieve optimal performance does not lead to the conclusion that it does not exist even though this work is concerned with a different formulation of the problem as the service rates are specified and the order is unimportant from the point of view of production on the line. Relevant references include Tembe and Wolff (1974), Whitt (1985), Greenberg and Wolff (1988), Huang and Weiss (1990), Suresh and Whitt (1990), Shanthikumar, Yamazaki and Sakasegawa (1991), Ding and Greenberg (1991), Liao and Rosenshine (1992), Yamazaki, Sakasegawa and Shanthikumar (1992), Cheng and Zhu (1993) and Wan and Wolff (1993), among others.

The classical concept of the bowl phenomenon is that the work-load should be allocated among the work-stations according to a strictly concave function. However, in the literature, approximations to the bowl phenomenon have been made using piecewise linear approximations. Two such approximations may be noted: one-level and two- or more-level allocations (the reader is referred to Buzacott and Shanthikumar, 1993, formulas (5.94) and (5.95) for equal buffer sizes, on page 202). The one-level allocation corresponds to the perfectly balanced line, whereas the two-level allocation consists of stations with two different levels of mean service times: the outer (first and last) stations equally having a higher mean service time and the intermediate stations each having equally lower mean service times (see Figure 4.2 for a two-level approximation to a bowl phenomenon).

However, it is important for practitioners to note that in many studies where the bowl phenomenon was found to exist, the coefficient of variation of the service times and the impact of different levels of buffers between the stations were not explicitly considered. The controversy about the practical value of the bowl phenomenon to production line designers and production managers still exists. There are at least three aspects to this controversy: (i) Does the bowl phenomenon exist? On balance, there are sufficient carefully executed studies to support the contention that it does exist and is important particularly in situations where there are limited buffer sizes

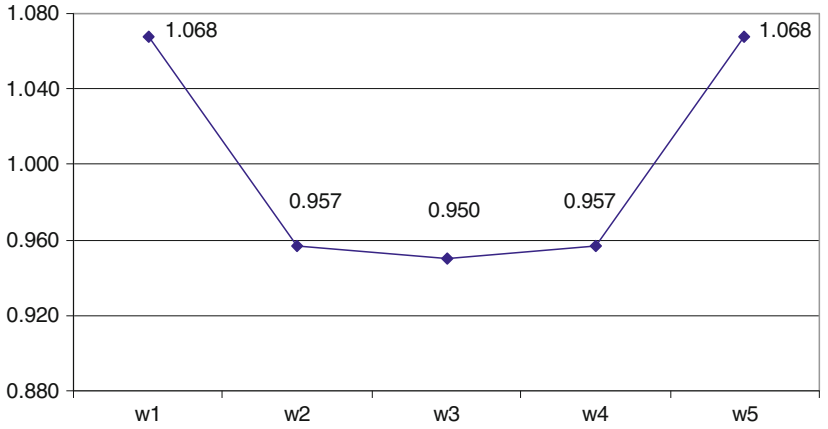


Fig. 4.2. Two-level approximation to a bowl phenomenon

and the coefficients of variation of service times are relatively close to one another. (ii) Is the bowl phenomenon a mirage in systems with a large number of states where exact analysis is not possible? Basically, this question cannot be really answered because generally the throughput in such systems is approximated by an algorithm and it is impossible to know the limits of the approximation to the throughput. (iii) Is it of value in practice? Some analysts might be of the view that 1% or 2% change in optimal throughput of a production system is of little relevance particularly having in mind how difficult it would be in practice to ensure that the service rates designated by the bowl phenomenon would be achieved. However, it should be remembered that production lines are designed for high volume and even a small change in throughput may well be worthwhile in commercial terms. In any case, it is probably as difficult practically to uniformly balance a line as to configure the line in accordance with the bowl phenomenon. So, why attempt a second best solution? The real issue is, of course, whether those mathematical models in which a bowl phenomenon is shown to be associated with maximum throughput are accurate models of the realities of actual production lines. This brings into question the validity of specifying processing times in stochastic terms generally using phase-type distributions. It might be the case that in production lines where there is a significant human operator involvement, as well as machine involvement, the bowl phenomenon is more relevant. The authors would encourage readers to make up their own minds about this controversy.

4.1.2 Computational issues

Here, the objective is to acquaint the reader with a few numerical techniques which have been found useful in relation to production line analysis regarding the work-load allocation problem. It is by no means a comprehensive survey of numerical analysis approaches.

Because of the general belief in the existence of the bowl phenomenon, there is often a need to obtain the maximum throughput through a process of numerical iteration. The usual efficient search procedures such as gradient search procedures are used. Given below is an algorithm for *the steepest ascent method of parallel tangents (PARTAN method)* which has been found useful. The approach is described in Buehler, Shah and Kempthorne (1964).

The steepest ascent method of parallel tangents (PARTAN method) for solving the work-load allocation problem

Step 1

Develop an appropriate initial feasible work-load allocation, $(w_1^0,\ldots,w_K^0)$ (uniform if there is no other information available). Then determine the throughput $X^0(w_1,\ldots,w_K)$ for this work-load after calculating the steady-state probabilities.

Step 2

Choose a small quantity h (based on experience) and determine the partial derivatives of $X^0(w_1^0,\ldots,w_K^0)$, numerically, as follows:

$$\frac{\partial X^0}{\partial w_i^0} = \frac{X(w_1^0,\ldots,w_i^0+h,\ldots,w_K^0) - X^0(w_1^0,\ldots,w_K^0)}{h}, \quad i=1,\ldots,K.$$

It should be noted that throughput, X, must be evaluated K times at this step (from the associated probabilities) and the value X^0 is also used.

Step 3

Evaluate $X^1(w_1^1,\ldots,w_K^1)$, where,

$$w_i^1 = w_i^0 + \left[\ell \frac{\partial X^0}{\partial w_i^0}\right], \quad i=1,\ldots,K,$$

for values of ℓ such that the work-load constraint: $\sum_{i=1}^K w_i^1 = K$ is satisfied.

Step 4

Repeat Step 3 to obtain in a similar fashion X^2, which is the optimal value of X along the line of the steepest ascent from Step 3.

Step 5

Knowing the starting point at Step 2 and the optimal point reached at Step 4, proceed along the line joining these two points again in steps until an optimal value of X is obtained. To clarify, if the initial point is given as $(w_1^0,\ldots,w_K^0)$ and the point reached in Step 4 is $(w_1^2,\ldots,w_K^2)$, evaluate X at the following point

$$(w_1^2+\ell(w_1^2-w_1^0),\ldots,w_K^2+\ell(w_K^2-w_K^0))$$

for values of $\ell > 0$ until an optimal for X^3 is reached.

Step 6

Return to Step 2 and continue on to Steps 3, 4 and 5 until a satisfactory convergence to an optimal X is achieved.

General remark

Any special features of the network, e.g., symmetry, may be explored to reduce computational effort.

Another numerical approach which is useful for solving the work-load allocation problem is described in Baruh and Altiok (1991). The authors used the first- and second-order numerical perturbations to determine the optimal work-load allocation in production lines.

Next, an approximate method, the two-level work-load allocation algorithm, proposed by Buzacott and Shanthikumar (1993), for obtaining near optimal throughputs and work-load allocations in production lines with single-machine work-stations is given. The algorithm is available at the website associated with this book with abbreviated name TLWLA. This is a stand-alone optimization algorithm.

If the inter-station buffers, $B_2,\ldots,B_K$, have the same size B ($B_2 = B_3 = \cdots = B_K = B$), where $(K-1)B = N$, the total number of buffer slots in the system, and the total work-load has been normalized to equal the total number of stations, K, Buzacott and Shanthikumar's near optimal approximations are as follows:

$$X^*(B,K) = \frac{K(B+1)+2}{K(B+3)}. \tag{4.2}$$

The associated work-load allocation is a two-level bowl approximation, given by:

$$w_1^* = w_K^* = \frac{K(B+2)}{K(B+1)+2} \tag{4.3}$$

$$w_i^* = \frac{K(B+1)}{K(B+1)+2}, \quad i = 2,\ldots,K-1. \tag{4.4}$$

If the inter-station buffers, $B_j,\ j = 2,3,\ldots,K$, are unequal with respective capacities $B_j,\ j = 2,3,\ldots,K$, and again the total work-load has been normalized to K, Buzacott and Shanthikumar's near optimal approximations are:

$$X^*(B_2,\ldots,B_K,K) = 1 - \frac{2}{K}\sum_{i=2}^{K}\frac{1}{B_i+3}. \tag{4.5}$$

The associated work-load allocation is a multi-level and not necessarily symmetrical bowl approximation, given by:

$$w_i^* = \frac{K\alpha_i}{\sum_{i=1}^{K}\alpha_i}, \quad i = 1,\ldots,K, \tag{4.6}$$

where

$$\alpha_1 = \frac{B_2+2}{B_2+3}, \tag{4.7}$$

$$\alpha_j = 1 - \frac{1}{B_j+3} - \frac{1}{B_{j+1}+3}, \quad j = 2,\ldots,K-1, \tag{4.8}$$

$$\alpha_K = \frac{B_K+2}{B_K+3}. \tag{4.9}$$

The following summary may be of value to the reader who wishes to use the software available at the website associated with this book in solving work-load allocation problems.

Work-Load Allocation Problem (WAP)

1. **TLWLA**
   - Exponential service time distributions.
   - Any distribution of buffers.

   The TLWLA procedure will give an approximate work-load allocation and approximate optimal throughput of the production line.
2. **MARKOV** and **SA/GA**
   - For short reliable or unreliable production lines with Erlang-k ($k \geq 1$) service and repair times and exponential times to failure.
3. **DECO-1** and simulated annealing/genetic algorithms **SA/GA**
   - For large reliable exponential production lines with single machine stations.
   - Finite intermediate buffers.
4. **DECO-2** and simulated annealing/genetic algorithms **SA/GA**[1]
   - For large reliable exponential production lines with multiple parallel identical machine stations.
   - Finite intermediate buffers.

4.2 The Server Allocation Problem

One of the allocation issues for the systems designer is to allocate the number of servers, S (when $S > K$), over the given fixed K stations. Conceptually what is involved is the assignment of service capacity to each of the K stations to meet the objectives of maximizing throughput or minimizing average WIP.

If $\mathbf{s} = (S_1, S_2, \ldots, S_K)$ denotes the vector of servers allocated to the i stations, $i = 1,2,\ldots,K$, in mathematical terms the server allocation problem, SAP, is as follows:

$$\max X(\mathbf{s})$$

[1] Details of simulated annealing (SA) and genetic algorithms (GA) as optimization procedures are given in Chapter 5, Section 5.4.

subject to:

$$\sum_{i=1}^{K} S_i = S$$

for fixed allocation of work to each station and fixed buffer alocation.

Interesting papers in this area include Hillier and So (1989), Futamura (2000), Hillier and So (1995) and Magazine and Stecke (1996), with the latter two papers dealing with various combinations of the work-load, server and buffer allocations.

Hillier and So (1989) considered production lines with exponential, Erlang with two phases of service and Coxian with two phases of service at each station, no intermediate buffers or with just one buffer slot among the stations and equal work-load allocation over all the stations. Define $n = [S/K]$ as the greatest integer $\leq S/K$ and $E = S - nK$, where $S > K$ is the total number of servers available for allocation over the stations. The main results of this study may be represented as rules for maximizing the throughput and are as follows:

- *Rule 1*: If S/K is an integer, allocate the servers uniformly among the K stations.
- If S/K is not an integer, initially allocate $n = [S/K]$ to each of the K stations making a total initial allocation of Kn servers, and the balance of the servers, E, are allocated according to the following rules:
  - *Rule 2*: If $E = 1$ and K is odd, then allocate the extra server to the center station.
  - *Rule 3*: If $E = 1$ and K is even, then allocate the extra server to one of the two central stations. If a lower $\overline{WIP}$ is of interest, choose the left central station, i.e., the station nearest the beginning of the line.
  - *Rule 4*: If $E = K - 1 > 1$, then allocate an extra server to each of the stations except station 1, but for extremely large n in which case more than one server may be assigned to a single station.
  - *Rule 5*: If $E = K - 2 > 1$, then allocate an extra server to every station except the first and last stations.
  - *Rule 6*: If $1 < E < K - 2$, then allocate the extra servers "almost uniformly" over the interior stations.

Although the work-load allocation is uniform, these design rules generally support the concept of allocating extra servers to the interior stations and therefore in accordance with the idea of an inverse bowl of service capacity.

Futamura (2000) considered the case where the coefficient of variation of the servers was not identical. The general rule of thumb is to allocate more servers to the stations with a higher coefficient of variation although, as the author indicated, more research is needed to derive precise design rules.

It is possible to solve any server allocation problem using the algorithms available at the website associated with this book for the following type of serial production line:

- Parallel exponential reliable machines at each station,
- Number of stations: No practical limit (over 1000 stations),
- Number of buffer slots: 5000.

Specifically, one uses the evaluative decomposition algorithm for solving serial production lines with multiple parallel-machine stations in conjunction with an optimization algorithm such as simulated annealing and genetic algorithm to find an optimal or near optimal solution to the server allocation problem.

The following detailed summary may be of value to the reader who wishes to use the software available at the website associated with this book in solving server allocation problems.

1. **Server Allocation Problem (SAP)**
   - Apply the rules given above for the allocation of servers and use DECO-2 to determine the throughput with the specified allocated work-load.
   - Exponential service time distributions and reliable machines.
2. **DECO-2** and simulated annealing/genetic algorithms **SA/GA**[*]
   - For large reliable exponential production lines with multiple parallel identical machine stations and finite intermediate buffers with the specified work-load allocation.

[*] Details of simulated annealing (SA) and genetic algorithms (GA) as optimization procedures are given in Chapter 5, Section 5.4.

4.3 The Simultaneous Work-Load and Server Allocation: The *L*-phenomenon

Up to now, design of production lines was considered in the context of one dimension only, i.e., work-load allocation or server allocation. Now consideration is given to the simultaneous allocation of work-load and servers. Investigations on the simultaneous allocations of both work-load and servers have led to the discovery by Hillier and So (1995, 1996) of the so called *L*-phenomenon, when the objective is to maximize throughput. The design rule here is to allocate just one server to each one of the stations and all the remaining servers $(S-K+1)$ to one of the two end stations of the line. If one considers the smallest WIP, then extra servers must be allocated to the first station, although this will slightly reduce the throughput from the optimal value which occurs when the extra servers are allocated to the last station. This allocation resembles the shape of the capital letter *L* and thus the name of the phenomenon. The associated work-load allocation at optimal illustrates the well-known bowl phenomenon but of course it is not symmetrical with a significantly higher work-load allocation to the station to which the extra servers have been allocated. Hillier and So (1996) also derived the marginal contributions in terms of throughput of additional servers and showed that in fact it increases with the addition of extra servers. This was designated by Hillier and So as "the multiple-server phenomenon" (type 1). Hillier and So (1995) also observed that a "second multiple-server phenomenon" is in operation in the absence of the *L*-phenomenon when the number of servers is an integer multiple of the number of stations, i.e., $S = sK$, where s denotes the number of servers allocated per station. In that case, as s is increased, the marginal increase

in throughput using the corresponding optimal work allocation increases monotonically in s. It should be noted that Hillier and So used a model consisting of K stations and that the service times are independent and identically distributed (i.i.d.) random variables following an exponential distribution with fixed means that are normalized in the analysis.

The primary concept behind the bowl phenomenon is that the interior stations are given preferential treatment, i.e., they are given less work to do in an expected sense. So, the situation is that if you have stations with equal service rate capacities, the designer should allocate work so that the processing time of a part in the interior stations is lower than the processing time of the part at the outer stations. Now, however, when work-load and servers are allocated simultaneously to achieve optimal throughput, it is the end stations that are given preferential treatment with respect to server allocation, although a non-symmetrical bowl phenomenon of work-load allocation still exists, at the optimal throughput. More precisely, the corresponding optimal work-load allocation assigns by far the largest amount of work per server to the station with the largest number of servers and the work allocations per server (the work assigned to the station) are monotonically decreasing from the station with the largest number of servers (either the first or the last station) to the next to the end station. The end station (either the first or the last station, to which only one server is assigned) has an increased work-load per server allocation over that allocated to the next to end station. These results show that unbalancing the servers and work-loads can provide substantial improvements in throughput of unpaced production lines with service rate variability over the balanced as possible allocations which, of course, would be optimal in paced lines or deterministic lines or lines with variable service rates with infinite inter-station buffer capacities.

The *L*-phenomenon which, as noted above, occurs under the simultaneous allocation of servers and work-load may be contrasted with the results obtained when only server allocation is being considered given a fixed work-load allocation. It may be pointed out that in our cases no evidence that the bowl phenomenon does not exist is forthcoming, although it might have been conjectured that even in the simultaneous allocation of servers and work-load, the allocation of the servers would give preferential treatment to the inner stations of the line contributing in a simplistic manner to the bowl phenomenon. Clearly, much fundamental research effort is required for a full understanding of the bowl phenomenon and its ramifications.

Hillier and So (1995) gave a heuristic allocation scheme for the case where there are upper and lower bounds on the number of servers to be allocated at each station. The steps of this scheme are as follows:

Step 1: Allocate the minimum number of servers required at each station.
Step 2: Allocate as many as possible extra servers at the last station.
Step 3: If there are extra servers left, allocate as many as possible to the first station.
Step 4: If there are still servers remaining, allocate as many as possible to the next to the last station, then as many as possible to the second station, etc.
Step 5: The procedure is concluded when there are no remaining servers.

Hillier and So also indicated that in their experience, if the number of stations is itself a decision variable, throughput would be maximized if $K = [S/S_{\max}]$, where K is the optimal number of stations, $[x]$ indicates the maximum integer less than or equal to x, S is the number of available servers and $S_{\max}$ is the upper bound on the servers that can be assigned to any station. From Hiller and So's empirical studies, the optimal server allocation to achieve maximum throughput would be in accordance with the scheme of the L-phenomenon given above.

4.4 Related Bibliography

4.4.1 Bowl phenomenon

Hillier and Boling (1966) first observed and conjectured the existence of the bowl phenomenon by examining two-, three- and four-station lines with exponentially distributed processing times.

Rao (1975a) analyzed a two-station production line and showed that for moderate coefficients of variation of the processing times, the mean service rates for Erlang and normal density functions of the service times differ only marginally. Rao showed that throughput improves by allocating a slightly higher work-load to the less variable station.

Rao (1975b) concluded that at high values of coefficient of variation, the type of service time distribution has a considerable effect on the efficiency of a two-station series system when the variability of service times at the stations is not the same.

Rao (1976) analyzed a three-station production line and showed that the variability imbalance plays a decisive role and outweighs the bowl phenomenon.

De La Wyche and Wild (1977) investigated via simulation the imbalance in service time variability, the imbalance in buffer storage and the interaction of service time and buffer imbalance.

Hillier and Boling (1977) considered short lines with Erlang service times and proposed three conjectures implying the bowl phenomenon.

Magazine and Silver (1978) studied the effect on throughput from different choices of design parameters. They proposed some heuristics to find approximate values for the optimal work-load allocation and for the throughput.

El-Rayah (1979b) examined the effect of inequality of interstage buffer capacities and operation time variability on the throughput of production lines.

Hillier and Boling (1979) studied the change of the optimal allocation of work between stations with respect to (i) the number of work-stations in the line, (ii) the limit on the amount of work-in-process (WIP) and (iii) the variance of station service or processing times. They re-confirmed the existence of the bowl phenomenon.

El-Rayah (1979a) conducted computer simulations and confirmed the bowl phenomenon too.

Mishra et al. (1985) showed that in systems consisting of a hyperexponential station, the guideline of allocating more work-load to the stations with less variability is violated due to the fact that hyperexponential is a composite distribution.

Lau and Martin (1986) developed a decision support system for the design of production lines, incorporating a bowl phenomenon.

Muth and Alkaff (1987) presented a method for analyzing distribution-free three-station production lines and offered a bibliography on the work-load allocation and the bowl phenomenon.

Thompson and Burford (1988) showed that the bowl phenomenon is associated with an imbalance in absolute variability and that the bowl effect vanishes in cases where a minimal level of in-process buffer stock is provided.

So (1989) conducted simulation experiments in production lines with normally distributed processing times and showed that throughput can be improved by appropriately unbalancing work allocations.

Pike and Martin (1994) studied the bowl phenomenon in production lines under realistic operating conditions and found out that bowl-shaped configurations perform better than perfectly balanced lines for systems of at least 30 stations in length and with inter-station buffer capacities of up to one unit. They also showed that the optimal two-level allocation of mean service times performs no worse than the optimal multi-level allocation. In addition, they discovered that the amount of imbalance in a line can generally be double the imbalance in an optimal bowl and still perform at least as well as the balanced line. Finally, the authors showed that optimal bowl configurations are not particularly sensitive to coefficient of variation or distribution shape within a realistic range.

Lau (1994) simulated production lines with different buffer sizes and different combinations of station service times' means and variances and concluded that throughput is maximized when the means and variances are both balanced. Lau also found out that variance imbalance has a very small effect on throughput, among other findings.

Hillier and So (1995) considered combinations of the three design problems in production lines: the work-load allocation, the buffer allocation and the server allocation problems.

Spinellis, Papadopoulos and MacGregor Smith (2000) also examined combinations of the above three design problems using a robust generalized queueing network algorithm as an evaluative procedure and simulated annealing for optimizing production line configurations.

Shanthikumar and Yao (1988) dealt with the server allocation problem in multiple center manufacturing systems. They formulated the problem as a nonlinear integer program of allocating servers in a closed queueing network to maximize the throughput of the system.

Dallery and Stecke (1990) addressed the problem of the optimal allocation of servers and work-loads in closed queueing networks. They used decomposition to obtain results for the subnetworks in isolation and then to solve the optimal configuration problem. The authors also recommended applications of their results to design and planning of flexible manufacturing systems.

4.4.2 Reversibility

Dattatreya (1978) Defined *C*- and *D*-reversibility, as follows. A tandem queueing system is said to be *C-reversible* if the original system has the same throughput as its reversed system. A blocking system, on the other hand, is defined as *D-reversible* if the distributions of times of the departure epochs from both systems are all identical.

Makino (1964) proved *C*-reversibility for simple systems.

Yamazaki and Sakasegawa (1975) proved *D*-reversibility, whereas Dattatreya (1978) and Muth (1979) proved independently the reversibility property.

Yamazaki, Kawashima and Sakasegawa (1985) proved *C*-reversibility in two-station blocking systems with parallel machines at each station and stochastic service times. The same authors proved that this property cannot be extended to larger similar systems (with parallel machines at each station, stochastic service times and finite intermediate buffers).

Melamed (1986) provided some results on the reversibility and duality of some tandem blocking queueing systems.

Dallery, Liu and Towsley (1991) considered reversibility in fork/join queueing networks with blocking after servcie (manufacturing blocking).

References

1. Altiok, T. (1997), *Performance Analysis of Manufacturing Systems*, Springer-Verlag.
2. Baruh, H. and Altiok, T. (1991), Analytical perturbations in Markov chains, *European Journal of Operational Research*, Vol. 51, pp. 210–222.
3. Buehler, R.J., Shah, B.V. and Kempthorne, O. (1964), Methods of parallel tangents, *Chemical Engineering Progress Symp. Series*, Vol. 60, No. 50, pp. 1–7.
4. Buzacott, J.A. and Shanthikumar, J.G. (1993), *Stochastic Models of Manufacturing Systems*, Prentice Hall.
5. Cheng, D.W. and Zhu, Y. (1993), Optimal order of servers in a tandem queue with general blocking, *Queueing Systems*, Vol. 14, pp. 427–437.
6. Dallery, Y., Liu, Z. and Towsley, D. (1991), Reversibility in fork/join queueing networks with blocking after service, Laboratoire MASI, Universite P. et M. Curie, Paris, France.
7. Dallery, Y. and Stecke, K.E. (1990), On the optimal allocation of servers and work-loads in closed queueing networks, *Operations Research*, Vol. 38, No. 4, pp. 694–703.
8. Dattatreya, E.S. (1978), *Tandem Queueing Systems with Blocking*, Ph.D. Dissertation, Department of Industrial Engineering and Operations Research, University of California, Berkeley.
9. De La Wyche, P. and Wild, R. (1977), The design of imbalanced series queue flow lines, *Operational Research Quarterly*, Vol. 28, No. 3, ii, pp. 695–702.
10. Ding, J. and Greenberg, B. (1991), Bowl shapes are better with buffers–sometimes, *Probability in the Engineering and Informational Sciences*, Vol. 5, pp. 159–169.
11. Dudley, N.A. (1968), *Work Measurement, Some Research Studies*, Macmillan.
12. El-Rayah, T.E. (1979a), The efficiency of balanced and unbalanced production lines, *International Journal of Production Research*, Vol. 17, No. 1, pp. 61–75.

13. El-Rayah, T.E. (1979b), The effect of inequality of interstage buffer capacities and operation time variability on the efficiency of production line systems, *International Journal of Production Research*, Vol. 17, No. 1, pp. 77–89.
14. Futamura, K. (2000), The multiple server effect: Optimal allocation of servers to stations with different service-time distributions in tandem queueing networks, *Annals of Operations Research*, Vol. 93, pp. 71–90.
15. Greenberg, B. and Wolff, R.W. (1988), Optimal order of servers for tandem queues in light traffic, *Management Science*, Vol. 34, No. 4, pp. 500–508.
16. Helber, S. (1999), *Performance Analysis of Flow Lines with Non-Linear Flow of Material*, Springer-Verlag.
17. Hillier, F.S. and Boling, R.W. (1966), The effect of some design factors on the efficiency of production lines with variable operation times, *The Journal of Industrial Engineering*, Vol. 17, pp. 651–658.
18. Hillier, F.S. and Boling, R.W. (1977), Toward characterizing the optimal allocation of work in production line systems with variable operation times, *Advances in Operations Research* (Marc Roubens, Ed.), North-Holland, Amsterdam, pp. 649–658.
19. Hillier, F.S. and Boling, R.W. (1979), On the optimal allocation of work in symmetrically unbalanced production line systems with variable operation times, *Management Science*, Vol. 25, No. 8, pp. 721–728.
20. Hillier, F.S. and So, K.C. (1989), The assignment of extra servers to stations in tandem queueing systems with small or no buffers, *Performance Evaluation*, Vol. 10, pp. 219–231.
21. Hillier, F.S. and So, K.C. (1995), On the optimal design of tandem queueing systems with finite buffers, *Queueing Systems*, Vol. 21, pp. 245–266.
22. Hillier, F.S. and So, K.C. (1996), On the simultaneous optimization of server and work allocations in production line systems with variable processing times, *Operations Research*, Vol. 44, No. 3, pp. 435–443.
23. Huang, C.C. and Weiss, G. (1990), On the optimal order of M machines in tandem, *Operations Research Letters*, Vol. 9, pp. 299–303.
24. Iyama, T. and Ito, S. (1987), The maximum production rate for an unbalanced multi-server flow line system with finite buffer storage, *International Journal of Production Research*, Vol. 25, No. 8, pp. 1157–1170.
25. Knott, K. and Sury, R.J. (1987), A study of work-time distributions in unpaced tasks, *IIE Transactions*, Vol. 19, pp. 50–55.
26. Lau, H.-S. (1994), Allocating work to a two-stage production system with interstage buffer, *International Journal of Production Economics*, Vol. 36, pp. 281–289.
27. Lau, H.-S. and Martin, G.E. (1986), A decision support system for the design of unpaced production lines, *International Journal of Production Research*, Vol. 24, No. 3, pp. 599–610.
28. Liao, C.-J. and Rosenshine, M. (1992), A heuristic for determining the optimal order in a tandem queue, *Computers and Operations Research*, Vol. 19, No. 2, pp. 133–137.
29. Magazine, M.J. and Silver, G.L. (1978), Heuristics for determining output and work allocations in serial flow lines, *International Journal of Production Research*, Vol. 16, No. 3, pp. 169–181.
30. Magazine, M.J. and Stecke, K.E. (1996), Throughput for production lines with serial work stations and parallel service facilities, *Performance Evaluation*, Vol. 25, pp. 211–232.
31. Makino, T. (1964), On the mean passage time concerning some queueing problems of the tandem type, *Journal of the Operations Research Society of Japan*, Vol. 7, pp. 17–47.
32. Melamed, B. (1986), A note on the reversibility and duality of some tandem blocking queueing systems, *Management Science*, Vol. 32, No. 12, pp. 1648–1650.

33. Mishra, A., Acharya, D., Rao, N.P., and Sastry, G.P. (1985), Composite stage effects in unbalancing of series production systems, *International Journal of Production Research*, Vol. 23, No. 1, pp. 1–20.
34. Murrell, K.F.H. (1962), Operator variability and its industrial consequence, *International Journal of Production Research*, Vol. 1, pp. 39–55.
35. Muth, E.J. (1977), Numerical methods applicable to a production line with stochastic servers, *TIMS Studies in the Management Sciences*, 7:143–159.
36. Muth, E.J. (1979), The reversibility property of production lines, *Management Science*, Vol. 25, No. 2, pp. 152–158.
37. Muth, E.J. and Alkaff, A. (1987), The bowl phenomenon revisited, *International Journal of Production Research*, Vol. 25, No. 2, pp. 161–173.
38. Noble, J.S. and Tanchoco, J.M.A. (1993), Design justification of manufacturing systems – A review, *The International Journal of Flexible Manufacturing Systems*, Vol. 5, pp. 5–25.
39. Papadopoulos, H.T., Heavey, C., and Browne, J. (1993), *Queueing Theory in Manufacturing Systems Analysis and Design*, Chapman & Hall.
40. Pike, R. and Martin, G.E. (1994), The bowl phenomenon in unpaced lines, *International Journal of Production Research*, Vol. 32, No. 3, pp. 483–499.
41. Rao, N.P. (1975a), Two-stage production systems with intermediate storage, *AIIE Transactions*, Vol. 7, No. 4, pp. 414–421.
42. Rao, N.P. (1975b), On the mean production rate of a two-stage production system of the tandem type, *International Journal of Production Research*, Vol. 13, No. 2, pp. 207–217.
43. Rao, N.P. (1976), A generalization of the 'bowl phenomenon' in series production systems, *International Journal of Production Research*, Vol. 14, No. 4, pp. 437–443.
44. Shanthikumar, J.G., Yamazaki, G., and Sakasegawa, H. (1991), Characterization of optimal order of servers in a tandem queue with blocking, *Operations Research Letters*, Vol. 10, pp. 17–22.
45. Shanthikumar, J.G. and Yao, D.D. (1988), On server allocation in multiple center manufacturing systems, *Operations Research*, Vol. 36, No. 2, pp. 333–342.
46. Slack, N. (1982), Work time distributions in production system modelling, Research paper, Oxford Centre for Management Studies.
47. Smunt, T.L. and Perkins, W.C. (1990), The efficiency of unbalancing production lines: an alternative interpretation, *International Journal of Production Research*, Vol. 28, No. 6, pp. 1219–1220.
48. So, K.C. (1989), On the efficiency of unbalancing production lines, *International Journal of Production Research*, Vol. 27, No. 4, pp. 717–729.
49. So, K.C. (1990), On the efficiency of unbalancing production lines: a reply to an alternative interpretation, *International Journal of Production Research*, Vol. 28, No. 6, pp. 1221–1222.
50. Spinellis, D., Papadopoulos, C., and MacGregor Smith, J. (2000), Large production line optimization using simulated annealing, *International Journal of Production Research*, Vol. 38, No. 3, pp. 509–541.
51. Suresh, S. and Whitt, W. (1990), Arranging queues in series: A simulation experiment, *Management Science*, Vol. 36, No. 9, pp. 1080–1091.
52. Tembe, S.V. and Wolff, R.W. (1974), The optimal order of service in tandem queues, *Operations Research*, Vol. 24, pp. 824–832.
53. Thompson, W.W. and Burford, R.L. (1988), Some observations on the bowl phenomenon, *International Journal of Production Research*, Vol. 26, No. 8, pp. 1367–1373.
54. Yamazaki, G., Kawashima, T., and Sakasegawa, H. (1985), Reversibility of tandem blocking systems, *Management Science*, Vol. 31, No. 1, pp. 78–83.

55. Yamazaki, G. and Sakasegawa, H. (1975), Properties of duality in tandem queueing systems, *Annals of the Institute of Statistical Mathematics*, Vol. 27, pp. 201–212.
56. Yamazaki, G., Sakasegawa, H., and Shanthikumar, J.G. (1992), On optimal arrangement of stations in a tandem queueing system with blocking, *Management Science*, Vol. 38, No. 1, pp. 137–153.
57. Wan, Y.-W. and Wolff, R.W. (1993), Bounds for different arrangents of tandem queues with nonoverlapping service times, *Management Science*, Vol. 39, pp. 1173–1178.
58. Whitt, W. (1985), The best order for queues in series, *Management Science*, Vol. 31, No. 4, pp. 475–487.

5

The Buffer Allocation Problem

The buffer allocation problem, BAP, is concerned with the allocation of a certain fixed number of buffer slots, N, among the $K-1$ intermediate buffer locations of a production line in order to meet some specified objective. The number of stations of the line is fixed at K, the number of servers assigned to each station is fixed and the work allocation $\mathbf{w} = (w_1, w_2, \ldots, w_K)$ is also fixed.

The buffer allocation problem is of particular interest to operations management in that in many practical production line situations, the allocation of buffer space may be the primary flexibility available to the organization. Clearly, buffer space is an expensive resource and so, ideally models involving cost considerations are very desirable. Of course, there are also plant layout issues involved.

At least three buffer allocation problems have been identified in the literature and these are described in Section 5.1. Solutions of the buffer allocation problems are discussed in Section 5.2. Special solution approaches to buffer allocation problems in short lines are the subject of Section 5.3, whereas solution approaches to buffer allocation problems in longer lines are treated in Section 5.4.

5.1 Formulation of the Buffer Allocation Problems

The formulation of the buffer allocation problems depends on the objective function chosen. These objective functions may be concerned with maximizing throughput, minimizing average work-in-process, or minimizing the total number of buffer slots, subject in each case to appropriate constraints. In detail:

Problem **BAP-A**:

$$\max X(\mathbf{n}) = \max X(N_2, \ldots, N_K)$$

subject to

$$\sum_{j=2}^{K} N_j = N$$

$$N_j \geq 0$$

C. T. Papadopoulos et al., *Analysis and Design of Discrete Part Production Lines*, Springer Optimization and Its Applications,
DOI: 10.1007/978-0-387-89494-2_5,

where $\mathbf{n} = (N_2, N_3, \ldots, N_K)$ denotes the vector of the buffer sizes, $N_j,\ j = 2, \ldots, K$, which are integer numbers and $X(\mathbf{n}) = X(N_2, \ldots, N_K)$ is the throughput of the K-station production line as a function of the buffers' sizes vector.

Problem **BAP-B**:

$$\min L(\mathbf{n}) = \min L(N_2, \ldots, N_K)$$

subject to

$$\begin{aligned} X(\mathbf{n}) = X(N_2, \ldots, N_K) &\geq X_0 \\ \sum_{j=2}^{K} N_j &\leq N \\ N_j &\geq 0 \end{aligned}$$

where $L(\mathbf{n}) = L(N_2, \ldots, N_K)$ denotes the average WIP, $\overline{WIP}$, as a function of N_j, $j = 2, \ldots, K$, which are integer numbers and X_0 is a specified throughput level.

Problem **BAP-C**:

$$\min N = \sum_{j=2}^{K} N_j$$

subject to

$$\begin{aligned} X(\mathbf{n}) = X(N_2, \ldots, N_K) &\geq X_0 \\ N_j &\geq 0 \end{aligned}$$

where $N_j, j = 2, \ldots, K$ are integer numbers and X_0 is a specified throughput level.

As a consequence of the proof by Meester and Shanthikumar (1990) of the concavity of the throughput of tandem queueing systems with finite buffer storage space, it is clear that problem BAP-A is an increasing function of the total buffer space N. Hence the results obtained for problem BAP-A can be used to solve problem BAP-C. Thus, the above three problems really are reduced to two problems. Generally speaking, because of the discrete nature of the buffer allocation and the unavailability of expressions in closed form, numerical approaches to the solution of the problems are inevitable even in situations with relatively small number of states. Such approaches are discussed below.

5.2 Solution of the Buffer Allocation Problems

A useful division of buffer allocation problems is based on the concepts of *short* and *longer* lines. Although precise definitions are impossible, in our experience, short lines might be designated as production lines with up to six stations with a maximum of up to 20 buffer slots in total. Lines with specifications outside these ranges may be classified as longer lines. Either of these lines consisting of single-machine stations may be *balanced* or *unbalanced*. By balanced is meant a line with equal mean service or processing times at each of the K stations. Unbalanced lines may

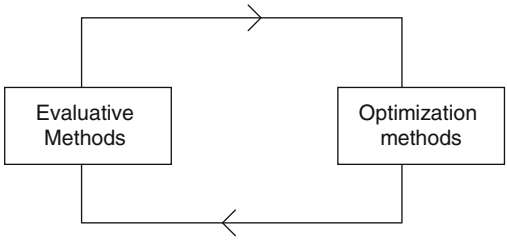


Fig. 5.1. General process of solution of buffer allocation problems

be classified as (i) μ-unbalanced, where the allocation of work is unbalanced across the stations, (ii) c.v.-unbalanced, where the c.v. (coefficient of variation) of the service times at the stations are not identical and (iii) fully unbalanced lines which are both μ-unbalanced and c.v.-unbalanced. A further characteristic of production lines is *reliable* and *unreliable*. Our definition of reliable lines is that each station of the line cannot fail, i.e., if free, the single server at the station is available to immediately serve a waiting part and all servers are perfectly reliable.

The method of solution of the buffer allocation problem follows the process as indicated in Figure 5.1. This solution process, not unique to the buffer allocation problem, basically consists of a loop process which leads to an optimal solution after a finite number of iterations. The analyst initiates the process by specifying an initial configuration of the system assuming values for the decision variables, in this case, the buffer allocation. Clearly, the experience of the analyst is of value at this stage. The evaluative method determines the value of the performance objective for the system as specified. The optimization or generative method (search algorithm) takes over and presents to the evaluative method a sequence of candidate systems with new values for the decision variables. The evaluative method calculates for each system presented the value of the performance measure. The effectiveness of the overall process depends on the efficiency with which the generative method generates suitable candidate systems for evaluation as well as on the effictiveness of the evaluative method itself. Clearly, a very efficient internal accounting process is required.

Evaluative methods, which predict the performance measures of the system, are based on aggregation approaches, decomposition methods, expansion or any approximate methods, Markovian exact models and simulation. Optimization methods, on the other hand, which lead to the optimal values of the decision variables and work on results of the evaluative methods, are very varied as will be described further below in Sections 5.3 and 5.4. A clear distinction should be made between an exact solution to the specified problem (however idealized initially) and an approximate solution to the same problem. For example, a Markovian model gives an exact solution to series production lines, whereas a decomposition approach to the same system attempts to develop a solution that is very close to the exact solution. However, decomposition methods because of their computational efficiency may be the only practical approach for the solution of systems with a large number of states. Again, as far as optimization is concerned, it is clear that enumeration, if possible, will give an exact

optimal solution. Enumeration is generally only possible for very small systems and other approaches to optimization are required for large systems. Such other numerical approaches may have some difficulty in actually reaching the precise optimal solution and so, the analyst should exercise caution in ascertaining that the optimal solution has been achieved.

In the following Sections 5.3 and 5.4 an attempt is made to assist the reader in understanding the methods used for the solution of the buffer allocation problems in the specified production lines.

5.3 Solution Approaches to the BAP in Short Lines

The methods used for the solution of the buffer allocation problems in long lines are also applicable to short lines. Here, however, the objective is to indicate to the reader those methods which are practically applicable only to short lines. Buffer allocation problems in short production lines whether balanced or unbalanced, reliable or unreliable are generally solved using Markovian and expansion evaluative methods (see Sections 2.1 and 2.3 in Chapter 2) with complete enumeration as the optimizing method. The latter algorithm is available at the website associated with this text with abbreviated name CE.

An example of the results obtained are given in Papadopoulos and Vidalis (1998). Their model consists of single-machine K stations which are perfectly reliable with processing times following an exponential or Erlang-k distribution with k phases of service ($k = 2,3,4$), with $K = 13$ for the case of the exponential distribution and $K = 6$ for the case of the Erlang-2 distribution. The objective function was to maximize the throughput of the line, by allocating a given total number, N, of buffer slots among the $K-1$ intermediate buffers of the line (problem BAP-A). Using essentially an optimization technique based on the Hooke-Jeeves algorithm, the authors derived results on the form of the optimal buffer allocation for the exponential and Erlang-k service times. A few sample results for production lines with up to $K = 11$ stations and exponential and Erlang-2 service times are shown in Figure 5.2.

From an analysis of their results the authors derived two basic design rules, viz.,

- *Rule 1:* This is a confirmation and an extension of a rule earlier stated by Conway et al. (1998) to the effect that "*For the optimal buffer allocation of N buffer slots among the $K-1$ buffers of a K-station line, first allocate equally these N slots to all the $K-1$ buffers, i.e., allocate $n = [N/(K-1)]$ slots to each of the $K-1$ buffers, where $[x]$ is the maximum integer $\leq x$, and the remaining E buffer slots so that $E+1$ sub-lines with equal "distance" to be produced.*"
  Explanation of Rule 1: As there are E privileged buffers denoted by $B'_1, \ldots, B'_E$ (which receive one extra buffer slot above the uniform allocation), the original line may be divided into $E+1$ buffer sub-lines as follows:

$$B_1 \rightarrow B'_1; B'_1 \rightarrow B'_2; \cdots; B'_E \rightarrow B_K.$$

  The measure "distance," $D(B_i, Bj)$, is defined as the number of buffers in between buffer B_i and buffer B_j, not including buffer B_i and buffer B_j.

K	N=0mod(K-1)	N=1mod(K-1)	N=2mod(K-1)	N=3mod(K-1)	N=4mod(K-1)	N=5mod(K-1)	N=6mod(K-1)	N=7mod(K-1)	N=8mod(K-1)
3	(q,q)	(r,q)							
4	(q,q,q)	(q,r,q)	(q,r,r)						
5	(q,q,q,q)	(q,q,r,q)	(q,r,r,q)	(q,r,r,r)					
6	(q,q,q,q,q)	(q,q,r,q,q)	(q,r,q,r,q)	(q,r,r,r,q)	(q,r,r,r,r)				
7	(q,q,q,q,q,q)	(q,q,q,r,q,q)	(q,q,r,r,q,q)	(q,r,q,r,r,q)	(q,r,r,r,r,q)	(q,r,r,r,r,r)			
8	(q,q,q,q,q,q,q)	(q,q,q,r,q,q,q)	(q,q,r,q,r,q,q)	(q,r,q,r,q,r,q)	(q,r,r,q,r,r,q)	(q,r,r,r,r,r,q)	(q,r,r,r,r,r,r)		
9	(q,q,q,q,q,q,q,q)	(q,q,q,q,r,q,q,q)	(q,q,r,q,q,r,q,q)	(q,q,r,q,r,r,q,q)	(q,q,r,q,r,q,r,q)	(q,r,r,q,r,r,r,q)	(q,r,r,r,r,r,r,q)	(q,r,r,r,r,r,r,r,r)	
10	(q,q,q,q,q,q,q,q,q)	(q,q,q,q,r,q,q,q,q)	(q,q,q,r,q,r,q,q,q)	(q,q,r,q,r,q,r,q,q)	(q,q,r,r,q,r,r,q,q)	(q,r,q,r,r,r,q,r,q)	(q,r,r,r,q,r,r,r,q)	(q,r,r,r,r,r,r,r,q)	(q,r,r,r,r,r,r,r,r)
11	(q,q,q,q,q,q,q,q,q,q)	(q,q,q,q,q,r,q,q,q,q)	(q,q,q,r,q,q,r,q,q,q)	(q,q,q,r,q,r,q,r,q,q)	(q,q,r,q,r,r,q,r,q,q)	(q,q,r,r,q,r,r,q,r,q)	(q,r,q,r,r,r,q,r,r,q)	(q,r,r,r,q,r,r,r,r,q)	(*q,r,r,r,r,r,r,r,r,q*)

q : denotes the uniform allocation of the n = [N/K–1] slots whereas r denotes one extra buffer slot, i.e., r = q + 1

Given the integers N, i and j, the expression $N = i$ (**mod** j) means that N has as remainder the integer i when divided by the integer j

Fig. 5.2. Schematic representation of the form of optimal buffer allocation in terms of E and K of balanced production lines with exponential and Erlang-2 service times

- *Rule 2:* Each buffer that is allocated an extra slot must be closer to a central buffer that has been also allocated extra slot(s) rather than to a buffer lying toward the end stations which has also been alloted an extra slot.

However, Hillier and So (1995) noted that in BAP-A the optimal buffer allocation may begin to deviate from the uniform as possible allocation when the number of buffer slots available increases.

An allocation routine based on Rule 1 and Rule 2 (which have been obtained empirically) was developed and may be used to obtain the optimal buffer configuration.

With respect to μ-balanced unreliable production lines, Papadopoulos and Vidalis (1999) considered the buffer allocation problem, BAP-A, and in particular the effects of the distribution of service times, the availability (assumed identical) of the $m \leq K$ unreliable stations and of the repair rates on the throughput and the optimal buffer allocation. The assumptions of the model include single-machine stations, exponential or Erlang-k service times at each station and times to failure and repair times are all exponentially distributed with different mean rates. Complete enumeration was the search procedure used initially, but as the experimentation continued an efficient reduction search procedure was developed.

With the usual definition of the availability, A_i, of unreliable stations

$$A_i = \frac{r_i}{r_i + \beta_i},$$

where $1/r_i$ is the mean time to repair station i and $1/\beta_i$ is the mean time to failure of station i, some of the conclusions may be given as follows:

1. As far as the optimal buffer allocation (OBA) is concerned, there are three separate cases. For small values of the availability of the m unreliable stations ($m \leq K$):
   (i) when $m < K$ and even, the OBA resembles the shape of a bowl;
   (ii) when $m < K$ and odd, the OBA resembles the shape of a non-symmetric bowl and
   (iii) when $m = K$, the OBA resembles the shape of an "inverted bowl." This observation is in contrast with the well-known result about the uniformity of the optimal buffer allocation in a balanced line.

   In all three cases, as the availability (assumed identical) of the unreliable stations tends to unity, all the buffers are allocated evenly the buffer slots, at the optimal situation.
2. As the number of service phases increase (from exponential to Erlang-k ($k > 1$) distribution) then
   (i) the coefficient of variation (c.v.) of the effective service time decreases and this results in an increase in the throughput of the line;
   (ii) it becomes more difficult to justify economically the provision of extra buffer spaces, i.e., the marginal increase in throughput per buffer slot is decreasing;

(iii) the shape of the OBA as given in conclusions 1((i), (ii), (iii)) above become more pronounced and
(iv) there is a linear relationship between the value (assumed identical) of the c.v. of the service time distribution and the number of buffer slots required to achieve a given throughput.

In a further paper, Papadopoulos and Vidalis (2001a) have considered the buffer allocation in short unreliable and unbalanced production lines with $K \leq 6$ stations. Times to failure are assumed to be exponential, whereas service and repair times are assumed to follow any Erlang-k distribution, with $k \geq 1$. Single-machine stations are also assumed. An algorithm was developed to solve the buffer allocation problem in this type of production line. The algorithm is available at the website associated with this text with abbreviated name BA . This algorithm is a stand-alone optimization algorithm which cooperates with the MARKOV evaluative algorithm and consists of the following steps:

1. Preparation for a 'good' initial buffer allocation
   Order the stations $M_1, \ldots, M_K$ of the production line from the slower to the faster based on the value of the mean effective service rate, $e_i = \mu_i A_i = \mu_i \frac{r_i}{r_i+\beta_i}$, where μ_i is the mean service rate of station i, r_i is the mean repair rate of station i and β_i is the mean failure rate of station i. Let this arrangement be: $M'_1, \ldots, M'_K$.
2. Determination of a 'good' initial buffer allocation
   Apply the following linear buffer allocation scheme (LBAS):
   (i) The buffer that is located toward the center of the actual line and next to station M'_j is assigned a weight of $2(K+1-j)$;
   (ii) The buffer that is located toward the end of the line and next to station M'_j is assigned a weight of $2(K+1-j)-1$;
   (iii) The central buffer is assigned a weight of K (if K is odd) and when there are two central buffers (if K is even) these are equally weighted $K/2$.
3. Search for the OBA using a sectioning method Starting with the 'good' initial buffer vector determined in Step 2, above, use the sectioning (segmentation) routine to find the optimal or near optimal buffer vector. More specifically this search loop operates by increasing or decreasing by one unit each of the $K-1$ intial buffer decision variables and evaluating the throughput for the corresponding buffer allocation. The buffer allocation that gives the maximum throughput during this process is the initial buffer allocation for the next cycle of searches. A usual stopping criterion is adopted.

Example 1

A numerical example is given below to show the application of the above algorithm, taken from Papadopoulos and Vidalis (2001a).

Consider a four-station unreliable production line with the service and repair times following the two-stage Erlang distribution. The various parameters of this system have the following values: mean service rates: $\mu_1 = 3.7$, $\mu_2 = 1.5$, $\mu_3 = 1.1$,

$\mu_4 = 3$; mean repair rates: $r_1 = 0.17$, $r_2 = 0.37$, $r_3 = 0.78$, $r_4 = 0.5$; mean failure rates: $\beta_1 = 0.07$, $\beta_2 = 0.11$, $\beta_3 = 0.49$, $\beta_4 = 0.19$. Find the optimal buffer allocation of $N = 9$ total buffer slots among the three intermediate buffer locations of the production line which maximizes its throughput.

Step 1: Ordering of stations

By application of the relevant formulae, $e_i = \mu_i A_i$, where A_i, e_i denote, respectively, the availability and the mean effective service rate (efficiency) of the unreliable station i, $i = 1,\ldots,4$, one finds: $e_1 = 3.0683$, $e_2 = 1.3059$, $e_3 = 0.8371$, $e_4 = 2.521$. Therefore, the new ordering of the stations from the bottleneck station to the faster station is:

$$M_1' = M_3, \quad M_2' = M_2, \quad M_3' = M_4, \quad M_4' = M_1.$$

Step 2: Application of the LBAS

(i & ii) Give preferential treatment to the buffers that are close to the bottleneck stations. Buffer B_3 is assigned $2(4+1-1) = 8$ points, whereas buffer B_4 is assigned $2(4+1-1)-1 = 7$ points. Buffer B_3 is assigned $2(4+1-2) = 6$ points, whereas buffer B_2 is assigned $2(4+1-2)-1 = 5$ points. Buffer B_4 is assigned $2(4+1-3) = 4$ points and buffer B_2 is assigned $2(4+1-4) = 2$ points.

(iii) Give preferential treatment to the central buffer(s). There is only one central buffer, the B_3, which is assigned $K = 4$ points.

Adding all these points one may see that buffer B_2 is assigned 7 points (19.44%), buffer B_3 18 points (50%) and buffer B_4 11 points (30.56%). These percentages split the total number of buffer slots, $N = 9$ as follows:

$$B_2 = 1.75 \doteq 2, \quad B_3 = 4.5 \doteq 5, \quad B_4 = 2.75 \doteq 3.$$

However, this allocation gives a total $\sum_{i=2}^{K=4} B_i = 10 > 9 = N$. To overcome this, we subtract the one extra buffer slot from the buffer with the least priority, which in this case is buffer B_2. Thus, the initial buffer allocation vector is the $(1,5,3)$.

Step 3: Search for the optimal buffer allocation, OBA, via the application of the sectioning approach

This search process consists of 13 iterations, which are given in Table 5.1, to find the OBA which is the $(0,6,3)$ allocation. By complete enumeration, 55 iterations are needed to find the OBA.

Comments on Step 3: The initial buffer allocation is the $(1,5,3)$. The first cycle of iterations consists of three two-step searches (two searches per component of the buffer vector), i.e., #1 and #2 for buffer B_2, #3 and #4 for buffer B_3 and #5 and #6 for buffer B_4 (see Table 5.1). While keeping the buffer combination corresponding to highest throughput, in the first step, the value of the buffer component is increased by 1, whereas in the second step, this value is decreased by 1. The buffer allocation

Table 5.1. The 13 iterations to find the OBA in the production line of example 1

Iteration	Buffer allocation			Throughput
#	B_2	B_3	B_4	$X_{K=4}$
0	1	5	3	0.6453
1	2	4	3	0.6408
2	0	6	3	0.6471
3	0	7	2	0.6470
4	0	5	4	0.6431
5	0	5	4	0.6431
6	0	7	2	0.6470
7	0	6	3	0.6471
8	1	5	3	0.6453
9	0	7	2	0.6470
10	0	5	4	0.6431
11	0	5	4	0.6431
12	0	7	2	0.6470
13	0	6	3	0.6471

that gives the maximum throughput, $(0,6,3)$, is kept and forms the initial buffer allocation for the second cycle of iterations. This cycle consists of only five searches (#8 – #12) as the first element of the initial buffer allocation is zero and it can only take the value 1. The resulting allocations do not give higher throughput than allocation $(0,6,3)$ and thus the iteration procedure terminates at this point. There is no need to go further as the initial buffer allocation for the third cycle of searches is identical to that of the second cycle and therefore would lead to the same result.

This algorithm was found to give the exact optimal allocation in over 97% of the 373 experiments undertaken.

With respect to the problem of allocating buffer space with the objective of the minimization of the average work-in-process, $\overline{\text{WIP}}$, subject to a minimum required throughput (problem BAP-B, stated above), Papadopoulos and Vidalis (2001b) extended the work by So (1997) and showed:

- the choice of minimum throughput level has a critical impact on the minimum $\overline{\text{WIP}}$ achievable;
- A "self-similarity" phenomenon prevails in the case of balanced lines which significantly reduces the search space required to determine the buffer allocation associated with minimum $\overline{\text{WIP}}$.

Figure 5.3 and Figure 5.4 give the throughput and the $\overline{\text{WIP}}$, respectively, as a function of the ordered buffer allocations for a five-station production line with $N = 5$ buffer slots to be allocated among the 4 intermediate buffers, showing the "self-similarity" phenomenon. The details of the sequence of the ordered buffer allocations are given in Papadopoulos and Vidalis (2001b).

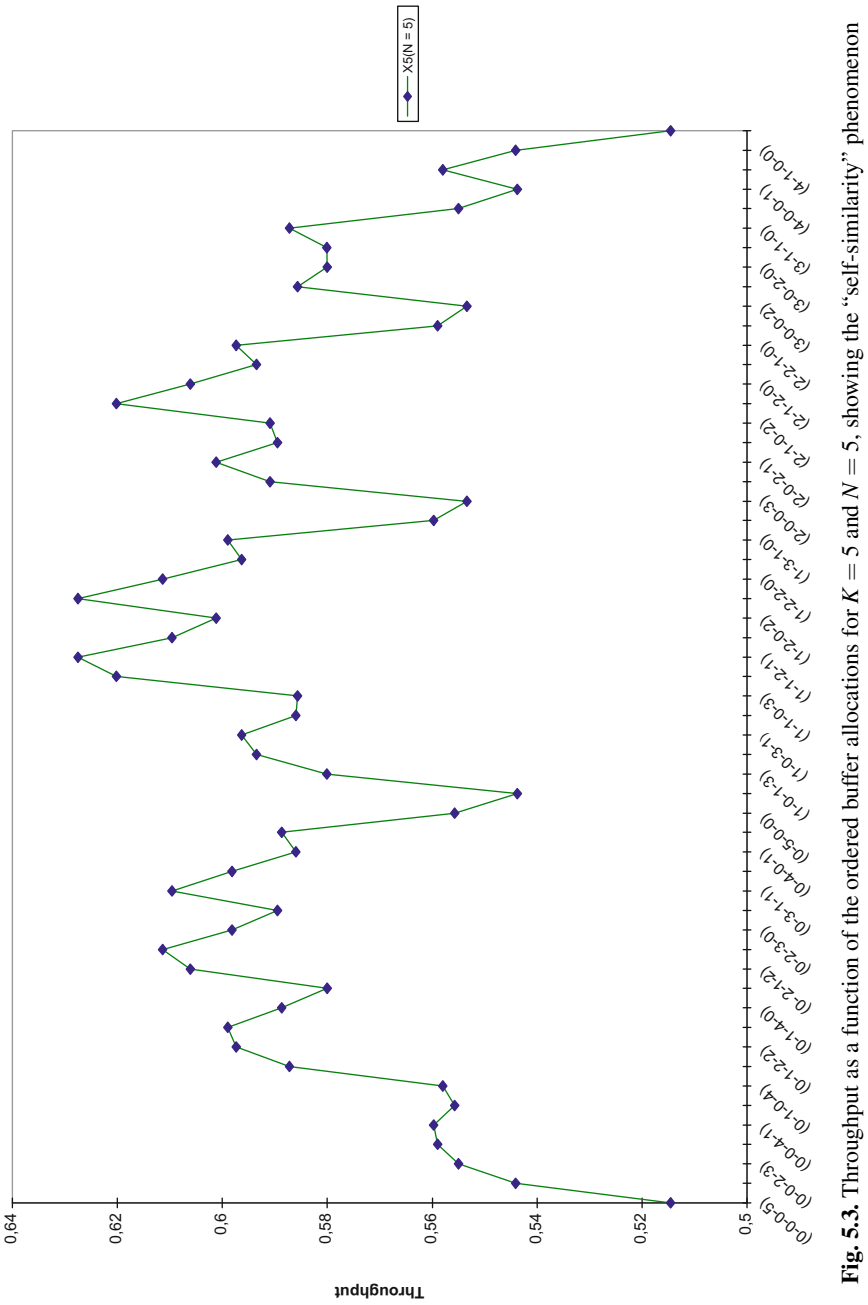


Fig. 5.3. Throughput as a function of the ordered buffer allocations for $K = 5$ and $N = 5$, showing the "self-similarity" phenomenon

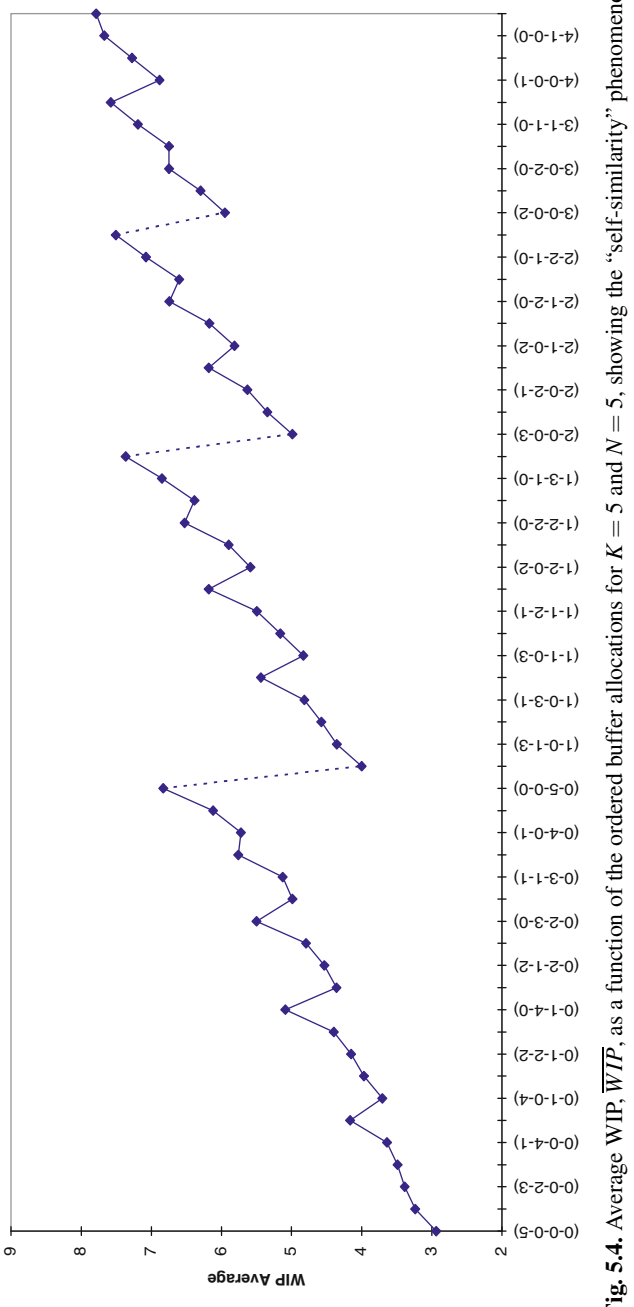


Fig. 5.4. Average WIP, $\overline{WIP}$, as a function of the ordered buffer allocations for $K = 5$ and $N = 5$, showing the "self-similarity" phenomenon

The authors found that in many cases the optimal buffer allocation (OBA) has a monotonic increasing characteristic of the buffer allocations, i.e., $B_K \geq B_{K-1} \geq B_{K-2} \geq \cdots$ and made observations on how to allocate the buffer slots when this property does not hold. An algorithm was developed to reduce the search space.

To understand the details of this algorithm, some definitions developed by the authors are required. For a production line with K stations and N buffer slots that are to be allocated among the $K-1$ intermediate buffers, let $\mathscr{B}$ denote the set of all possible buffer allocations.

$$\mathscr{B} = \{B_1, B_2, \ldots, B_L\},$$

where

$$L = \binom{N+K-2}{K-2} = \frac{(N+1)(N+2)\cdots(N+K-2)}{(K-2)!}$$

The B_i's are vectors with $K-1$ elements which are nonnegative integer numbers of the form

$$B_i = \{B_{i2}, B_{i3}, \ldots, B_{iK}\},$$

where $B_{ij},\ 2 \leq j \leq K$ expresses the capacity of the jth buffer.

Set $\mathscr{B}$ is split into $N+1$ *equivalence buffer classes* which are characterized as classes of *first generation*, the following: $[0], [1], [2], \ldots, [N]$. A class of first generation, say the $[I],\ 0 \leq I \leq N$, consists of all the allocations B_i with $B_{i2} = I$. All classes of first generation $I,\ 0 \leq I \leq N$, are divided into $N+1-I$ classes which are defined as classes of *second generation*. In turn, each class of second generation, e.g., class $[I,J],\ 0 \leq I \leq N, 0 \leq J \leq N+1-I$, consists of all the buffer allocations with the first two elements equal to I and J, respectively, and is divided into $N+1-(I+J)$ classes of *third generation* and so forth. Each element of the $(K-2)$ generation class specifies the contents of the last buffer.

Definition of *subsequent buffer classes*: Let $[\zeta_1, \zeta_2, \ldots, \zeta_\kappa]$ and $[\eta_1, \eta_2, \ldots, \eta_\kappa]$ be two buffer classes of the same generation $\kappa,\ 1 \leq \kappa \leq K-3$. We say that $[\eta_1, \eta_2, \ldots, \eta_\kappa]$ is subsequent to $[\zeta_1, \zeta_2, \ldots, \zeta_\kappa]$ if $\eta_i = \zeta_i$ for $i = 1, 2, \ldots, \kappa-1$ and $\eta_\kappa = \zeta_\kappa + 1$.

The input data to the algorithm consists of:

- K, the number of stations of the line,
- μ_i's, $i = 1, \ldots, K$, the mean service rates,
- N, the number of total buffer slots to be allocated among the $K-1$ buffers of the line and
- X_0, the minimum throughput level that has to be achieved.

The algorithm finds the integer values i_2, i_3, …, i_{K-2} which correspond to the classes of first, second, …, $(K-3)$rd generation up to which the average throughput increases and at which the throughput attains its maximum value. The steps of the algorithm are as follows:

Step 1: (Initialization phase)

Step 1.1: Put $B_2 = B_3 = \ldots = B_{K-2} = 0$ and search for the maximum value $i = i_{K-1}$ that buffer B_{K-1} can take such that for any $j = 0, 1, \ldots, N$:

$$X_K(0,0,\ldots,i,N-i) \geq X_K(0,0,\ldots,j,N-j).$$

i.e., throughput is maximized.

Step 1.2: Put $B_2 = B_3 = \ldots = B_{K-3} = 0$ and search for the maximum value $i = i_{K-2}$ that buffer B_{K-2} can take such that for any $j = 0, 1, \ldots, i_{K-1}$:

$$\max X_K(0,0,\ldots,i) \geq \max X_K(0,0,\ldots,j).$$

i.e., throughput is maximized.

Step 1.3: Find the upper values of the remaining buffers, i.e., the values $i_{K-3}, i_{K-4}, \ldots, i_3, i_2$ ($[x]$ denotes the maximum integer less than or equal to x):

$$\begin{aligned}
B_2 &= 1, \ldots, i_2 \left(= \left[\frac{N}{K-1} \right] \right), \\
B_2 &= 0, \ldots, i_3 (= i_4 - 1), \\
&\vdots = \vdots \\
B_{K-4} &= 0, \ldots, i_{K-4} (= i_{K-3} - 1), \\
B_{K-3} &= 0, \ldots, i_{K-3} (= i_{K-2} - 1), \\
B_K &= N - \sum_{j=2}^{K-1} B_j
\end{aligned}$$

Step 2: (Search phase) The algorithm searches for the optimal buffer allocation which minimizes the average WIP, $\overline{WIP}$, and gives a throughput that is greater than or equal to the specified level, X_0 in the reduced search space given by the values of $i_{K-1}, \ldots, i_2$.

Example 2

A numerical example is given below to show the application of the above algorithm, taken from Papadopoulos and Vidalis (2001b).

Consider a five-station reliable balanced production line with the service times following the exponential distribution. Find the optimal buffer allocation of $N = 5$ total buffer slots among the four intermediate buffer locations of the production line which minimizes the average WIP, $\overline{WIP}$, and gives a throughput that exceeds a given level $X_0 = 0.5961$.

Step 1.1: The upper value of buffer $B_{K-1} = B_4$, $i_{K-1} = i_4 = 4$, as the buffer allocation $(0,0,4,1)$ gives the maximum throughput, 0.5597, after 6 searches (of the respective number) of buffer allocations of class $[0,0]$ (see Table 5.2).

Table 5.2. Searching in classes $[0,0]$, $[0,1]$, $[0,2]$ and $[0,3]$

Iteration #	Equivalence buffer class	Buffer allocation	Throughput $X_{K=5}$	Average WIP $\overline{WIP}$
1		$(0,0,0,5)$	0.5146	
2		$(0,0,1,4)$	0.5441	
3	$[0,0]$	$(0,0,2,3)$	0.5550	
4		$(0,0,3,2)$	0.5590	
5		$(0,0,4,1)$	0.5597	
6		$(0,0,5,0)$	0.5557	
7		$(0,1,0,4)$	0.5580	
8		$(0,1,1,3)$	0.5872	
9	$[0,1]$	$(0,1,2,2)$	0.5974	4.1518
10		$(0,1,3,1)$	0.5990	4.3964
11		$(0,1,4,0)$	0.5887	
12		$(0,2,0,3)$	0.5800	
13		$(0,2,1,2)$	0.6061	4.5340
14	$[0,2]$	$(0,2,2,1)$	0.6114	4.7960
15		$(0,2,3,0)$	0.5982	5.5007
16		$(0,3,0,2)$	0.5895	
17	$[0,3]$	$(0,3,1,1)$	0.6096	5.1276
18		$(0,3,2,0)$	0.5982	5.7633

Step 1.2: The upper value of buffer $B_{K-2} = B_3$, $i_{K-2} = i_3 = 2$, as class $[0,2]$ gives the maximum throughput, 0.6114, after 12 searches: 5 in class $[0,1]$, 4 in class $[0,2]$ and 3 in class $[0,3]$ (see Table 5.2).

Step 1.3: The upper value of buffer $B_{K-3} = B_2$, $i_{K-3} = i_2 = 1 = [5/(5-1)]$ ($[x]$ is the largest integer less than or equal x). This is the end of the initialization phase.

Step 2: The values of the buffers are determined:

$$B_2 = 1, \quad B_3 = 0,1,2, \quad B_4 = 0,1,2,3,4, \quad B_5 = N - \sum_{i=2}^{4} B_i.$$

Comments

Since class $[0]$ was already checked in Steps 1.1 and 1.2, B_2 takes only the value 1. The number of iterations (searches) in Step 2 are the following 12: 5 in class $[1,0]$, 4 in class $[1,1]$ and 3 in class $[1,2]$ (see Table 5.3).

Again none of all these iterations gives $\overline{WIP}$ less than 4.1518, the minimum value found so far. Thus, the buffer allocation that minimizes the $\overline{WIP}$ is $(0,1,2,2)$ and the corresponding minimum $\overline{WIP}$ is 4.1518 for the selected throughput level $X_0 = X_{0,2} = 0.5961$ that has to be exceeded. The total numer of searches from all steps of the algorithm is 30 as compared with the 56 allocations from enumeration. This means that the algorithm leads to a 46% reduction in the number of searches to find the optimal buffer allocation.

From experimental work, the percentage reduction of the search space was found to be generally over 50%.

Table 5.3. Searching in classes $[1,0]$, $[1,1]$ and $[1,2]$

Iteration #	Equivalence buffer class	Buffer allocation	Throughput $X_{K=5}$	Average WIP $\overline{WIP}$
19		$(1,0,0,4)$	0.5438	
20		$(1,0,1,3)$	0.5801	
21	$[1,0]$	$(1,0,2,2)$	0.5935	
22		$(1,0,3,1)$	0.5963	4.8100
23		$(1,0,4,0)$	0.5860	
24		$(1,1,0,3)$	0.5857	
25		$(1,1,1,2)$	0.6202	5.1638
26	$[1,1]$	$(1,1,2,1)$	0.6275	5.4941
27		$(1,1,3,0)$	0.6096	6.1794
28		$(1,2,0,2)$	0.6012	5.5889
29	$[1,2]$	$(1,2,1,1)$	0.6275	5.8978
30		$(1,2,2,0)$	0.6114	6.5231

5.4 Solution Approaches to the BAP in Longer Lines

It should be noted that optimization procedures used in long lines (with $K > 6$ stations in series) may also be used for shorter lines. Such procedures were not considered in Section 5.3 because, with respect to shorter lines, exact results are available by way of enumeration. The optimization techniques used for longer lines in effect are approximate techniques. Limitations of the evaluative technique used are crucial in determining which technique to use in short or longer lines. Basically, as far as short lines are concerned, Markov-based evaluative techniques are adequate and accurate, whereas in longer lines approaches based on Markovian analysis are not applicable because of the required computer storage issues. Decomposition methods extend greatly the range of cases which may be evaluated but at the cost of some reduction in accuracy. However, this reduction in accuracy may not be significant in practical situations. To achieve a solution to the system in an efficient manner, it is necessary for both the evaluative technique and the optimization technique to be efficient (see Figure 5.1 which shows the feedback loop between the two techniques used in determining the solution to the system). This leads to the exclusion of simulation as a possible optimization technique in large systems. The decomposition approach (described in Chapter 2) is the main evaluative technique used in the analysis of large lines. Associated with this evaluative approach, there are a number of different choice possibilities with respect to the optimization techniques. The latter include:

- Gradient methods
- Dynamic programming
- Simulated annealing
- Genetic algorithms and
- Tabu search

Gradient methods

The well-known numerical analysis approach, *gradient method* (see Ho et al., 1979 and Gershwin and Schor, 2000), may be adopted to solve problem BAP-A, and the steps of an appropriate algorithm are given below:

Step 1: Specify an admissible set of initial buffer allocations and use the evaluative technique to determine the throughput $X_K(B_2, B_3, \ldots, B_K)$ of the line.

Step 2: Calculate the gradient $\mathbf{g} = (g_2, g_3, \ldots, g_K)$ given by

$$g_i = \frac{X(B_2, \ldots, B_i + \delta B_i, \ldots, B_K) - X(B_2, \ldots, B_i, \ldots, B_K)}{\delta B_i}$$

where δB_i is a step size of integer value.

Step 3: Obtain the search direction v by projecting the gradient $\mathbf{g} = (g_2, g_3, \ldots, g_K)$ on to the hyperplane such that:

$$\sum_{i=2}^{K} \delta B_i = 0$$

giving $v_i = g_i - \bar{g}$, where,

$$\bar{g} = \frac{1}{K-1} \sum_{i=2}^{K} g_i.$$

Step 4: Find Λ such that $X(B + \Lambda v)$ is maximized.

Step 5: Define $B' = B + \Lambda v$.

Step 6: If B' is close to B, Stop (B is the optimal buffer allocation), otherwise, let $B = B'$ and go to Step 2 and continue.

A difficulty of the above procedure arises if $B_i = 0$ and $v_i < 0$ for some i. In that case, the new direction is calculated by deleting the ith component of $\mathbf{g}$ and setting $v_i = 0$. The other components of v are determined as before and the process is continued until a feasible v is determined or all components of $\mathbf{g}$ are deleted.

In Gershwin and Schor (2000), an algorithm for the solution of the BAP-B problem, designated primal by the authors, using the solution of the BAP-A problem is given. Gershwin and Schor noted that the maximum throughput as a function of the total buffer space, N, for a three-station two-buffer system is monotonically increasing in N and may be approximated by two linear functions with different slopes. This observation is crucial to the development of an algorithm for the solution of the BAP-B problem, the steps of which are given below.

Step 1: Let $N^0 = (0, 0, \ldots, 0)$. Calculate $X_{\max}(0, 0, \ldots, 0)$ using an appropriate evaluative method (Markovian for short lines and decomposition for longer lines).

Step 2: Specify a new estimate of the total buffer slots to be allocated, N^1 and solve problem BAP-A (designated as dual problem by Gershwin and Schor) using the gradient algorithm given above. Set $j = 2$.

Step 3: Calculate N^j from

$$N^j = aN^{j-1} + bN^{j-2},$$

where a and b are determined from the assumed linear approximation between maximum throughput and total buffer size. Note that because of the two linear approximations, the slope of the line will be different from one range of total buffer size to another.

Step 4: Use the gradient algorithm to determine $X_{\max}(N^j)$.

Step 5: If $X_{\max}$ is sufficiently close to the desired throughput level, stop, otherwise return to Step 3.

The solution specifies N, the total number of buffer slots, and the distribution of N among the intermediate $K-1$ buffers of the production line.

Dynamic programming

In general, dynamic programming, DP, is a multi-stage decision process where the objective is to allocate a limited resource sequentially over the stages so as to optimize the objective function based on Bellman's principle of optimality in that, at any stage in the analysis, the optimal solution involves being on an optimal path from that stage onwards. In general terms, with respect to problem BAP-B the stations are considered to be the stages and the total number of buffer slots to be allocated are considered to be the states. An appropriate recursive relationship must be developed having in mind the overall objective function of minimizing the total number of buffer slots to be allocated among the intermediate buffers of the line subject to a minimum throughput. The structure of the BAP-B problem may be utilized to effect computational efficiencies with a dynamic programming approach.

Simulated annealing

Simulated annealing, SA, is an adaptation of the simulation of physical thermodynamic annealing principles to combinatorial optimization problems. It follows a logical improvement paradigm having regard to the exponential complexity of the solution space. The algorithm is based on randomization and as a result there is no certainty in relation to achieving the precise optimal solution. Spinellis and Papadopoulos (2000a) used a simulated annealing approach for the solution of the buffer allocation problem BAP-A in reliable (large) production lines. The authors assumed exponential processing times at each station with mean service rates μ_i, $i = 1,\ldots,K$. The exact numerical algorithm of Heavey, Papadopoulos and Browne (1993) and the decomposition algorithm A3 of Dallery and Frein (1993) were used in conjunction with the simulated annealing algorithm developed by the authors. In the improvement process of the simulated annealing approach, the probabilistic "uphill" energy movement avoids the entrapment of the solution in a local minimum. The authors gave the correspondence between annealing in the physical world and simulated annealing used in the optimal buffer allocation as shown in Table 5.4.

The associated simulated annealing algorithm is given in Figure 5.5.

Table 5.4. Correspondence between annealing in the physical world and simulated annealing used for production line optimization

Physical World	Production Line Optimization
Atom placement	Line configuration
Random atom movements	Buffer space, server, work-load movement
Energy, E	Throughput, X
Energy differential, ΔE	Configuration throughput differential, ΔX
Energy state probability distribution	Changes according to the Metropolis criterion, $\exp(\frac{-\Delta E}{T}) > \text{rand}(0\ldots1)$, implementing the Boltzmann probability distribution
Temperature	Variable for establishing configuration acceptance termination

The SA procedure was run by Spinellis and Papadopoulos (2000a) with the following characteristics based on the number of stations K:

Maximum trials at given temperature: $100 \times K$
Maximum successes at given temperature: $10 \times K$
Initial temperature: 0.5
Cooling schedule: Exponential: $T_{J+1} = 0.9T_J$
Initial line configuration: Equal division of buffers and servers among stations with any remaining resources placed on the station(s) in the middle
Reported time: Elapsed wall clock time in seconds.

As the reader will note, the algorithm given in Figure 5.5 is a simulated annealing algorithm for not only distributing N total buffer slots but also simultaneously $S(S \geq K)$ servers and work-load normalized to K.

The algorithm is available at the website associated with this text with abbreviated name SA and may be accessed by selecting the corresponding generative/optimization algorithm.

The authors demonstrated the accuracy of the simulated annealing approach compared to the complete enumeration for shorter lines with up to 9 stations, and in the case of longer lines (15 stations), a comparison was made between the solutions obtained using reduced enumeration and decomposition and simulated annealing and decomposition. For comparing the results, the authors used the formalism $S(G,E)$ to describe a closed loop system using the generative method G and the evaluative method E. A major contribution of this work was that solutions (buffer allocations) were achieved using a simulated annealing for very large lines (with up to 400 stations). Clearly, only decomposition may be used as evaluative model in these cases. With respect to these large lines it should be noted that the time requirements of the simulated annealing method becomes competitive with other methods as the number of stations and the available buffer size increases as indicated in Figure 5.6 and Figure 5.7. Figure 5.8 shows that the number of enumerations required for simulated annealing solutions increases linearly with the number of stations.

1. [Set initial line configuration.] Set $B_i \leftarrow \lfloor N/(K-1) \rfloor$,[1] set $B_{K/2} \leftarrow B_{K/2} + N - \sum_{i=2}^{K} \lfloor N/(K-1) \rfloor$, set $s_i \leftarrow \lfloor S/K \rfloor$, set $s_{K/2} \leftarrow s_{K/2} + S - \sum_{i=1}^{K} \lfloor S/K \rfloor$, set $w_i \leftarrow 1/K$.
2. [Set initial temperature T_0.] Set $T \leftarrow 0.5$.
3. [Initialize step and success count.] Set $I \leftarrow 0$, set $U \leftarrow 0$.
4. [Create new line with a random redistribution of buffer space, servers, or work-load.]
   (i) [Create a copy of the configuration vectors.] Set $\mathbf{n}' \leftarrow \mathbf{n}$, set $\mathbf{s}' \leftarrow \mathbf{s}$, set $\mathbf{w}' \leftarrow \mathbf{w}$.
   (ii) [Determine which vector to modify.] Set $\ell \leftarrow \lfloor \text{rand}[0 \ldots 2] \rfloor$.
   (iii) if $\ell = 0$ [Create new line with a random redistribution of buffer space.] Move r_n space from a source buffer B_{r_s} to a destination buffer B_{r_d}: set $r_s \leftarrow \lfloor \text{rand}[2 \ldots K) \rfloor$, set $r_d \leftarrow \lfloor \text{rand}[2 \ldots K) \rfloor$, set $r_n \leftarrow \lfloor \text{rand}[1 \ldots B_{r_s} + 1) \rfloor$, set $B_{r_s} \leftarrow B_{r_s} - r_n$, set $B_{r_d} \leftarrow B_{r_d} + r_n$.
   (iv) if $\ell = 1$ [Create new line with a random redistribution of server allocation.] Move r_n servers from source station s_{r_s} to a destination station s_{r_d}: set $r_s \leftarrow \lfloor \text{rand}[1 \ldots K) \rfloor$, set $r_d \leftarrow \lfloor \text{rand}[1 \ldots K) \rfloor$, set $r_n \leftarrow \lfloor \text{rand}[1 \ldots s_{r_s} + 1) \rfloor$, set $s_{r_s} \leftarrow s_{r_s} - r_n$, set $s_{r_d} \leftarrow s_{r_d} + r_n$.
   (v) if $\ell = 2$ [Create new line with a random redistribution of work-load.] Move r_n work-load from source station s_{r_s} to a destination station s_{r_d}: set $r_s \leftarrow \lfloor \text{rand}[1 \ldots K) \rfloor$, set $r_d \leftarrow \lfloor \text{rand}[1 \ldots K) \rfloor$, set $r_n \leftarrow \text{rand}(0 \ldots w_{r_s}]$, set $w_{r_s} \leftarrow w_{r_s} - r_n$, set $w_{r_d} \leftarrow w_{r_d} + r_n$.
5. [Calculate energy differential.] Set $\Delta E \leftarrow X(\mathbf{n}, \mathbf{s}, \mathbf{w} - X(\mathbf{n}', \mathbf{s}', \mathbf{w}')$.
6. [Decide acceptance of new configuration.] Accept all new configurations that are more efficient and, following the Boltzmann probability distribution, some that are less efficient: if $\Delta E < 0$ or $\exp(\frac{-\Delta E}{T}) > \text{rand}(0 \ldots 1)$, set $\mathbf{n} \leftarrow \mathbf{n}'$, set $\mathbf{s} \leftarrow \mathbf{s}'$, set $\mathbf{w} \leftarrow \mathbf{w}'$, set $U \leftarrow U + 1$.
7. [Repeat for current temperature.] Set $I \leftarrow I + 1$. If $I <$ *maximum number of steps*, go to step 4.
8. [Lower the annealing temperature.] Set $T \leftarrow cT$ $(0 < c < 1)$.
9. [Check if progress has been made.] If $U > 0$, go to step 3; otherwise the algorithm terminates.

[1] We employ the notation for open intervals $(a \ldots b)$, closed intervals $[a \ldots b]$, and the floor function $\lfloor x \rfloor$ to create sets of random numbers in a given range. For this purpose we use a function rand() generating random real numbers in the range specified by an interval. Specifically, $\lfloor \text{rand}(a \ldots b) \rfloor$ will produce a random integer $x : a < x \leq b$, and $\lfloor \text{rand}[a \ldots b] \rfloor$ will produce a random integer $x : a \leq x \leq b$. The half closed intervals, $[a \ldots b)$ and $(a \ldots b]$, work in a similar way.

Fig. 5.5. Simulated annealing algorithm for distributing N buffer space, S servers, and K work-load in a K-station line

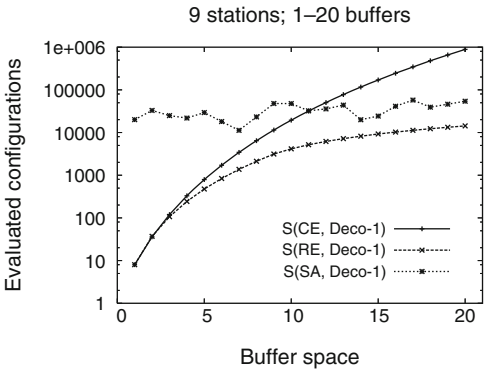


Fig. 5.6. Performance of simulated annealing S(SA, Deco) compared with complete S(CE, Deco) and reduced S(RE, Deco) enumerations for 9 stations (left, middle) (Note the $\log_{10}$ scale on the ordinate axis)

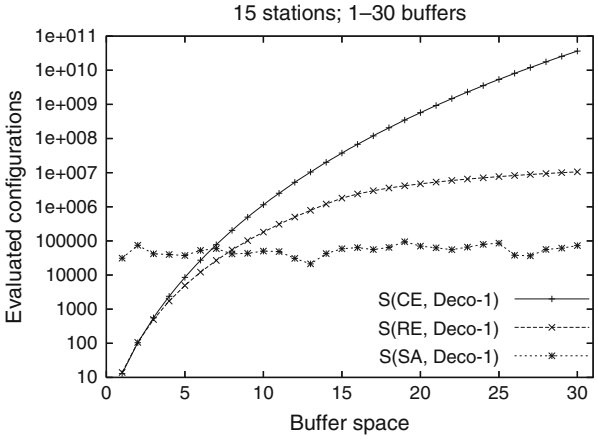


Fig. 5.7. Performance of simulated annealing S(SA, Deco) compared with complete S(CE, Deco) and reduced S(RE, Deco) enumerations for 15 stations (left, middle) (Note the $\log_{10}$ scale on the ordinate axis)

Genetic algorithms

Another optimization method for solving the buffer allocation problem is based on genetic algorithms. These are global stochastic optimization techniques that avoid a number of the shortcomings exhibited by local search techniques on difficult search spaces and are based on the mechanics of natural selection and genetics. Spinellis and Papadopoulos (2000b) used this approach for reliable lines with exponentially

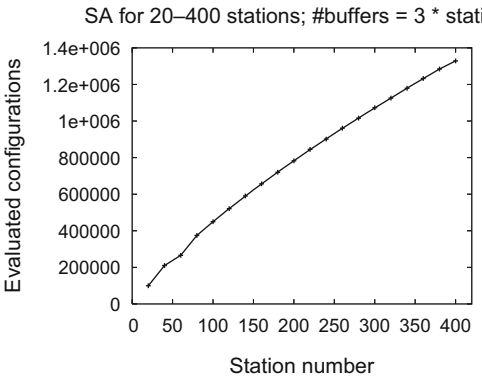


Fig. 5.8. Number of enumerations required for simulated annealing vs. the number of stations (Note the $\log_{10}$ scale on the ordinate axis)

distributed service times with mean service rates $\mu_i, i = 1, \ldots, K$ for the solution of the BAP-A problem.

Genetic algorithms rely on modeling the buffer allocation problem as a population of organisms. Each organism represents a possible valid solution to the problem. Organisms are composed of alleles representing parts of a given solution. In each iteration which corresponds to a generation, a new population is created by retaining all solutions and generating new solutions from the previous population using three basic genetic operators, viz., reproduction operator, crossover operator and mutation operator.

The genetic algorithm for solving the BAP-A problem is available at the website associated with this text with abbreviated name GA. This algorithm may be described in the following steps:

1. [Initialize a population of size M.] Set $P_{0\ldots M,0\ldots N} \leftarrow \lfloor \text{rand}[0 \ldots K-1) \rfloor$.
2. [Evaluate population members creating throughput vector $\mathbf{X}$.] For $i \leftarrow 0 \ldots M$: set $X_i \leftarrow X_K(\mathbf{P_i})$.
3. [Create roulette selection probability vector $\mathbf{R}$.] Set $R_i \leftarrow \sum_{j=0}^{i}(X_j / \sum_{k=0}^{M} X_k)$.
4. [Create new population using crossovers from the previous population.] For $i \leftarrow 0 \ldots M$: if $\text{rand}[0 \ldots 1) <$ crossover rate, set $c \leftarrow \lfloor \text{rand}[0 \ldots M) \rfloor$, set $P'_{i,0\ldots c} \leftarrow P_{R_r,0\ldots c}$, set $P'_{i,c+1\ldots N} \leftarrow P_{R_r,c+1\ldots N}$; otherwise set $\mathbf{P}'_{\mathrm{i}} \leftarrow \mathbf{P}_{\mathbf{R}_{\mathrm{r}}}$ by selecting each r using the roulette selection probability vector so that $R_r \leq \text{rand}[0 \ldots 1) < R_{r+1}$.
5. [Introduce mutations.] For $i \leftarrow 0 \ldots M$: for $j \leftarrow 0 \ldots N$: if $\text{rand}[0 \ldots 1) <$ mutation rate, set $P'_{i,j} \leftarrow \lfloor \text{rand}[0 \ldots K-1) \rfloor$.
6. [Keep fittest organism for elitist selection strategy.] Select k so that $X_k \geq X_{0\ldots M}$, set $\mathbf{P}'_{\lfloor \text{rand}[\mathbf{0}\ldots\mathbf{M}) \rfloor} \leftarrow \underline{P_k}$.
7. [Make new population the current population.] Set $\mathbf{P} \leftarrow \mathbf{P}'$.
8. [Loop based on the population's variance.] If $\sum_{i=0}^{P} |X_k - X_i| >$ maximum variance go to step 2; otherwise the algorithm terminates with the optimal line setup in $\underline{P_k}$.

The implementation of genetic algorithms may be tuned using different parameters. Spinellis and Papadopoulos used the parameters that Grefenstette (1986) derived:

- a populations size of 50,
- a crossover rate of 0.6,
- a mutation rate of 0.0001,
- a generation gap of 1 (the entire population is replaced during each generation),
- no scaling window, and
- an elitist selection strategy (the organism with the best performance survives intact into the next generation).

The crossover points, the mutation rates and the selection of organisms are produced using the subtractive method algorithm described by Knuth (1981). The decomposition method was used as the evaluative method for calculating throughput. From this work it appears that simulated annealing approach is slower than the genetic algorithm in terms of computer time, but as the simulated annealing results in a higher throughput using the same evaluative method it could be argued that simulated annealing is the more accurate method. Details are shown in Figures 5.9 and 5.10, respectively.

Tabu search algorithm

According to Glover and Laguna (1998): "The word tabu or taboo comes from Tongan, a language of Polynesia, where it was used by the aborigines of Tonga island to indicate things that cannot be touched because they are sacred. According to

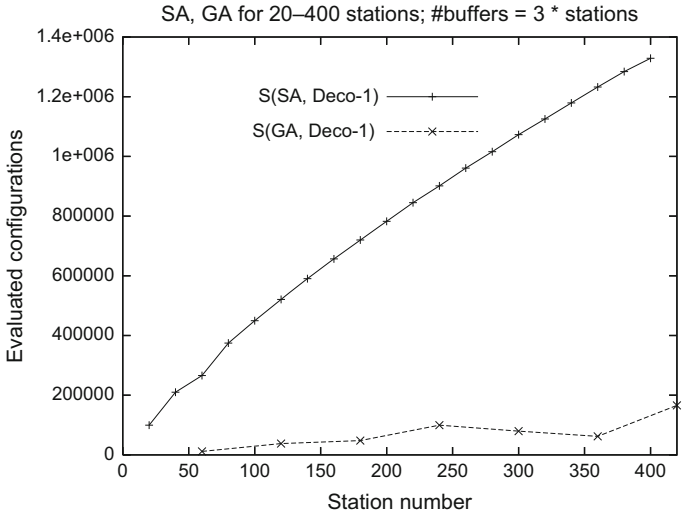


Fig. 5.9. Performance of simulated annealing S(SA, Deco) compared with genetic algorithms S(GA, Deco) for large production lines

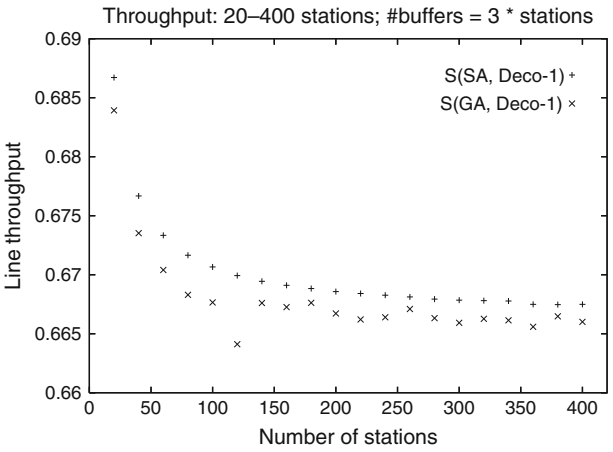


Fig. 5.10. Accuracy of simulated annealing S(SA, Deco) compared with genetic algorithms S(GA, Deco) for large production lines

Webster's dictionary, the word now means a prohibition imposed by social custom as a protective measure or of something banned as constituting a risk. These current more pragmatic senses of the word accord well with the theme of tabu search. The risk to be avoided in this case is that of following a counter-productive course, including one which may lead to entrapment without hope of escape. On the other hand, as in a broader social context where protective prohibitions are capable of being superseded when the occasion demands, the tabus of tabu search are to be overruled when evidence of a preferred alternative becomes compelling."

Tabu search was first introduced by Glover (1986) and a few of its basic ideas were also given by Hansen (1986).

Let $\mathbf{x}$ be a certain initial solution from a set of solutions, Λ, and $G(\mathbf{x})$ be a neighborhood of this solution. The solution is feasible if it satisfies a certain set of constraints. Let also $\mathbf{x}^*$ denote the best solution found so far, i be an iteration counter and Λ^* be a subset of solutions in the neighborhood $G(\mathbf{x})$. $f(\mathbf{x})$ denotes a function which is sought to be optimized, e.g., to be minimized. Tabu search is an iterative method more sophisticated than the ordinary descent method in at least two dimensions, as follows:

(i) It makes systematic use of memory in order to avoid re-visiting the same solutions considered previously.
(ii) It uses an elaborate exploration process in order to escape from a local minimum by using intensification and diversification. Intensification's role is to ensure that the next solution in the search process is close enough to the current solution when both of them have common features. This can be achieved by adding an extra term in the objective function and penalizing solutions which are far from the current solution. On the other hand, the role of diversification is exactly the opposite one, viz., to guarantee that the next solution is far from the current

one when it is discovered that this solution does not have the desired features. Mathematically, diversification is carried out by inserting another term in the objective function penalizing solutions that are close to the current solution. By performing intensification and diversification, the initial objective function, f, is modified to the following modified function, f_m, which may be written as:

$$f_m(\mathbf{x}) = f(\mathbf{x}) + \text{intensification} + \text{diversification}.$$

When a solution visited at a stage is not acceptable, then it has to be removed from the neighborhood $G(\mathbf{x})$ as it is considered a tabu solution which should be avoided in the future iterations of the search process. That way a tabu list, T_ℓ, $\ell = 1,\ldots,t$, is created which is a collection of tabu conditions, $t_\ell(\mathbf{x},\mathbf{h})$ which are some constituents $t_\ell(\mathbf{x},\mathbf{h})$ that are given a tabu status to indicate that these constituents are currently not allowed to be involved in the next solution. t_ℓ are functions of solution $\mathbf{x}$ or $\mathbf{h}$, where $\mathbf{h}$ is a move applied to the solution $\mathbf{x}$ to direct it to a new solution, say, $\mathbf{y} = \mathbf{x} \bigoplus \mathbf{h}$ (symbol $\bigoplus$ denotes the application of move $\mathbf{h}$ to $\mathbf{x}$ to obtain $\mathbf{y}$). This move $\mathbf{h}$ is said to be a tabu move if all conditions are satisfied. (Usually, reversible moves are used. A move $\mathbf{h}$ is called a reversible move when there exists a move $\mathbf{h}^{-1}$ such that: $(\mathbf{x} \bigoplus \mathbf{h}) \bigoplus \mathbf{h}^{-1} = \mathbf{x}$.) However, sometimes, tabu move h may appear attractive because it gives a solution better than the best solution found so far in the search process. In that case, one would like to accept h in spite of its status. This may be done if it has an aspiration level, $\alpha_\ell(\mathbf{x},\mathbf{h})$, which is better than $A_\ell(\mathbf{x},\mathbf{h})$, a given threshold value. If at least one of the following conditions:

$$\alpha_\ell(\mathbf{x},\mathbf{h}) \in A_\ell(\mathbf{x},\mathbf{h}), \quad \ell = 1,\ldots,\alpha,$$

is satisfied by the tabu move $\mathbf{h}$ applied to solution $\mathbf{x}$, then move $\mathbf{h}$ will be accepted. Tabu search uses a tabu list with variable size to prevent cycling of the solutions.

Following the lines of Hertz, Taillard and de Werra (1995), the steps of the Tabu search, TS, are summarized below:

Step 1: Choose an initial solution $\mathbf{x}$ in Λ. Set $\mathbf{x}^* = \mathbf{x}$ and $i = 0$.
Step 2: Set $i = i+1$ and generate a subset Λ^* of solution in $G(\mathbf{x},i)$ such that either one of the tabu conditions $t_\ell(\mathbf{x},\mathbf{h}) \in T_\ell$ is violated ($\ell = 1,\ldots,t$) or at least one of the aspiration conditions $\alpha_\ell(\mathbf{x},\mathbf{h}) \in A_\ell(\mathbf{x},\mathbf{h})$, $\ell = 1,\ldots,\alpha$, holds.
Step 3: Choose a best $\mathbf{y} = \mathbf{x} \bigoplus \mathbf{h}$ in Λ^* with respect to $f(\mathbf{x})$ or to the modified function $f_m(\mathbf{x})$ and set $\mathbf{x} = \mathbf{y}$.
Step 4: If $f(\mathbf{x}) < f(\mathbf{x}^*)$ then set $\mathbf{x} = \mathbf{x}^*$.
Step 5: Update tabu and aspiration conditions.
Step 6: If a stopping condition is met then stop. Else go to Step 2.

Many applications of tabu search are given in the book by Glover, Taillard, Laguna and de Werra (1992).

Summary

The following summary may be of value to the reader who wishes to use the software available at the website associated with this book in solving buffer allocation problems.

Buffer Allocation Problem (BAP)

1. **BA**
   - For short unreliable production lines with Erlang-k ($k \geq 1$) service and repair times and exponential times to failure and intermediate buffers.
2. **DECO-1** and **SA/GA**
   - For large reliable exponential production lines with single machine stations and finite intermediate buffers.
3. **DECO-2** and **SA/GA**
   - For large reliable exponential production lines with multiple parallel identical machine stations and finite intermediate buffers.

5.5 Related Bibliography

Conway et al. (1988), considering problem BAP-A and based on simulation, conjectured, among other results, that the loss of throughput due to interference between stations in production lines occurs in the first few stations and that the variability of the service times has a major impact on this phenomenon and there is a decreasing marginal advantage with placing buffers between work-stations. They demonstrated the existence of the "inverted bowl" phenomenon and suggested that buffers in unbalanced lines should be placed toward the bottleneck station.

Powell (1994) dealt with problem BAP-A for three-station production lines with the service time probability distributions differing in either or both mean and variance. He developed rules of thumb for the optimal buffer allocation and showed that a balanced or nearly balanced allocation is optimal for many highly unbalanced lines. The author also demonstrated that an imbalance in means has a greater impact on buffer allocation than does an imbalance in variances. He used simulation as an evaluative technique.

A knowledge-based approach to problem BAP-A was given in Vouros and Papadopoulos (1998).

Ho et al. (1979) dealt with a gradient method for solving the BAP-A problem.

Chow (1987) proposed a dynamic programming algorithm to solve the buffer allocation problem BAP-A.

Jensen et al. (1991) also dealt with the buffer optimization problem in serial and diverging-branch (non-linear) configurations of production systems. They aplied a classical dynamic programming algorithm for solving the problem by taking into account production system costs.

Yamashita and Altiok (1998) solved the BAP-B problem by applying a dynamic programming algorithm associated with Altiok's (1989) decomposition method for analyzing the production line.

Kubat and Sumita (1985) and Jafari and Shanthikumar (1989) also used dynamic programming approaches for solving the BAP in automatic transfer lines.

Papadopoulos and Karagiannis (2001) and Spinellis and Papadopoulos (2000b) developed a genetic algorithm for solving the buffer allocation problem in unreliable and reliable production lines, respectively, with exponentially distributed service completion times.

Colledani, Matta, Grasso and Tolio (2005) developed a new analytical method for the buffer space allocation problem in production lines.

Levantesi, Matta and Tolio (2001) presented a new search algorithm which in conjunction with a decomposition method for the performance evaluation of the production lines solved the buffer allocation problem very fast.

Singh and MacGregor Smith (1997) dealt with the buffer allocation problem in production lines with multiple parallel machines at each work-station.

Additional works on tabu search are given in Glover (1989), de Werra and Hertz (1989),Glover (1990). Theoretical aspects of tabu search are presented in Faigle and Kern (1992), Glover (1992) and Fox (1993).

References

1. Altiok, T. (1989), Approximate analysis of queues in series with phase-type service times and blocking, *Operations Research*, Vol. 37, pp. 601–610.
2. Altiok, T. and Stidham, S. Jr. (1983), The allocation of interstage buffer capacities in production lines, *IIE Transactions*, Vol. 15, No. 4, pp. 292–299.
3. Anderson, D.R. and Moodie, C.L. (1989), Optimal buffer storage capacity in production line systems, *International Journal of Production Research*, Vol. 7, No. 3, pp. 233–240.
4. Bulgak, A.A., Diwan, P.D., and Inozu, B. (1995), Buffer size optimization in asynchronous systems using genetic algorithms, *Computers and Industrial Engineering*, Vol. 28, No. 2, pp. 309–322.
5. Cheng, D.W. (1994), On the design of a tandem queue with blocking: Modeling analysis, and gradient estimation, *Naval Research Logistics*, Vol. 41, pp. 759–770.
6. Chow, W.-M. (1987), Buffer capacity analysis for sequential production lines with variable process times, *International Journal of Production Research*, Vol. 25, No. 8, pp. 1183–1196.
7. Colledani, Matta, A., Grasso, M., and Tolio, T. (2005), A new analytical method for buffer space allocation in production lines, *CIRP–Journal of Manufacturing Systems*, Vol. 34, No. 4.
8. Conway, R., Maxwell, W., McClain, J.O., and Thomas, L.J. (1988), The role of work-in-process inventory in serial production lines, *Operations Research*, Vol. 36, No. 2, pp. 229–241.
9. Dallery, Yves and Frein, Yannick (1993), On decomposition methods for tandem queueing networks with blocking, *Operations Research*, Vol. 41, No. 2, pp. 386–399.
10. de Werra, D. and Hertz, A. (1989), Tabu Search techniques: A tutorial and an application to neural networks, *OR Spektrum*, pp. 131–141.
11. Enginarlar, E., Li, J., Meerkov, S., and Zhang, Q. (2002), Buffer capacity for accommodating machine downtime in serial production lines, *International Journal of Production Research*, Vol. 40, No. 3, pp. 601–624.
12. Erel, E. (1993), Effect of discrete batch WIP transfer on the efficiency of production lines, *International Journal of Production Research*, Vol. 36, No. 2, pp. 343–358.

13. Faigle, U. and Kern, W. (1992), Some convergence results for probabilistic Tabu search, *ORSA Journal on Computing*, Vol. 4, pp. 32–37.
14. Fox, B.L. (1993), Integrating and accelerating tabu search, simulated annealing and genetic algorithms, *Annals of Operations Research*, Vol. 41, pp. 46–67.
15. Glover, F. (1986), Future paths for integer programming and links to artificial intelligence, *Computers and Operations Research*, Vol. 13, pp. 533–549.
16. Glover, F. (1989), Tabu search, Part I, *ORSA Journal on Computing*, Vol. 1, pp. 190–206.
17. Glover, F. (1990), Tabu search, Part II, *ORSA Journal on Computing*, Vol. 2, pp. 4–32.
18. Glover, F. (1992), Private communication.
19. Glover, F. and Laguna, M. (1998), *Tabu search*, Kluwer Academic Publishers.
20. Glover, F., Taillard, E., Laguna, M., and de Werra, D. (1993), Tabu search, *Annals of Operations Research*, Vol. 41, pp. 3–28.
21. Grefenstette, J.J. (1986), Optimization of control parameters for genetic algorithms, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 16, No. 1, pp. 122–128.
22. Gershwin, S.B. and Schor, J.E. (2000), Efficient algorithms for buffer space allocation, *Annals of Operations Research*, Vol. 93, pp. 117–144.
23. Grefenstette, J. (1986), Optimization of control parameters for genetic algorithms, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 16, Issue 1, pp. 122–128.
24. Hansen, P. (1986), The steepest ascent mildest descent heuristic for combinatorial programming, Presented at the Congress on *Numerical Methods in Combinatorial Optimization*, Capri, Italy.
25. Harris, J.H. and Powell, S.G. (1999), An algorithm for optimal buffer placement in reliable serial lines, *IIE Transactions*, Vol. 31, pp. 287–302.
26. Hatcher, J.M. (1969), The effect of internal storage on the production rate of a series of stages having exponential service times, *AIIE Transactions*, Vol. 1, No. 2, pp. 150–156.
27. Heavey, C., Papadopoulos, H.T., and Browne, J. (1993), The throughput rate of multistation unreliable production lines, *European Journal of Operational Research*, Vol. 68, No. 1, pp. 69–89.
28. Hertz, A., Taillard, E., and de Werra, D. (1995), A tutorial on Tabu search, *In Proc. of Giornate di Lavoro AIRO '95, (Enterprise Systems: Management of Technological and Organizational Changes)*, pp. 13–24.
29. Hillier, F.S. and So, K.C. (1991a), The effect of machine breakdowns and interstage storage on the performance of production line systems, *International Journal of Production Research*, Vol. 29, No. 10, pp. 2043–2055.
30. Hillier, F.S. and So, K.C. (1991b), The effect of the coefficient of variation of operation times on the allocation of storage space in production line systems, *IIE Transactions*, Vol. 23, No. 2, pp. 198–206.
31. Hillier, F.S. and So, K.C. (1995), On the optimal design of tandem queueing systems with finite buffers, *Queueing Systems*, Vol. 21, pp. 245–266.
32. Hillier, F.S., So, K.C., and Boling, R.W. (1993), Notes: Towards characterizing the optimal allocation of storage space in production line systems with variable processing times, *Management Science*, Vol. 39, No. 1, pp. 126–133.
33. Ho, Y.C., Eyler, M.A., and Chien, T.T. (1979), A gradient technique for general buffer storage design in a production line, *International Journal of Production Research*, Vol. 17, No. 6, pp. 557–580.
34. Jafari, M.A. and Shanthikumar, J.G. (1989), Determination of optimal Buffer storage capacities and optimal allocation in multistage automatic transfer lines, *IIE Transactions*, Vol. 21, No. 2, pp. 130–135.

35. Jensen, P.A., Pakath, R., and Wilson, J.R. (1991), Optimal buffer inventories for multistage production systems with failures, *European Journal of Operational Research*, Vol. 51, pp. 313–326.
36. Knuth, D.E. (1981), *The Art of Computer Programming*, Vol. 2/Seminumerical Algorithms, Second Edition, pp. 171–173, Addison-Wesley.
37. Koulamas, C.P. (1989), Optimal buffer space in two-stage machining systems with Markovian or non-Markovian tool life processes, *International Journal of Production Research*, Vol. 27, No. 7, pp. 1167–1178.
38. Kraemer, S.A. and Love, R.F. (1970), A model for optimizing the buffer inventory storage size in a sequential production system, *AIIE Transactions*, Vol. 2, No. 1, pp. 64–69.
39. Kubat, P. and Sumita, U. (1985), Buffers and backup machines in automatic transfer lines, *International Journal of Production Research*, Vol. 23, pp. 1259–1270.
40. Levantesi, R., Matta, A., and Tolio, T. (2001), A new algorithm for buffer allocation in production lines, *Proceedings of the Third Aegean International Conference on Design and Analysis of Manufacturing Systems*, May 19–22, 2001, Tinos Island, Greece, pp. 279–288.
41. Liu, C. and Tu, F.-S. (1994), Buffer allocation via genetic algorithm, *Proceedings of the 33rd Conference on Decision and Control*, Lake Buena Vista, FL, December, 1994, pp. 609–610.
42. Masso, J. and Smith, M.L. (1974), Interstage storages for three stage lines subjet to stochastic failures, *AIIE Transactions*, Vol. 6, No. 4, pp. 354–358.
43. Martin, G.E. (1994), Optimal design of production lines, *International Journal of Production Research*, Vol. 32, No. 5, pp. 989–1000.
44. Meester, L.E. and Shanthikumar, J.G. (1990), Concavity of the throughput of tandem queueing systems with finite buffer storage space, *Advances in Applied Probability*, Vol. 22, pp. 764–767.
45. Okamura, K. and Yamashita, H. (1977), Analysis of the effect of buffer storage capacity in transfer line systems, *AIIE Transactions*, Vol. 9, No. 2, pp. 127–135.
46. Papadopoulos, H.T. and Karagiannis, T.I. (2001), A genetic algorithm approach for the buffer allocation problem in unreliable production lines, *International Journal of Operations and Quantitative Management*, Vol. 7, No. 1, pp. 23–35.
47. Papadopoulos, H.T. and Vidalis, M.I. (1998), Optimal buffer storage allocation in balanced reliable production lines, *International Transactions in Operational Research*, Vol. 5, No. 4, pp. 325–339.
48. Papadopoulos, H.T. and Vidalis, M.I. (1999), Optimal buffer allocation in short μ-balanced unreliable production lines, *Computers & Industrial Engineering*, Vol. 37, pp. 691–710.
49. Papadopoulos, H.T. and Vidalis, M.I. (2001a), A heuristic algorithm for the buffer allocation in unreliable unbalanced production lines, *Computers & Industrial Engineering*, Vol. 41, pp. 261–277.
50. Papadopoulos, H.T. and Vidalis, M.I. (2001b), Minimizing WIP inventory in reliable production lines, *International Journal of Production Economics*, Vol. 70, pp. 185–197.
51. Park, T. (1993), A two-phase heuristic algorithm for determining buffer sizes of production lines, *International Journal of Production Research*, Vol. 31, No. 3, pp. 613–631.
52. Powell, S.G. (1994), Buffer allocation in unbalanced three-station serial lines, *International Journal of Production Research*, Vol. 32, No. 9, pp. 2201–2217.
53. Powell, S.G. and Pyke, D.F. (1994), Optimal allocation of buffers in serial production lines with a single bottleneck, The Amos Tuck School of Business Administration, Dartmouth College, Working Paper No. 301.

54. Sevast'yanov B.A. (1962), Influence of storage bin capacity on the average standstill time of a production line, *Theory of Probability and its Applications*, Vol. 7, pp. 429–438.
55. Singh, A. and Smith, MacGregor, J. (1997), Buffer allocation for an integer nonlinear network design problem, *Computers and Operations Research*, Vol. 24, No. 5, pp. 453–472.
56. Smith, MacGregor, J., and Chikhale, N. (1995), Buffer allocation for a class of nonlinear stochastic knapsack problems, *Annals of Operations Research*, Vol. 58, pp. 323–360.
57. Smith, MacGregor, J., and Daskalaki, S. (1988), Buffer space allocation in automated assembly lines, *Operations Research*, Vol. 36, No. 2, pp. 343–358.
58. So, K.C. (1990), The impact of buffering strategies on the perfomance of production line systems, *International Journal of Production Research*, Vol. 28, No. 2, pp. 2293–2307.
59. So, K.C. (1997), Optimal buffer allocation strategy for minimizing work-in-process inventory in unpaced production lines, *IIE Transactions*, Vol. 29, No. 1, pp. 81–88.
60. Spinellis, D.D. and Papadopoulos, C.T. (2000a), A simulated annealing approach for buffer allocation in reliable production lines, *Annals of Operations Research*, Vol. 93, pp. 373–384.
61. Spinellis, D.D. and Papadopoulos, C.T. (2000b), Stochastic algorithms for buffer allocation in reliable production lines, *Mathematical Problems in Engineering*, Vol. 5, pp. 441–458.
62. Vouros, G.A. and Papadopoulos, H.T. (1998), Buffer allocation in unreliable production lines using a knowledge based system, *Computers & Operations Research*, Vol. 25, No. 12, pp. 1055–1067.
63. Yamashita, H. and Altiok, T. (1998), Buffer capacity allocation for a desired throughput in production lines, *IIE Transactions*, Vol. 30, pp. 883–891.

6

Double and Triple Optimization

There are three pure allocation problems, viz., the work-load allocation problem, the server allocation problem and the buffer allocation problem, all concerned with maximizing throughput. Mathematically, these problems may be described as follows:

The work-load allocation problem, WAP:

$$\max X(\mathbf{w}) = \max X(w_1, w_2, \ldots, w_K)$$

subject to:

$$\sum_{i=1}^{K} w_i = 1, \quad \text{for } w_i > 0$$

for normalized total work-load equal to unity and fixed allocation of servers and fixed buffer allocation.

The server allocation problem, SAP:

$$\max X(\mathbf{s}) = \max X(S_1, S_2, \ldots, S_K))$$

subject to:

$$\sum_{i=1}^{K} S_i = S, \quad \text{for } S_i \geq 1 \text{ and integer}$$

for fixed allocation of work to each station and fixed buffer allocation.

The buffer allocation problem, BAP:

$$\max X(\mathbf{n}) = X(N_2, \ldots, N_K)$$

subject to:

$$\sum_{i=2}^{K} N_i = N, \quad \text{for } N_i \geq 0 \text{ and integer}$$

for fixed allocation of work to each station and fixed allocation of servers.

C. T. Papadopoulos et al., *Analysis and Design of Discrete Part Production Lines*, Springer Optimization and Its Applications,
DOI: 10.1007/978-0-387-89494-2_6, © Springer Science+Business Media, LLC 2009

As indicated above, there are three single-variable decision problems. Combining these problems into two-variable problems leads to the following three problems which may be mathematically described as follows:

The combined work-load allocation and server allocation problems, W + S:

$$\max X(\mathbf{w}, \mathbf{s})$$

subject to:

$$\sum_{i=1}^{K} w_i = 1, \quad \text{for } w_i > 0 \text{ and normalized work-load}$$

and

$$\sum_{i=1}^{K} S_i = S, \quad \text{for } S_i \geq 1 \text{ and integer}$$

and for fixed buffer allocation.

The reader may note that this problem has already been discussed in Chapter 4.

The combined work-load allocation and buffer allocation) problems, W + B:

$$\max X(\mathbf{w}, \mathbf{n})$$

subject to:

$$\sum_{i=1}^{K} w_i = 1, \quad \text{for } w_i > 0 \text{ and normalized work-load}$$

and

$$\sum_{i=2}^{K} N_i = N, \quad \text{for } N_i \geq 0 \text{ and integer}$$

and for fixed server allocation.

The combined server allocation and buffer allocation problems, S + B:

$$\max X(\mathbf{s}, \mathbf{n})$$

subject to:

$$\sum_{i=1}^{K} S_i = S, \quad \text{for } S_i \geq 1 \text{ and integer}$$

and

$$\sum_{i=2}^{K} N_i = N, \quad \text{for } N_i \geq 0 \text{ and integer}$$

and for fixed work-load allocation.

If all three decision variables are considered together, the following combined problem, W + S + B, may be described mathematically as:

$$\max X(\mathbf{w}, \mathbf{s}, \mathbf{n})$$

subject to:

$$\sum_{i=1}^{K} w_i = 1, \quad \text{for } w_i > 0 \text{ and normalized work-load}$$

and

$$\sum_{i=1}^{K} S_i = S, \quad \text{for } S_i \geq 1 \text{ and integer}$$

and

$$\sum_{i=2}^{K} N_i = N, \quad \text{for } N_i \geq 0 \text{ and integer.}$$

In practice, it is likely that the design of production line systems would involve decisions on at least two if not on all three of the decision variables, i.e., work-load, servers and buffers.

A further consideration in the above problems would be the service time distribution, given the number of servers. Usually, the service distribution variability is captured through the concepts of the mean of the service time distribution and the coefficient of variation of the distribution itself with the usual assumption of phase-type distributions. Even given the restriction of phase-type distribution (which is used for ease of computation), it must be realized that in a set of K work-stations there could be a difference in the coefficient of variation (cv) of the work-stations.

In Section 6.1, the W + B problem is treated and some design guide rules are given. Likewise, in Section 6.2 the S + B problem is discussed and some design guidelines are presented. Finally, in Section 6.3 the W + S + B problem is discussed and a range of results are presented.

6.1 Simultaneous Allocation of Work and Buffers, W + B

In the $W + B$ problem the number of servers at each station is fixed.

Buzacott and Shanthikumar (1993) showed analytically that if the buffer slots are an integer multiple of the $K - 1$ intermediate buffers, a balanced buffer allocation is associated with maximum throughput. Hillier and So (1995) obtained empirical results for the optimal buffer and work allocations for maximum throughput for $K = 3, 4$ and 5 stations and $N = 0, 1, \ldots, 8$ total buffer slots. The reader will note that in this paper, Hillier and So considered saturated lines, i.e., the first station is never starved and results are only given for lines with perfectly reliable machines. Buzacott and Shanthikumar's results were confirmed and generally a bowl phenomenon of work allocation was associated with maximum throughput although in two cases a type of double bowl phenomenon of work allocation was indicated (in the cases of $K = 5, N = 2$ and $K = 5, N = 6$). Symmetrical bowl allocations of work-load are associated with strictly uniform allocation of buffers. With respect to buffer allocation the optimal pattern is as follows:

Step 1: Allocate the same maximum number of buffer slots to each buffer of the line.
Step 2: Allocate the extra buffer slots over the above uniform buffer allocation to the interior buffers, particularly to the central buffers rather than the end buffers.

Hillier and So (1995) results for the $W+B$ problem do not extend to large buffer slots and the published results assumed that there is only one server at each station although the authors stated that their conclusions apply for multi-server workstations. Table 7 of Hillier and So (1995) gives a timely warning to researchers and designers alike by showing the relatively small increase in throughput achieved by unbalancing the lines and using the optimal allocation of buffers slots as derived from the $W+B$ optimization.

Over the range of experiments undertaken by Hillier and So (1995) it may be concluded that the work allocation and the corresponding buffer allocation solutions for the W + B optimization problem produce the following general rules:

- Larger work-loads are associated with stations with larger buffer capacities.
- Higher work-loads are assigned to stations at both ends of the line.

It is clear from this work that as uniform as possible buffer allocation will generally lead to better throughput independently of the actual work-load allocation in production lines, where only the server allocation is faced.

Suggested solution procedures for the W + B problem using the algorithms available at the website associated with this text will be discussed at the end of Section 6.3.

6.2 Simultaneous Allocation of Servers and Buffers, S + B

In the $S+B$ problem the work-load allocation is given.

Hillier and So (1995) studied the $S+B$ problem under uniform work-load allocation for $K = 3,4$ and 5 stations and total number of servers, S up to twice the number of stations and buffer spaces, $N = 1,\ldots,4$. Their results indicate that the optimal server allocation is given by the as uniform as possible allocation and the extra servers assigned to the interior stations with the two end stations having the lowest priority in receiving the extra servers with the last station receiving the extra server before the first station.

More specifically, in the cases published by Hillier and So (1995) the following pattern of the optimal server allocation has been observed:

1. If $S = aK$, $a > 1$ and integer, a uniform server allocation is optimal.
2. If $S = aK - 1$, $a > 1$ and integer, a server allocation in which the first station has $a-1$ servers and all the others have a servers is optimal and as would be expected the mirror image of this server allocation where the deficit in the number of servers is at the last station results in a near optimal solution.
3. If $S = aK - 2$, $a > 1$ and integer, the optimal server allocation is to allocate $a-1$ servers to both end stations and a servers to the other $K-2$ stations.

Beyond the above three rather definite patterns it is difficult to develop a server allocation rule from the Hillier and So's published results for other levels of total number of servers.

It appears from Hillier and So's results for the $S+B$ problem that the optimal buffer allocation does not follow any particular pattern. It is well known that an extra server at a station in effect provides an extra buffer slot to that station. The general guideline therefore is to provide extra buffer slots to stations which have received fewer servers as a result of the optimal allocation of servers as outlined above. For example, in the cases where $S = 2K-2$ and $K = 3,4$ and 5, the buffer slots are assigned to the two end stations which are deficient by one server in comparison to all the other stations using the optimal server allocation procedure. It would be fair to say that arising from Hillier and So's results, a designer would be well advised to allocate the servers first according to the heuristic indicated above and then to allocate the buffer slots to near stations that have received fewer servers. Clearly, this guideline must be understood in the context of the relatively small number of stations, servers and buffer slots involved in Hillier and So's pioneering studies (1995).

Suggested solution procedures for the S + B problem using the algorithms available at the website associated with this text will be discussed at the end of Section 6.3.

6.3 Simultaneous Allocation of Work, Servers and Buffers, W + S + B

In this problem the number of stations, K, the total number of servers, S, and the total number of buffer slots, N, are given. Because of the computational complexity involved, Hillier and So (1995) studied systems for $K = 3,4,5$ stations, $S = 4,\ldots,8$ servers and total buffer slots, N up to 4 maximum.

The most persistent result is the existence of the L-phenomenon for the server allocation, i.e., to allocate all extra servers to the end stations. For the buffer allocation the tendency is to allocate the buffer slots as uniformly as possible with the extra buffers being assigned toward the last station of the line. Thus, it appears that results from the $W+S$ problem and the $W+B$ problem somewhat dominate the results from the $S+B$ problem. The optimal work-load allocation in all cases follows a bowl pattern with a significantly increased work-load assigned to the last station. Indeed, the optimal work-load per server has a much higher value for the last station than do any of the other stations and the first station has the next highest work-load per server. In the cases studied by Hillier and So (1995), the work-load allocation to the stations other than the two end stations tended to be almost uniform resulting in what might be described as a three-level bowl of work-load allocation.

Hillier and So (1995) proposed the following heuristic allocation procedure:

Step 1: Determine the server allocation by following the L-phenomenon procedure already discussed in Chapter 4, taking into account any adjustments required for upper and lower bounds on the number of servers per station.

Step 2: Determine the buffer allocation as follows:
- Step 2.1: If the server allocation is balanced, select as uniform as possible buffer allocation with buffers closest to the center having a higher priority to the buffers away from the center, in accordance with the results obtained from the $W + S$ problem. Any tie should be resolved in favor of buffer slots toward the end of the line.
- Step 2.2: If the server allocation is unbalanced, select as uniform as possible buffer allocation with extra slots assigned to buffers that are closest to stations with the most servers.

Step 3: Determine the work-load allocation once the server and buffer allocations are assigned.

In Spinellis, Papadopoulos and MacGregor Smith (2000), simulated annealing was used in conjunction with the expansion algorithm to obtain results for both short and long lines with perfectly reliable stations consisting of parallel machines at each station. It should be noted that the application here is with respect to *quasi-saturated* lines in contrast to that treated by Hillier and So (1995) and described above where *saturated* lines were considered. In quasi-saturated lines the number of buffer slots in front of the first station in the Spinellis et al. (2000) application may not have ensured that the first station is always occupied which was the case in the Hillier and So (1995) application. The analytical approach also differs in that Spinellis et al. (2000) used the novel search method of simulated annealing and the approximate expansion algorithm for both short and long lines with single-machine and multiple-machine stations, whereas Hillier and So (1995) used complete enumeration and an exact Markovian method for single-machine and multiple-machine station short lines. Additionally, in the Spinellis et al. (2000) model, and within the context of the expansion method, the contents of the buffer in front of the first station are taken into consideration when allocating the total number of buffer slots, which is not the case in the model of Hillier and So (1995). Despite these differences, it is illuminating to compare the results of Spinellis et al. (2000) with those of Hillier and So (1995) in relation to short lines. Of course, as the topologies of the systems are slightly different one could expect different results. However, the main point must surely be the fact that Hillier and So (1995) used exact methods (Markovian and complete enumeration), whereas Spinellis et al. (2000) used the approximate expansion algorithm and simulated annealing.

Comparing the results obtained for short lines in both papers, the following observations may be made:

1. The work-load allocation in Spinellis et al. (2000) does not follow the bowl phenomenon as in Hillier and So (1995), but in general continues to diminish toward the end of the line.
2. The buffer allocation in Spinellis et al. (2000) does not follow the inverse bowl phenomenon as in Hillier and So (1995), but increases monotonically across the line toward the end stations. As the number of total buffer slots increases in Spinellis et al. (2000), a certain fixed number of slots are allocated in front of the first station and the remaining slots are allocated almost uniformly among

the interstation buffers with some preference given to the downstream buffer locations.

3. The server allocation in Spinellis et al. (2000) for a small number of servers follows a pattern similar to the one presented in Hillier and So (1995). However, as the number of servers increases in Spinellis et al. (2000), servers tend to accumulate toward the beginning of the line.
4. The server and work-load allocation in Spinellis et al. (2000) do not exhibit the *L*-phenomenon shown in Hillier and So (1995).
5. The buffer and work-load allocation results are different in the two papers. As far as the buffer allocation is concerned, in Spinellis et al. (2000) buffers tend to accumulate toward the end of the line, whereas in Hillier and So (1995), the buffer allocations are more uniform. The allocation of work, however, does not exhibit the symmetrical bowl phenomenon presented in Hillier and So (1995) and follows the usual descending rate across the line.
6. The buffer and server allocation results of Spinellis et al. (2000) are roughly similar to those presented in Hillier and So (1995) in both the server and the buffer allocation vectors. In both models servers are rather uniformly allocated, but in Spinellis et al. (2000), servers tend to accumulate toward the beginning and middle of the line, whereas in Hillier and So (1995) they are allocated toward the middle and end of the line. In both models, buffers tend to accumulate toward the line ends.
7. Finally, the buffer, server, and work-load allocation roughly follows the shape of work-load allocation presented in Hillier and So (1995), but the allocation of buffers and servers is quite dissimilar.

In the Spinellis et al. (2000) paper, the expansion method was always used in developing the results presented. Recently, the authors decided to investigate saturated lines using the expansion method (EXPA), the decomposition method (DECO) and the Markovian method (MARK) as evaluative tools. The expansion method and the Markovian method can accommodate parallel-machine stations, whereas the decomposition method is usually used in single-machine station lines. The following experiments, given in Table 6.1, were run using the above three evaluative methods in conjunction with complete enumeration (CE) and simulated annealing (SA) as the search methods.

Below, in Tables 6.2–6.3, are the numerical results of these experiments in all of which the buffer in front of the first station was assigned 5 slots, i.e., $B_1 = 5$. For long lines it is unlikely that $B_1 = 5$ will lead to saturated lines. The Markovian model used by the authors is based on the algorithm given in Heavey, Papadopoulos and Browne (1993), the decomposition algorithm used is based on one of the algorithms given in Dallery and Frein (1993) and the expansion algorithm is based on the method given in Kerbache and MacGregor Smith (1987) and Jain and MacGregor Smith (1994). All of these methods are described in Chapter 2.

Table 6.1. Overall plan of experiments

Evaluative Method	Search Method	No. of Stations K	Total # of Buffer Slots N
MARK	CE	4,6,8	1–26
DECO	CE	4,6,8	1–26
EXPA	CE	4,6,8	1–26
DECO	SA	5,7,9	9–30
EXPA	SA	5,7,9	9–30
DECO	CE	5,7,9	9–30
EXPA	CE	5,7,9	9–30
DECO	SA	10	11–21
EXPA	SA	10	11–21
DECO	CE	10	11–18
EXPA	CE	10	11–18
DECO	SA	16	16–45
EXPA	SA	16	16–45
DECO	SA	5–9, 11(10)61	8–16, 20–120
EXPA	SA	5–9, 11(10)61	8–16, 20–120
DECO	CE	5–6	8–10
EXPA	CE	5–6	8–10

Table 6.2. Throughput and buffer allocation for 4-, 6- and 8-station lines via CE

(Evaluative-Search) (Method-Method)	No. of Stations K	No. of Buffer Slots N	Throughput X_K	Buffer Allocation $(B_1, B_2, \cdots, B_K)$
(MARK-CE)	8	1	0.464806	(5,0,0,0,1,0,0,0)
(MARK-CE)	8	2	0.485547	(5,0,0,1,0,1,0,0)
(MARK-CE)	8	3	0.503378	(5,0,1,0,1,0,1,0)
(MARK-CE)	8	4	0.521864	(5,0,1,1,0,1,1,0)
(MARK-CE)	8	5	0.542124	(5,0,1,1,1,1,1,0)
(MARK-CE)	8	6	0.554479	(5,1,1,1,1,1,1,0)
(MARK-CE)	8	7	0.572014	(5,1,1,1,1,1,1,1)
(MARK-CE)	6	6	0.60923	(5,1,1,2,1,1)
(MARK-CE)	6	7	0.624668	(5,1,2,1,2,1)
(MARK-CE)	4	21	0.844477	(5,7,7,7)
(MARK-CE)	4	24	0.858043	(5,8,8,8)
(MARK-CE)	4	25	0.862382	(5,8,9,8)
(MARK-CE)	4	26	0.866002	(5,8,10,8)
(DECO-CE)	8	1	0.428197	(5,0,0,0,0,0,1,0)
(DECO-CE)	8	2	0.45005	(5,0,0,0,1,0,1,0)
(DECO-CE)	8	3	0.472079	(5,0,0,1,0,1,1,0)
(DECO-CE)	8	4	0.493967	(5,0,1,0,1,1,1,0)
(DECO-CE)	8	5	0.515725	(5,0,1,1,1,1,1,0)
(DECO-CE)	8	6	0.535165	(5,0,1,1,1,1,1,1)
(DECO-CE)	8	7	0.550428	(5,1,1,1,1,1,1,1)

(*continued*)

Table 6.2. (Continued)

(Evaluative-Search) (Method-Method)	No. of Stations K	No. of Buffer Slots N	Throughput X_K	Buffer Allocation $(B_1, B_2, \cdots, B_K)$
(DECO-CE)	6	6	0.587922	(5,1,1,2,1,1)
(DECO-CE)	6	7	0.606644	(5,1,1,2,2,1)
(DECO-CE)	4	21	0.842941	(5,7,8,6)
(DECO-CE)	4	24	0.857248	(5,8,9,7)
(DECO-CE)	4	25	0.861613	(5,8,9,8)
(DECO-CE)	4	26	0.865664	(5,8,10,8)
(EXPA-CE)	8	1	0.204673	(5,0,0,0,0,0,0,1)
(EXPA-CE)	8	2	0.224199	(5,0,0,0,0,0,1,1)
(EXPA-CE)	8	3	0.247024	(5,0,0,0,0,1,1,1)
(EXPA-CE)	8	4	0.273771	(5,0,0,0,1,1,1,1)
(EXPA-CE)	8	5	0.306172	(5,0,1,0,1,1,1,1)
(EXPA-CE)	8	6	0.343134	(5,0,1,1,1,1,1,1)
(EXPA-CE)	8	7	0.379798	(5,1,1,1,1,1,1,1)
(EXPA-CE)	6	6	0.456404	(5,1,1,1,1,2)
(EXPA-CE)	6	7	0.480015	(5,1,1,1,2,2)
(EXPA-CE)	4	21	0.830446	(5,7,7,7)
(EXPA-CE)	4	24	0.846169	(5,8,8,8)
(EXPA-CE)	4	25	0.850921	(5,8,8,9)
(EXPA-CE)	4	26	0.855233	(5,8,9,9)

Table 6.3. Throughput and buffer allocation for 5- and 6-station lines via CE

(Evaluative-Search) (Method-Method)	No. of Stations K	No. of Buffer Slots N	Throughput X_K	Buffer Allocation $(B_1, B_2, \cdots, B_K)$
(DECO-CE)	5	8	0.665763	(5,2,2,2,2)
(DECO-CE)	6	10	0.653177	(5,2,2,2,2,2)
(EXPA-CE)	5	8	0.590449	(5,2,2,2,2)
(EXPA-CE)	6	10	0.555666	(5,2,2,2,2,2)

Table 6.4. Throughput and buffer allocation for 5-, 7- and 9-station lines via SA

(Evaluative-Search) (Method-Method)	No. of Stations K	No. of Buffer Slots N	Throughput X_K	Buffer Allocation $(B_1, B_2, \cdots, B_K)$
(DECO-SA)	9	9	0.556706	(5,1,1,1,1,1,2,1,1)
(DECO-SA)	9	10	0.56886	(5,1,1,1,2,1,2,1,1)
(DECO-SA)	9	11	0.581225	(5,1,1,2,1,2,1,2,1)
(DECO-SA)	9	12	0.593139	(5,1,1,2,1,2,2,2,1)
(DECO-SA)	9	13	0.604882	(5,1,1,2,2,2,2,2,1)
(DECO-SA)	9	14	0.617087	(5,1,2,2,2,2,2,2,1)
(DECO-SA)	9	15	0.626453	(5,1,2,2,2,2,2,2,2)
(DECO-SA)	7	12	0.644706	(5,2,2,2,2,2,2)
(DECO-SA)	7	13	0.655936	(5,2,2,2,3,2,2)

(*continued*)

Table 6.4. (Continued)

(Evaluative-Search) (Method-Method)	No. of Stations K	No. of Buffer Slots N	Throughput X_K	Buffer Allocation $(B_1, B_2, \cdots, B_K)$
(DECO-SA)	5	25	0.819322	(5,6,7,7,5)
(DECO-SA)	5	28	0.833229	(5,6,8,8,6)
(DECO-SA)	5	29	0.836937	(5,7,8,8,6)
(DECO-SA)	5	30	0.840874	(5,7,8,8,7)
(EXPA-SA)	9	9	0.372453	(5,1,1,1,1,1,1,1,2)
(EXPA-SA)	9	10	0.386788	(5,1,1,1,1,1,1,2,2)
(EXPA-SA)	9	11	0.402043	(5,1,1,1,1,1,2,2,2)
(EXPA-SA)	9	12	0.418135	(5,1,1,1,1,2,2,2,2)
(EXPA-SA)	9	13	0.434875	(5,1,1,1,2,2,2,2,2)
(EXPA-SA)	9	14	0.451915	(5,1,1,2,2,2,2,2,2)
(EXPA-SA)	9	15	0.468776	(5,1,2,2,2,2,2,2,2)
(EXPA-SA)	7	12	0.527447	(5,2,2,2,2,2,2)
(EXPA-SA)	7	13	0.541095	(5,2,2,2,2,2,3)
(EXPA-SA)	5	25	0.789839	(5,6,6,6,7)
(EXPA-SA)	5	28	0.806145	(5,6,7,7,8)
(EXPA-SA)	5	29	0.811007	(5,7,7,7,8)
(EXPA-SA)	5	30	0.815795	(5,7,7,8,8)

Table 6.5. Throughput and buffer allocation for 5-, 7- and 9-station lines via CE

(Evaluative-Search) (Method-Method)	No. of Stations K	No. of Buffer Slots N	Throughput X_K	Buffer Allocation $(B_1, B_2, \cdots, B_K)$
(DECO-CE)	9	9	0.556706	(5,1,1,1,1,1,2,1,1)
(DECO-CE)	9	10	0.56886	(5,1,1,1,2,1,2,1,1)
(DECO-CE)	9	11	0.581225	(5,1,1,2,1,2,1,2,1)
(DECO-CE)	9	12	0.593139	(5,1,1,2,1,2,2,2,1)
(DECO-CE)	9	13	0.604882	(5,1,1,2,2,2,2,2,1)
(DECO-CE)	9	14	0.617087	(5,1,2,2,2,2,2,2,1)
(DECO-CE)	9	15	0.626453	(5,1,2,2,2,2,2,2,2)
(DECO-CE)	7	12	0.644706	(5,2,2,2,2,2,2)
(DECO-CE)	7	13	0.655936	(5,2,2,2,3,2,2)
(DECO-CE)	5	25	0.819322	(5,6,7,7,5)
(DECO-CE)	5	28	0.833229	(5,6,8,8,6)
(DECO-CE)	5	29	0.836937	(5,7,8,8,6)
(DECO-CE)	5	30	0.840874	(5,7,8,8,7)
(EXPA-CE)	9	9	0.372453	(5,1,1,1,1,1,1,1,2)
(EXPA-CE)	9	10	0.386788	(5,1,1,1,1,1,1,2,2)
(EXPA-CE)	9	11	0.402043	(5,1,1,1,1,1,2,2,2)
(EXPA-CE)	9	12	0.418135	(5,1,1,1,1,2,2,2,2)
(EXPA-CE)	9	13	0.434875	(5,1,1,1,2,2,2,2,2)
(EXPA-CE)	9	14	0.451915	(5,1,1,2,2,2,2,2,2)
(EXPA-CE)	9	15	0.468776	(5,1,2,2,2,2,2,2,2)

(*continued*)

Table 6.5. (Continued)

(Evaluative-Search) (Method-Method)	No. of Stations K	No. of Buffer Slots N	Throughput X_K	Buffer Allocation $(B_1, B_2, \cdots, B_K)$
(EXPA-CE)	7	12	0.527447	(5,2,2,2,2,2,2)
(EXPA-CE)	7	13	0.541095	(5,2,2,2,2,2,3)
(EXPA-CE)	5	25	0.789839	(5,6,6,6,7)
(EXPA-CE)	5	28	0.806145	(5,6,7,7,8)
(EXPA-CE)	5	29	0.811007	(5,7,7,7,8)
(EXPA-CE)	5	30	0.815795	(5,7,7,8,8)

Table 6.6. Throughput and buffer allocation for 10-station lines via SA

(Evaluative-Search) (Method-Method)	No. of Stations K	No. of Buffer Slots N	Throughput X_K	Buffer Allocation $(B_1, B_2, \cdots, B_K)$
(DECO-SA)	10	11	0.561615	(5,1,1,1,1,1,2,1,2,1)
(DECO-SA)	10	12	0.572518	(5,1,1,2,1,2,1,2,1,1)
(DECO-SA)	10	13	0.583461	(5,1,1,2,1,2,1,2,2,1)
(DECO-SA)	10	14	0.593987	(5,1,1,2,1,2,2,2,2,1)
(DECO-SA)	10	15	0.604453	(5,1,1,2,2,2,2,2,2,1)
(DECO-SA)	10	16	0.615375	(5,1,2,2,2,2,2,2,2,1)
(DECO-SA)	10	17	0.623801	(5,1,2,2,2,2,2,2,2,2)
(DECO-SA)	10	18	0.630954	(5,1,2,2,2,2,2,3,2,2)
(DECO-SA)	10	19	0.637935	(5,1,2,2,2,3,2,3,2,2)
(DECO-SA)	10	20	0.645319	(5,2,2,2,3,2,2,3,2,2)
(DECO-SA)	10	21	0.65282	(5,2,2,3,2,3,2,3,2,2)
(EXPA-SA)	10	11	0.365613	(5,1,1,1,1,1,1,1,2,2)
(EXPA-SA)	10	12	0.378946	(5,1,1,1,1,1,1,2,2,2)
(EXPA-SA)	10	13	0.393029	(5,1,1,1,1,1,2,2,2,2)
(EXPA-SA)	10	14	0.407759	(5,1,1,1,1,2,2,2,2,2)
(EXPA-SA)	10	15	0.422938	(5,1,1,1,2,2,2,2,2,2)
(EXPA-SA)	10	16	0.438231	(5,1,1,2,2,2,2,2,2,2)
(EXPA-SA)	10	17	0.453198	(5,1,2,2,2,2,2,2,2,2)
(EXPA-SA)	10	18	0.4664	(5,2,2,2,2,2,2,2,2,2)
(EXPA-SA)	10	19	0.475801	(5,2,2,2,2,2,2,2,2,3)
(EXPA-SA)	10	20	0.485532	(5,2,2,2,2,2,2,2,3,3)
(EXPA-SA)	10	21	0.495542	(5,2,2,2,2,2,2,3,3,3)

Table 6.7. Throughput and buffer allocation for 10-station lines via CE

(Evaluative-Search) (Method-Method)	No. of Stations K	No. of Buffer Slots N	Throughput X_K	Buffer Allocation $(B_1, B_2, \cdots, B_K)$
(DECO-CE)	10	11	0.561615	(5,1,1,1,1,1,2,1,2,1)
(DECO-CE)	10	12	0.572518	(5,1,1,2,1,2,1,2,1,1)
(DECO-CE)	10	13	0.583461	(5,1,1,2,1,2,1,2,2,1)
(DECO-CE)	10	14	0.593987	(5,1,1,2,1,2,2,2,2,1)

(*continued*)

Table 6.7. (Continued)

(Evaluative-Search) (Method-Method)	No. of Stations K	No. of Buffer Slots N	Throughput X_K	Buffer Allocation $(B_1, B_2, \cdots, B_K)$
(DECO-CE)	10	15	0.604453	(5,1,1,2,2,2,2,2,2,1)
(DECO-CE)	10	16	0.615375	(5,1,2,2,2,2,2,2,2,1)
(DECO-CE)	10	17	0.623801	(5,1,2,2,2,2,2,2,2,2)
(DECO-CE)	10	18	0.630954	(5,1,2,2,2,2,2,3,2,2)
(EXPA-CE)	10	11	0.365613	(5,1,1,1,1,1,1,1,2,2)
(EXPA-CE)	10	12	0.378946	(5,1,1,1,1,1,1,2,2,2)
(EXPA-CE)	10	13	0.393029	(5,1,1,1,1,1,2,2,2,2)
(EXPA-CE)	10	14	0.407759	(5,1,1,1,1,2,2,2,2,2)
(EXPA-CE)	10	15	0.422938	(5,1,1,1,2,2,2,2,2,2)
(EXPA-CE)	10	16	0.438231	(5,1,1,2,2,2,2,2,2,2)
(EXPA-CE)	10	17	0.453198	(5,1,2,2,2,2,2,2,2,2)
(EXPA-CE)	10	18	0.4664	(5,2,2,2,2,2,2,2,2,2)

Table 6.8. Throughput and buffer allocation for 16-station lines via SA

(Evaluative-Search) (Method-Method)	No. of Stations K	No. of Buffer Slots N	Throughput X_K	Buffer Allocation $(B_1, B_2, \cdots, B_K)$
(DECO-SA)	16	16	0.531584	(5,1,1,1,1,1,1,1,1,1,1,1,1,2,1)
(DECO-SA)	16	17	0.538177	(5,1,1,1,1,1,1,1,1,1,1,2,1,2,1)
(DECO-SA)	16	18	0.544701	(5,1,1,1,1,1,1,1,1,1,2,1,2,2,1)
(DECO-SA)	16	19	0.551151	(5,1,1,1,1,1,1,1,1,2,1,2,2,2,1)
(DECO-SA)	16	20	0.557554	(5,1,1,1,1,1,1,1,2,1,2,2,2,2,1)
(DECO-SA)	16	21	0.564139	(5,1,1,2,1,2,1,1,2,1,2,1,2,1,1)
(DECO-SA)	16	22	0.570869	(5,1,1,2,1,2,1,2,1,2,1,2,1,2,1)
(DECO-SA)	16	23	0.577393	(5,1,1,2,1,2,1,2,1,2,1,2,2,2,1)
(DECO-SA)	16	24	0.583771	(5,1,1,2,1,2,1,2,1,2,2,2,2,2,1)
(DECO-SA)	16	25	0.590093	(5,1,1,2,1,2,1,2,2,2,2,2,2,2,1)
(DECO-SA)	16	26	0.596421	(5,1,1,2,1,2,2,2,2,2,2,2,2,2,1)
(DECO-SA)	16	27	0.602862	(5,1,2,1,2,2,2,2,2,2,2,2,2,2,1)
(DECO-SA)	16	28	0.609608	(5,1,2,2,2,2,2,2,2,2,2,2,2,2,1)
(DECO-SA)	16	29	0.614872	(5,1,2,2,2,2,2,2,2,2,2,2,2,2,2)
(DECO-SA)	16	30	0.619337	(5,1,2,2,2,2,2,2,2,2,2,2,3,2,2)
(DECO-SA)	16	31	0.623678	(5,1,2,2,2,2,2,2,2,2,2,3,2,3,2)
(DECO-SA)	16	32	0.627994	(5,1,2,2,2,2,2,2,2,2,2,3,2,3,3)
(DECO-SA)	16	33	0.632321	(5,1,2,2,2,2,2,2,2,2,3,2,3,3,3)
(DECO-SA)	16	34	0.636609	(5,1,2,2,2,2,2,2,2,3,2,3,3,3,3)
(DECO-SA)	16	35	0.640958	(5,2,2,3,2,3,2,2,3,2,2,3,2,3,2)
(DECO-SA)	16	36	0.645451	(5,2,2,3,2,3,2,3,2,3,2,3,2,3,2)
(DECO-SA)	16	37	0.649859	(5,2,2,3,2,3,2,3,2,3,2,3,2,3,3)
(DECO-SA)	16	38	0.654271	(5,2,2,3,2,3,2,3,2,3,2,3,3,3,3)

(*continued*)

Table 6.8. (Continued)

(Evaluative-Search) (Method-Method)	No. of Stations K	No. of Buffer Slots N	Throughput X_K	Buffer Allocation $(B_1, B_2, \cdots, B_K)$
(DECO-SA)	16	39	0.658581	(5,2,2,3,2,3,2,3,2,3,3,3,3,3,3,2)
(DECO-SA)	16	40	0.662864	(5,2,2,3,2,3,3,2,3,3,3,3,3,3,2)
(DECO-SA)	16	41	0.6672	(5,2,2,3,3,2,3,3,3,3,3,3,3,3,2)
(DECO-SA)	16	42	0.671569	(5,2,2,3,3,3,3,3,3,3,3,3,3,3,2)
(DECO-SA)	16	43	0.676039	(5,2,3,3,3,3,3,3,3,3,3,3,3,3,2)
(DECO-SA)	16	44	0.679259	(5,2,3,3,3,3,3,3,3,3,3,3,3,3,3)
(DECO-SA)	16	45	0.682455	(5,2,3,3,3,3,3,3,3,3,3,3,4,3,3)
(EXPA-SA)	16	16	0.277368	(5,1,1,1,1,1,1,1,1,1,1,1,1,1,2)
(EXPA-SA)	16	17	0.284163	(5,1,1,1,1,1,1,1,1,1,1,1,1,2,2)
(EXPA-SA)	16	18	0.291307	(5,1,1,1,1,1,1,1,1,1,1,1,2,2,2)
(EXPA-SA)	16	19	0.298814	(5,1,1,1,1,1,1,1,1,1,1,2,2,2,2)
(EXPA-SA)	16	20	0.30669	(5,1,1,1,1,1,1,1,1,1,2,2,2,2,2)
(EXPA-SA)	16	21	0.314933	(5,1,1,1,1,1,1,1,1,2,2,2,2,2,2)
(EXPA-SA)	16	22	0.323534	(5,1,1,1,1,1,1,1,2,2,2,2,2,2,2)
(EXPA-SA)	16	23	0.332466	(5,1,1,1,1,1,1,1,2,2,2,2,2,2,2)
(EXPA-SA)	16	24	0.341681	(5,1,1,1,1,1,1,2,2,2,2,2,2,2,2)
(EXPA-SA)	16	25	0.351103	(5,1,1,1,1,1,2,2,2,2,2,2,2,2,2)
(EXPA-SA)	16	26	0.360616	(5,1,1,1,1,2,2,2,2,2,2,2,2,2,2)
(EXPA-SA)	16	27	0.370053	(5,1,1,1,2,2,2,2,2,2,2,2,2,2,2)
(EXPA-SA)	16	28	0.379188	(5,1,1,2,2,2,2,2,2,2,2,2,2,2,2)
(EXPA-SA)	16	29	0.38777	(5,1,2,2,2,2,2,2,2,2,2,2,2,2,2)
(EXPA-SA)	16	30	0.39505	(5,2,2,2,2,2,2,2,2,2,2,2,2,2,2)
(EXPA-SA)	16	31	0.400635	(5,2,2,2,2,2,2,2,2,2,2,2,2,2,3)
(EXPA-SA)	16	32	0.406398	(5,2,2,2,2,2,2,2,2,2,2,2,2,3,3)
(EXPA-SA)	16	33	0.412335	(5,2,2,2,2,2,2,2,2,2,2,2,3,3,3)
(EXPA-SA)	16	34	0.41844	(5,2,2,2,2,2,2,2,2,2,2,3,3,3,3)
(EXPA-SA)	16	35	0.424701	(5,2,2,2,2,2,2,2,3,2,2,3,3,3,3)
(EXPA-SA)	16	36	0.4311	(5,2,2,2,2,2,2,2,2,2,3,3,3,3,3)
(EXPA-SA)	16	37	0.437612	(5,2,2,2,2,2,2,2,2,3,3,3,3,3,3)
(EXPA-SA)	16	38	0.444201	(5,2,2,2,2,2,2,2,3,3,3,3,3,3,3)
(EXPA-SA)	16	39	0.450818	(5,2,2,2,2,2,2,3,3,3,3,3,3,3,3)
(EXPA-SA)	16	40	0.457398	(5,2,2,2,2,2,3,3,3,3,3,3,3,3,3)
(EXPA-SA)	16	41	0.463855	(5,2,2,2,2,3,3,3,3,3,3,3,3,3,3)
(EXPA-SA)	16	42	0.470077	(5,2,2,2,3,3,3,3,3,3,3,3,3,3,3)
(EXPA-SA)	16	43	0.475921	(5,2,2,3,3,3,3,3,3,3,3,3,3,3,3)
(EXPA-SA)	16	44	0.481208	(5,2,3,3,3,3,3,3,3,3,3,3,3,3,3)
(EXPA-SA)	16	45	0.485701	(5,3,3,3,3,3,3,3,3,3,3,3,3,3,3)

As may be seen from these tables, the results obtained by Hillier and So (1995) are confirmed in all cases considered, using the Markovian model of Heavey et al. (1993) and complete enumeration. This result confirms that the Markovian model used by the authors gives the same results as the Markovian model used by Hillier and So (1995). As may be noted from the tables, the same lines were analyzed

Table 6.9. Throughput and buffer allocation for 5(1)9-, 11(10)61-station lines via SA

(Evaluative-Search) (Method-Method)	No. of Stations K	No. of Buffer Slots N	Throughput X_K	Buffer Allocation $(B_1, B_2, \cdots, B_K)$
(DECO-SA)	5	8	0.665763	(5,2,2,2,2)
(DECO-SA)	6	10	0.653177	(5,2,2,2,2,2)
(DECO-SA)	7	12	0.644706	(5,2,2,2,2,2,2)
(DECO-SA)	8	14	0.63876	(5,1,2,2,2,3,2,2)
(DECO-SA)	9	16	0.634412	(5,1,2,2,2,2,3,2,2)
(DECO-SA)	11	20	0.628133	(5,1,2,2,2,2,2,2,3,2,2)
(DECO-SA)	21	40	0.614733	(5,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,2,2)
(DECO-SA)	31	60	0.60998	(5,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,2,2)
(DECO-SA)	41	80	0.607546	(5,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,2,2)
(DECO-SA)	51	100	0.606066	(5,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,2,2)
(DECO-SA)	61	120	0.605072	(5,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,2,2)
(EXPA-SA)	5	8	0.590449	(5,2,2,2,2)
(EXPA-SA)	6	10	0.555666	(5,2,2,2,2,2)
(EXPA-SA)	7	12	0.527447	(5,2,2,2,2,2,2)
(EXPA-SA)	8	14	0.503894	(5,2,2,2,2,2,2,2)
(EXPA-SA)	9	16	0.483812	(5,2,2,2,2,2,2,2,2)
(EXPA-SA)	11	20	0.451099	(5,2,2,2,2,2,2,2,2,2,2)
(EXPA-SA)	21	40	0.358377	(5,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2)
(EXPA-SA)	31	60	0.311407	(5,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2)
(EXPA-SA)	41	80	0.28145	(5,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2)
(EXPA-SA)	51	100	0.260082	(5,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2)
(EXPA-SA)	61	120	0.243786	(5,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2)

using the decomposition method and simulated annealing and separately complete enumeration. These lines were also analyzed using the expansion method and again with simulated annealing and separately complete enumeration. With respect to each line the results were independent of the search method used but were dependent on the evaluative method. The decomposition method in all cases gave a higher throughput than that obtained using the expansion method and in some cases significantly higher when the number of buffer slots were very small. However, both approximate evaluative methods gave lower throughput rates than that calculated on an exact basis using the Markovian method and either complete enumeration or simulated annealing in the cases presented. With respect to the buffer allocation it would appear from the above results that neither approximate evaluative method achieves the optimal buffer allocation associated with the optimal throughput. The buffer allocation obtained using the decomposition method is closer to the optimal. There is a tendency for the expansion method to allocate the buffer slots toward the end buffers of the line.

6.4 Concluding Remarks

Taking all the results of this chapter into account, it would appear that simulated annealing is potentially a powerful optimizing technique and, as would be expected, can lead to the optimal solution. Such a technique is of course required in situations such as long lines where complete enumeration is infeasible. Despite the attractiveness of the expansion method in handling parallel-machine station lines, it must be recognized that its use results in a maximum throughput less than what the real line is capable of achieving with the given resources. Decomposition methods indicate higher throughput levels than those indicated by the expansion method but are still less than the exact optimal throughput determined through Markov methods where comparisons are computationally possible. As the line becomes larger and the number of buffer slots to be allocated increases, the throughput indicated using the expansion method tends to the exact result. A major advantage of the expansion method over the Markovian method when treating lines with parallel-machine stations is the faster computation time involved. Given this situation it would appear that the appropriate advice to give to practitioners at this time would be to use simulated annealing and decomposition methods for long lines with single-machine stations and to use simulated annealing or complete enumeration and Markov methods for short lines. If lines with parallel-machine stations are involved, consideration should be given to the use of decomposition methods particularly if the Markovian methods become computationally slow.

6.5 Related Bibliography

Lau and Martin (1986) developed a decision support system for the design of production lines and considered the work-load and buffer allocation problems (WAP and BAP) using simulation.

Hillier and So (1989) considered the pure server allocation problem in short production lines with small or no intermediate buffers and with processing times following the exponential, Erlang and Coxian-2 distributions.

Hillier and So (1996) treated the $W+S$ problem in production lines with variable processing times.

Magazine and Stecke (1996) dealt with the work-load, buffer and server allocation problems in production lines with $K=2$ and 3 stations with parallel servers at each station. They have listed several conjectures.

Futamura (2000) considered the optimal allocation of servers to stations with different service time distributions in tandem queueing networks. The author showed that the coefficient of variation (cv) of the service time distribution converges to unity as the number of servers increases independently of the cv of the individual servers. He examined the server-*cv* interaction.

Tempelmeier (2003) examined the $W+B$ optimization problem by considering it as a work-load optimization problem that includes the pure buffer allocation problem as a sub-problem. The treatment of the service facility is somewhat unusual in that the number of servers is not specified. His objective function is weighted with three factors to minimize the total number of buffer slots, to maximize the mean processing times at each station having in mind upper and lower bounds in the work-load on the stations and to equalize the mean processing times between stations so as to have as balanced a line as possible. A range for the buffer sizes at each buffer is given and a minimum throughput is specified. The author solved first the buffer allocation problem using an algorithm given by Gershwin and Schor (2000) and subsequently he used a greedy heuristic to solve the work-load allocation problem.

References

1. Buzacott, J.A. and Shanthikumar, J.G. (1993), *Stochastic Models of Manufacturing Systems*, Prentice Hall.
2. Dallery, Y. and Frein, Y. (1993), On decomposition methods for tandem queueing networks with blocking, *Operations Research*, Vol. 41, No. 2, pp. 386–399.
3. Diamantidis, A.C., Papadopoulos, C.T., and Heavey, C. (2006), Approximate analysis of serial flow lines with multiple parallel-machine stations, *IIE Transactions*, Vol. 39, issue 4, pp. 361–375.
4. Futamura, K. (2000), The multiple server effect: Optimal allocation of servers to stations with different service-time distributions in tandem queueing networks, *Annals of Operations Research*, Vol. 93, pp. 71–90.
5. Gershwin, S.B. and Schor, J.E. (2000), Effcient algorithms for buffer space allocation, *Annals of Operations Research*, Vol. 93, 1/4, pp. 117–144.
6. Heavey, C., Papadopoulos, H.T., and Browne, J. (1993), The throughput rate of multistation unreliable production lines, *European Journal of Operational Research*, Vol. 68, pp. 69–89.
7. Hillier, F.S. and So, K.C. (1989), The assignment of extra servers to stations in tandem queueing systems with small or no buffers, *Performance Evaluation*, Vol. 10, pp. 219–231.

8. Hillier, F.S. and So, K.C. (1995), On the optimal design of tandem queueing systems with finite buffers, *Queueing Systems*, Vol. 21, pp. 245–266.
9. Hillier, F.S. and So, K.C. (1996), On the simultaneous optimization of server and work allocations in production line systems with variable processing times, *Operations Research*, Vol. 44, No. 3, pp. 435–443.
10. Jain, S. and Smith, J.M. (1994), Open finite queueing networks with $M/M/C/K$ parallel servers, *Computers & Operations Research*, Vol. 21, No. 3, pp. 297–317.
11. Kerbache, L. and MacGregor Smith, J. (1987), The generalized expansion method for open finite queueing networks, *European Journal of Operational Research*, Vol. 32, pp. 448–461.
12. Lau, H.-S. and Martin, G.E. (1986), A decision support system for the design of unpaced production lines, *International Journal of Production Research*, Vol. 24, No. 3, pp. 599–610.
13. Magazine, M.J. and Stecke, K.E. (1996), Throughput for production lines with serial work stations and parallel service facilities, *Performance Evaluation*, Vol. 25, pp. 211–232.
14. Papadopoulos, C.T. and Karagiannis, T.I. (2001), A genetic algorithm approach for the buffer allocation problem in unreliable production lines, *International Journal of Operations and Quantitative Management*, Vol. 7, No. 1, pp. 23–35.
15. Spinellis, D.D. and Papadopoulos, C.T. (2000a), A simulated annealing approach for buffer allocation in reliable production lines, *Annals of Operations Research*, Vol. 93, pp. 373–384.
16. Spinellis, D.D. and Papadopoulos, C.T. (2000b), Stochastic algorithms for buffer allocation in reliable production lines, *Mathematical Problems in Engineering*, Vol. 5, issue 6, pp. 441–458.
17. Spinellis, D., Papadopoulos, C., and MacGregor Smith, J. (2000), Large production line optimization using simulated annealing, *International Journal of Production Research*, Vol. 38, No. 3, pp. 509–541.
18. Tempelmeier, H. (2003), Simultaneous buffer and work-load optimization for asynchronous flow production systems, in *Proceedings of the Fourth Aegean International Conference on the Analysis of Manufacturing Systems*, July, 1–4, 2003, Samos Island, Greece, pp. 31–39.

7

Cost Considerations

A major consideration in the management of production is to understand the cost impact of various designs. Much of the work relating to overall production management including the design of facilities would seem to indicate that the decision processes are serial rather than concurrent or iterative feedback. The classical idea would appear to be that the engineering designers decide on the layout of the line with primary interest in the engineering performance measures of the stations and subsequently this design is costed and justified on some concept closely related to discounted cash flow. There are a number of papers discussing the inadequacy of the approach just outlined, particularly in relation to systems with inherent flexibility and the justification of which may be more strategic rather than tactical.

An appropriate philosophy for world-class companies is for the company to re-invent itself from time to time. This is in contrast to the view that a company can retain its competitive advantage by simply being effective in existing markets. Although it is important to retain or improve one's position in existing markets, it is often in the development of new markets, new customers and new products or services that the health of the company is ensured. Re-invention therefore is a questioning philosophy of how the company can do better today and what it should be doing tomorrow. Clearly, it is a mixture of continuous improvement in all aspects of its activities and new product development. Thus, manufacturing strategy should support marketing strategy and give competitive advantage to the organization. In the light of this strategic trust, investments in production facilities should be carefully assessed from the point of view, of quality, flexibility, time-to-market, dependability, market positioning as well as cost. It is of course difficult to put all these diverse tangible and intangible benefits into one overall metric with the view to choosing particular marketing and manufacturing strategies including production systems designs. From the decision theory point of view what is involved is a multiple-criteria decision problem. However, the authors are unaware of any published work which applies techniques of multiple-criteria decision making such as ranking/scaling methods, analytical hierarchical processes to the selection, at a strategic level of a particular production system. There is an obvious need to reconcile the point of view of the economists, who would be concerned with opportunity

C. T. Papadopoulos et al., *Analysis and Design of Discrete Part Production Lines*, Springer Optimization and Its Applications,
DOI: 10.1007/978-0-387-89494-2_7, © Springer Science+Business Media, LLC 2009

costs, and the approach of accountants who are generally more interested in using actual historical or projected tangible costs, with the imperatives of manufacturing engineering and the operating philosophies and desires of production management, perhaps through a process of brainstorming and Delphi methods. Arising out of the strategic analysis, outline and broadly based decisions on the required manufacturing facilities and capabilities of the company would be developed. For example, decisions might be made to invest in automation to develop flexible manufacturing systems (FMS) or to use production lines. At a level below the strategic level, tactical level decisions must be undertaken to specify, for example, location, size and general layout of manufacturing facility, production machine processes, efficiency/effectiveness, required throughput, capital cost targets, quality targets, degree of flexibility envisaged and operating cost targets for each of the manufacturing systems specified in outline form at the strategic level. Finally, it is at the detailed design level that the production line, if one is required, is completely specified with regard to the number of stations, the equipment at each station, the number of operatives at each station, the inter-station buffer capacities and the work-load allocation to achieve the required throughput. Issues related to maintenance and reliability of equipment need to be considered. The overall objective is to meet the target cost per unit produced. The focus of this book is at this detailed design level.

In general, there are two approaches to the incorporation of cost parameters into the decision process at the detailed design stage of any manufacturing system. One approach is to do all the engineering work essentially without specific reference to costs relying on the experience of the engineers to develop an economical design and subsequently to submit this design for cost evaluation which is generally undertaken by accountants perhaps with some assistance from the engineers. The other approach is to involve implicit decisions in relation to costs concurrently at all stages of the engineering design. To some extent, the different approaches are adopted because of the organization structure of the company, engineers doing the engineering, accountants doing the costing. Clearly, the concurrent evaluation of the cost implication of engineering decisions is the preferable approach. Figures 7.1 and 7.2 illustrate the difference between these two approaches.

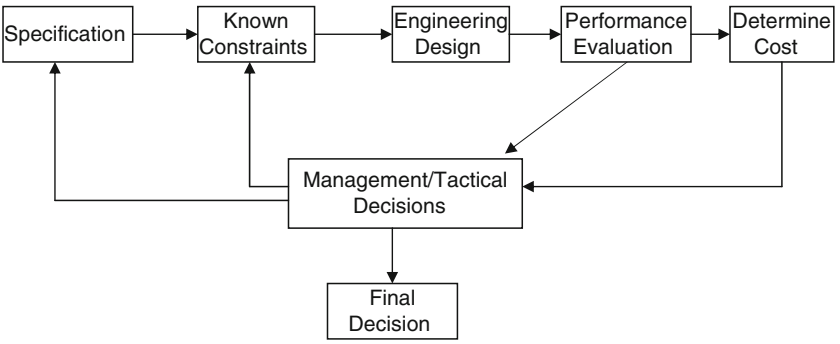


Fig. 7.1. Production line design: Historical approach

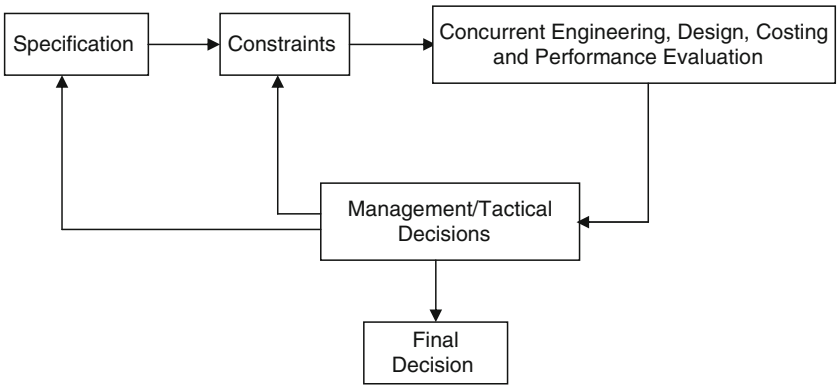


Fig. 7.2. Production line design: Modern approach

It is important to note that following operational experience of a particular implemented design, some re-adjustment and improvements are inevitable which in some cases may even require re-design of sections of the production line in certain circumstances. It is imperative that designers listen carefully to the views of the shop floor personnel and conduct appropriate industrial experiments if there is evidence that some re-engineering is required. The philosophy of continuous improvement should apply to the design process as to all other activities of the organization.

Basically, there are two approaches in developing cost objective functions. The first approach is concerned with the maximization of profit arising from production, while the second approach is concerned with the minimization of costs incurred in production. Because in the analysis one is dealing with stochastic processes, it is the expected values of the random variables of the costs and profits that are optimized based on the expected value of the throughput. More advanced analysis, particularly in relation to risk, would involve the use of measures of variance and higher moments of these random variables. Which particular objective function to use in a given situation is essentially a matter of managerial judgment. One of the advantages of using cost minimization is that there is no need to consider revenue issues or equivalently the price at which the product will be sold. Such minimum cost objective functions are generally used in association with a constraint on the target level of production. In other words, the production line is required to produce a target level of production at minimum cost and so, normally there will be no incentive to go beyond the target production level. The other objective function tends to look upon the production line more as a generator of profit and the production level is an outcome rather than a constraint on the system. Of course, even in the latter case, a minimum production rate might be required in practice, but the optimal profit level may be achieved at a higher production level. Some of the financial figures in cost objective functions are known and deterministic. Others may have to be assessed using estimating procedures such as regression and various types of curve fitting. Other financial figures in these objective functions are essentially stochastic in nature and it is usual to treat

them in terms of their estimated values. Of course, if the financial figures are truly uncertain, then an analytical treatment becomes virtually impossible. The financial parameters placed in cost objective functions are those associated more with management accountancy rather than cost accountancy. This distinction is made here to emphasize that allocated cost data, e.g., depreciation charges or distribution of overhead costs, needs to be subject to careful review before being included as genuine costs in these models.

Below, in Section 7.1, a number of appropriate cost models for profit maximization with increasing degrees of sophistication are presented. Likewise, in Section 7.2, cost minimization models are presented. In all of these models, the assumption is made that a decision has already been taken in regard to the specific type of production system which will be used. Thus, the models are, in effect, tactical level models.

7.1 Cost Models: Profit Maximization

Usually, in cost objective functions there is a need to consider the time value of cash flows. The usual approach is through a discounted cash flow in which cash flow in the future is discounted downwards to the present day. Although the concept of this cash flow is well understood and is analytically tractable, nevertheless a serious technical/managerial issue arises in regard to what interest rate ($I\%$, sometimes called the hurdle rate) to assume in the discounted cash flow calculations. The nature of the different elements in the objective function may be such that some parts of these elements are integer valued only, whereas others may be continuous. This gives rise to the usual computational difficulties associated with such objective functions. Because of the different considerations mentioned in this section, there are a number of different possible formulations of production line design problems involving cost considerations. A number of profit objective functions will be considered below in increasing order of sophistication.

A relatively simple profit objective function would be as follows:

$$\max F_1 = (R-C)X_K^* - C_h\overline{WIP}, \tag{7.1}$$

where

R is the deterministic selling price of a unit of the product.

C is the direct cost associated with each unit produced. This direct cost should include the operative wages cost per unit produced, the cost of material used per unit and the recurrent machine costs per unit produced. In some situations, the latter costs may not be included.

X_K is the normalized throughput of the specified production line consisting of K work-stations and $K-1$ intermediate buffers, whereas X_K^* is the normalized throughput multiplied by the maximum physical throughput of the system, i.e., the X_K^* is the actual physical output per unit time.

$\overline{WIP}$ is the average work-in-process (WIP) summed over all the $K-1$ buffers.

C_h is the inventory holding cost, reminiscent of the same factor, which appears in inventory models, and is a measure of the cost of holding an item in inventory for the same time period as is in the throughput.

Clearly, this rather simple objective function has a number of associated difficult measurement/estimation problems. As indicated above, C, the direct cost, may not be very inclusive because some of the direct costs associated with the production of one unit of product are fuzzy. For example, it may be quite difficult to estimate the per unit product cost of machine repair. Needless to remark, there is no specific assignment of overhead costs and F_1 could not be considered to be a normal profit function as understood by accountants. Sometimes, the term "contribution to profit and overhead" is used to indicate the term $(R-C)$ above. Moreover, it is necessary to specify the system to which objective function (7.1) is to be applied. What is fixed and what may be considered decision variables?

To illustrate the use of cost objective functions in general, the following experiment was undertaken. Perfectly reliable production lines with $K = 3,4,5$ and 6 stations were considered with the same average processing time at each station (balanced lines). The number of total buffer slots to be allocated among the $K-1$ buffers varied from 1 up to 65, depending on the size of the production line.

Consider the output of a production line with K stations in which the finished product is sold at a value of 50 financial units (FU) and costs 40 FU to produce. The contribution margin per unit produced is therefore 10 FU. The assessment of C_h, the average holding cost per unit held in work-in-process (WIP), may be approximated as follows:

$$C_h = \alpha CI,$$

where $0 < \alpha < 1$, C is the cost of production, defined above, and I is the relevant or assigned interest rate per annum. The function of α is to take into account the fact that the value of an item in WIP increases as the item progresses down the line. In the example here, if α is assumed to be 0.5 and $I = 10\%$ per annum, then $C_h = 0.5C(0.1) = 0.05C$. In general, the value of α would be estimated based on material, labor and machine content of the item in inventory.

In this case, define the ratio, r, of the marginal contribution divided by the average unit per annum holding cost as

$$r = \frac{f_1}{f_2} = \frac{R-C}{C_h} = \frac{50-40}{(0.05)(40)} = 5.$$

To continue the illustration, assume the isolated throughput of all the stations is 1 unit per minute (balanced line) and that the facility operates 2 shifts of 8 hours per day for 250 working days per annum. The total maximum output is therefore

$$60 * 2 * 8 * 250 = 240{,}000 \quad \text{units/annum.}$$

The objective function now becomes

$$\max F_1 = f_1 X_K^* - f_2 \overline{WIP} = f_1(240{,}000) X_K - f_2 \overline{WIP}.$$

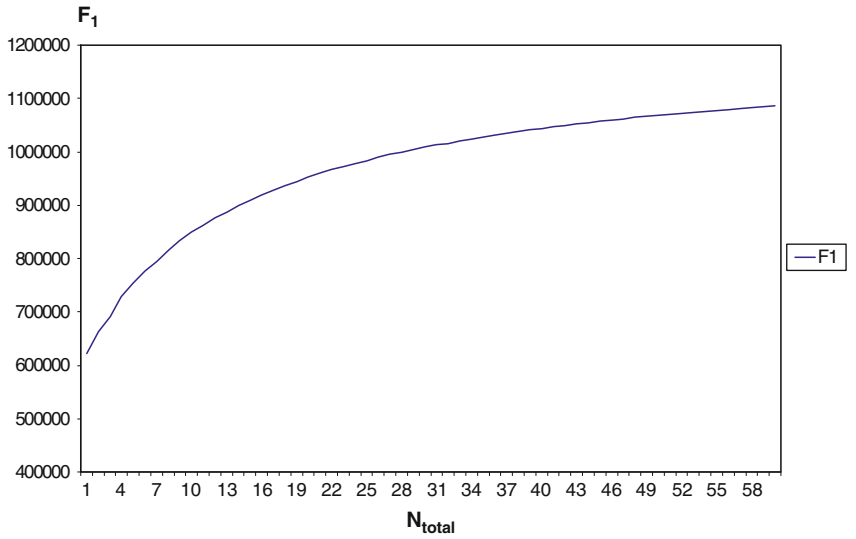


Fig. 7.3. Value of F_1 as a function of N for a 5-station production line with $R = 50$ FU, $C = 40$ FU, $I = 10\%$, $\alpha = 0.5$ ($r = 5$)

This is equivalent to maximizing

$$(240{,}000)rX_K - \overline{WIP} = 1{,}200{,}000X_K - \overline{WIP}.$$

This latter equation indicates that in any conceivable practical situation (where $\overline{WIP} < 1{,}200{,}000X_K$), an objective function of type (7.1), given above, leads to the same optimal operating strategy as the more simple objective function of maximizing throughput for the given number of buffer slots, N. Of course, equation (7.1) assumes that R and C are constants independent of the value of X_K^* and so, for example, there is no overtime premium and all product produced is sold. In Figure 7.3, the value of the objective function for each N for these parameter values is given.

A modification of the profit objective function given in (7.1) would be

$$\max F_2 = \max F_1 - bN = (R - C)X_K^* - C_h\overline{WIP} - bN \tag{7.2}$$

where bN represents the cost on an annual basis of the buffer space used, i.e., each buffer slot costs b financial units (FU) per annum, where physical output, X_K^*, and inventory costs, C_h in FU, are on a per annum basis. The term bN gives the planner some scope for financial justification of a proposed design of a production line in which the number of stations is fixed, the cost of the proposed machines at the stations is given but the decision in relation to the total number of buffer slots, N, is still open. In a practical situation, where the effect of the $\overline{WIP}$ term is small and the buffer allocation associated with maximum throughput for any N is known, it is only necessary to determine the lowest value of N at which the marginal contribution to the first term of F_2 is lower than the cost of providing an extra buffer slot. A slightly

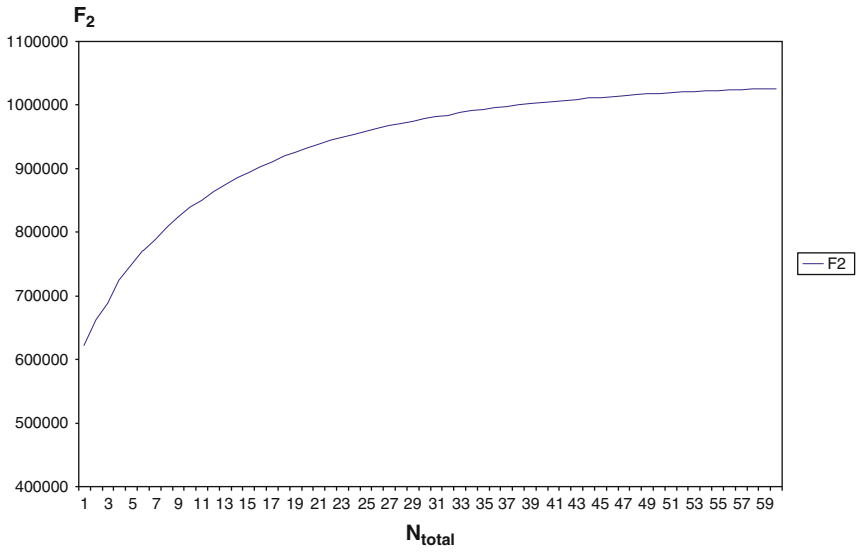


Fig. 7.4. Value of F_2 as a function of N for a 5-station production line with $R = 50$ FU, $C = 40$ FU, $I = 10\%$, $\alpha = 0.5$, $b = 1000$ FU ($r = 5$)

more appropriate objective function might replace the bN term with $\sum_{i=2}^{K} b_i N_i$ where b_i is the cost per annum of providing a buffer slot of type i for each of the N_i slots, $i = 2, 3, \ldots, K$.

It is clear from expression (7.2) that in a practical system, a situation will arise where the marginal advantage of increasing the throughput by the optimal allocation of an additional buffer slot will result in a reduction in the objective function, F_2. This is illustrated in Figures 7.4 and 7.5. Figure 7.4 refers to the following system with parameters: $K = 5$ stations, $R = 50$ FU, $C = 40$ FU, $I = 10\%$, $\alpha = 0.5$, $b = 1000$ FU, whereas, Figure 7.5 refers to a system with parameters as follows: $K = 5$ stations, $R = 50$ FU, $C = 40$ FU, $I = 10\%$, $\alpha = 0.5$, $b = 5000$ FU.

As indicated in Figure 7.4, F_2 increases monotonically as N increases rather similarly as F_1, shown in Figure 7.3. On the other hand, Figure 7.5 indicates that an optimal value of F_2 is achieved and that increasing the total number of slots, N, above a certain level leads to a reduction in the value of the objective function F_2 (from the value in this case of $N = 27$ onwards).

Extending the profit objective function further, it is desirable to bring into consideration discounted cash flows or the time value of money. If one considers a production line to be a generator of cash flows, there are three types of cash flows involved:

- Initial investment in production line facility, i.e., machines, stations, buffer slots, all at time $t = 0$. These flows are considered to be negative.
- Cash flows during the useful life of the production line. These flows would normally be considered to occur with regular frequency and consist of such flows

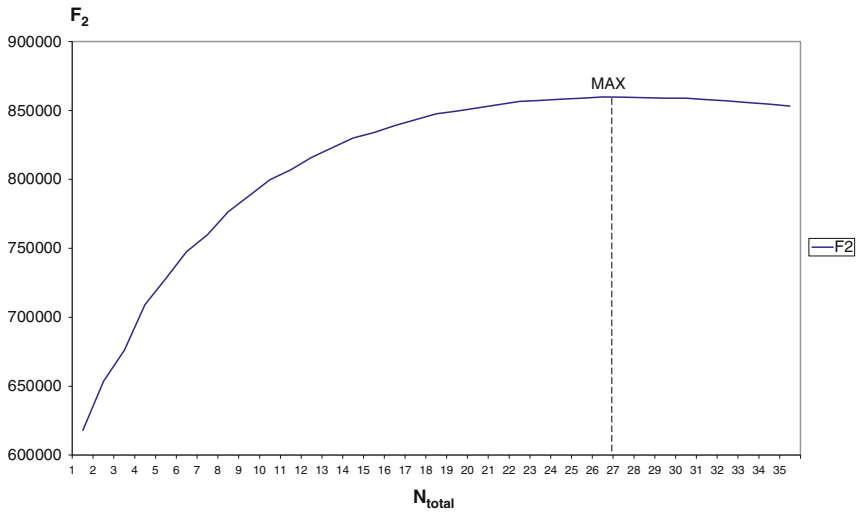


Fig. 7.5. Value of F_2 as a function of N for a 5-station production line with $R = 50$ FU, $C = 40$ FU, $I = 10\%$, $\alpha = 0.5$, $b = 5000$ FU ($r = 5$)

as revenue from sales, wages paid, energy used, materials purchased and used, maintainance/repair costs, etc. In a net sense, these flows would be expected to be positive.

- End of life flows, such as salvage value of machines and buffers, human resources consequences and final disposal of remaining work-in-process (WIP). Normally, these flows could be positive or negative.

In discounted cash flow analysis, the three different types of flows listed above must be treated separately. The basic concept is to develop a present value of all the flows at a chosen time t, usually, $t = 0$. If $t = 0$, the initial investment in the production line facility does not have to be discounted but it is considered negative and is here denoted by Ω_1. On the other hand, the other two cash flows have to be discounted. Taking the end of life flows first and assume that they occur at time $t = T$, each of these flows must be discounted using the following Present Worth Factor, $P.W.F.^*$, of a cash flow received at time T from now ($t = 0$):

$$P.W.F.^* = \frac{1}{(1+I)^T}$$

where I is the interest rate per unit time (usually, per annum).

The above $P.W.F.^*$ is difficult to use in analysis and as a result the concept of continuous discounting is introduced. The idea is basically as follows.

Assume an element of time Δ. The appropriate interest rate during this period of time, Δ, would be $(\Delta)I$ and hence the continuous Present Worth Factor would be approximated by

$$P.W.F.^* = \frac{1}{(1+(\Delta)I)^{\frac{T}{\Delta}}}.$$

As there are in effect T/Δ periods of discounting of duration Δ, as Δ tends to 0, (continuous discounting)

$$P.W.F.^* \to e^{-IT}.$$

Each end of life flow is discounted by using a factor e^{-IT}. The notation Ω_2 is used for the Present Worth Value, P.W.V., of the end of life cash flows, i.e.,

$$\Omega_2 = P.W.V. = (P.W.F.^*)(\text{Net end of life cash flows}) = e^{-IT}\,S$$

where S is the net end of life cash flows.

On the other hand, each cash flow during the useful life of the production line is discounted using the following formulation:

$$\Omega_3 = \int_0^T f(t)e^{-It}dt$$

where $f(t)dt$ is the cash flow between t and $t+dt$ and T is the life of the production line and I is the interest rate. If $f(t)$, $0 \le t \le T$, the cash flow during the operating life of the production line is equal to f, a constant,

$$\Omega_3 = \frac{f}{I}\,[1-e^{-IT}].$$

The reader might note that it may be necessary to modify the functional form of Ω_3 in cases where there are discrete rather than continuous cash flows during the life of the equipment. The modification would entail the addition of terms such as $\mathscr{F} \cdot e^{-It^*}$, where a discrete cash flow of net value $\mathscr{F}$ occurred at time t^*.

A further development of the profit objective functions would be as follows:

$$\max F_3 = P.W.V. = -\Omega_1 + \Omega_2 + \Omega_3 \tag{7.3}$$

where the different cash flows are now discounted. This model can be adopted to incorporate all reasonable cost objective functions.

As an illustration, it may be instructive to modify the above numerical examples as follows:

Scenario 1: Each of the machines costs 200,000 FU including tooling and has a salvage value of 20,000 FU at the end of the fifth year of its operation. Each buffer slot costs 1000 FU including capitalized rent and has a salvage value after five years of 200 FU. $R-C = 10$ FU is revenue per unit less materials and energy costs and $C_h = 2$ FU, $I = 10\%$, $\alpha = 0.5$, and 80,000 FU is a per station annual cost based on a working year of 4000 hours and 20 FU per hour to cover fixed wages and fixed routine repair costs per station. These 4000 hours represent 250 working days of two

eight-hours shifts per day. The mean production rate of the last station is 5 units per minute and therefore, 1,200,000 is the maximum mean production rate of the system per annum working on the basis of two eight-hour shifts per day and 250 working days per annum.

Scenario 2: Each of the machines costs 100,000 FU and has a salvage value of 20,000 FU at the end of the fifth year of its operation. Each buffer slot costs 10,000 FU and has a salvage value after five years of 2000 FU. $R-C=4$ FU, $C_h=1$ FU, $I=10\%$, $\alpha=0.5$, and 40,000 FU is the per station annual cost based on a working year of 2000 hours and 20 FU per hour to cover fixed wages and fixed routine repair costs per station. These 2000 hours represent 250 working days of one eight-hour shift per day. The mean production rate of the last station is 1 unit per minute and therefore, 120,000 is the maximum mean production rate of the system per annum working on the basis of one eight-hour shift per day and 250 working days per annum.

For scenario 1, the following may be determined, for a five-station production line ($K=5$):

$$\begin{aligned}\Omega_1 &= K(200{,}000)+1000N+C_h(\overline{WIP})\\ &= 5(200{,}000)+1000N+2(\overline{WIP})\\ &= 1{,}000{,}000+1000N+2(\overline{WIP}).\end{aligned}$$

In the above expression, the term $2(\overline{WIP})$ arises from valuing the beginning inventory at the same level as it is valued during the life of the production line and at the end of the life of the production line. This assumption could of course be modified,

$$\begin{aligned}\Omega_2 &= \left\{K(20{,}000)+200N+C_h(\overline{WIP})\right\}(e^{-5I})\\ &= \left\{5(20{,}000)+200N+2(\overline{WIP})\right\}(e^{-5I})\\ &= \left[100{,}000+200N+2\overline{WIP}\right](e^{-0.5})\\ &= \left[100{,}000+200N+2\overline{WIP}\right](0.6065)\end{aligned}$$

$$\begin{aligned}\Omega_3 &= \{(R-C)1{,}200{,}000X_K-80{,}000K\}\left(\frac{1-e^{-5I}}{I}\right)\\ &= [12{,}000{,}000X_K-400{,}000]\left(\frac{1-e^{-5(0.1)}}{0.1}\right)\\ &= [120{,}000{,}000X_K-4{,}000{,}000]\left(1-e^{-0.5}\right)\\ &= [120{,}000{,}000X_K-4{,}000{,}000](0.3935)\\ &= (47.22X_K-1.574)10^6\end{aligned}$$

leading to the following objective function:

$$\begin{aligned}\max F_3 &= -\Omega_1+\Omega_2+\Omega_3\\ &= -2{,}513{,}350-878.7N-0.787\overline{WIP}+(47.22)(10^6)X_K.\end{aligned}$$

For scenario 2, F_3 may be determined in a similar fashion leading to F_3 in this case being as follows:

$$\begin{aligned}\max F_3 &= -\Omega_1 + \Omega_2 + \Omega_3 \\ &= -1{,}226{,}350 - 8787N - 0.3935\overline{WIP} + 1{,}888{,}800X_K\end{aligned}$$

Figure 7.6 indicates the value of the objective function, F_3, for scenario 1, above. As may be seen, F_3 is a monotonically increasing function of N. Figure 7.7 shows the value of the objective function, F_3, for scenario 2, which attains its optimal value at $N = 26$.

It is interesting to note the form of F_1, F_2 and F_3 as developed for the above examples. In general, for this type of analysis the form of these objective functions is:

$$\max F = \alpha X_K - \beta\,\overline{WIP} - \gamma N$$

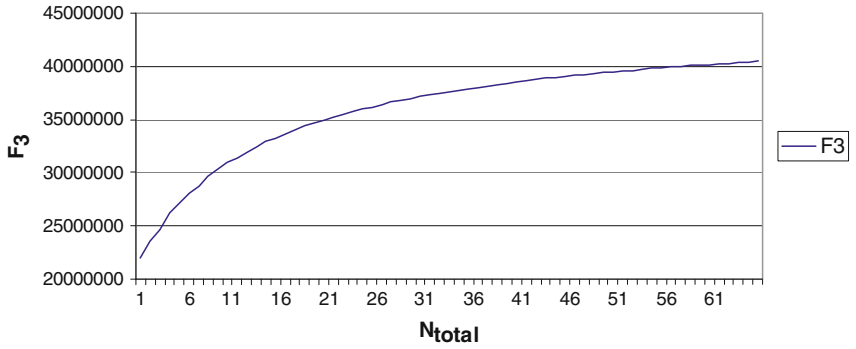


Fig. 7.6. Value of F_3 as a function of N for a 5-station balanced production line with $R - C = 10$ FU, $I = 10\%$, $\alpha = 0.5$, $C_h = 2$ FU, $b = 1000$ FU, 2 shifts per day, 5 units per minute maximum mean production rate of the system

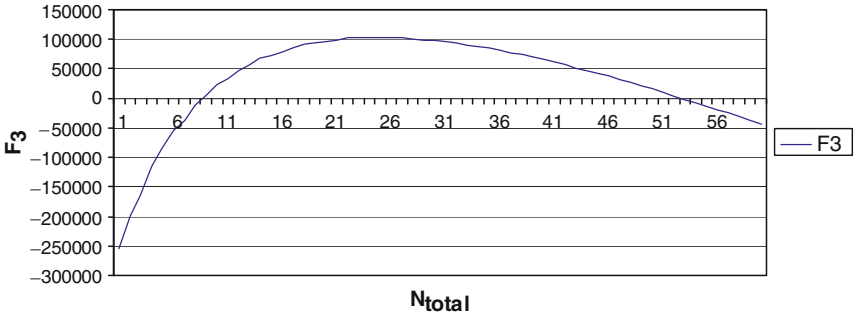


Fig. 7.7. Value of F_3 as a function of N for a 5-station balanced production line with $R - C = 4$ FU, $I = 10\%$, $\alpha = 0.5$, $C_h = 1$ FU, $b = 10{,}000$ FU, 1 shift per day, 1 unit per minute maximum mean production rate of the system

where $\alpha, \beta, \gamma > 0$. In many practical situations, α is large while the β and γ are much less than the α. Clearly, $\overline{WIP} \leq N$. So, for these lines, it is reasonable to say that the objective function is a function of X_K and N. If there is no cost associated with the provision of buffer space ($\gamma = 0$, the case of F_1 above), clearly, there is really no need to use a cost-based objective function as a simpler objective function of maximizing throughput will give the same result, always assuming that the contribution of $\beta\,\overline{WIP}$ does not lead to a restriction on N. It would appear that the only case of significant practical interest is when there is a limit on the size of N from a cost aspect, in other words, buffer sizes beyond this N in effect reduce the value of the cost objective function.

The reader may have some observations about the values of the parameters used in some of the examples presented above, but the intention is to illustrate the different cases which may arise rather than any particular practical example. The reader will note that an underlying assumption in the above examples is that the lines are balanced.

7.2 Cost Models: Cost Minimization

Up to now, the maximization of profit was the objective function. Attention is now turned to cost minimization objective functions where the objective function consists of a sum of costs (perhaps including present worth value terms) which is to be minimized and the set of constraints includes a production throughput target which must be at least achieved.

Consider a balanced production line with K stations. A total buffer capacity of N slots are available for allocation. The production line has to meet a target production level of at least X_0 at overall minimum cost.

The problem formulation is as follows:

$$\min G_1 = \min \left\{ \sum_{i=2}^{K} b_i N_i + C_h \overline{WIP} + \gamma (X_K - X_0) \right\} \tag{7.4}$$

s.t.

$X_K \geq X_0$

$N_{i,\min} \leq N_i \leq N_{i,\max}$

$\sum_{i=2}^{K} N_i \leq N$

Each $N_i, i = 2, \ldots, K$ is an integer,

where

$b_i, i = 2, \ldots, K$, may be a net present value type cost coefficient associated with each buffer slot.

C_h could also incorporate a net present value coefficient because of the cost of holding average work-in-process, $\overline{WIP}$, that is incurred throughout the life of the project.

γ is a linear penalty cost associated with going above the required production target, X_0, and it could include considerations of present value. γ would normally

be very large in comparison to C_h, in practical situations as a deviation of the mean production rate from the required target level, X_0, is heavily penalized.

In this formulation, the concept of a constraint on the size of the buffer at each location i is incorporated, where $N_{i,\min}$ is the lower bound and $N_{i,\max}$ the upper bound, respectively, of $N_i, i = 2,\ldots,K$. The usual assumption of integer N_i is made.

It should be observed that the objective function assumes that the N available buffer slots are not necessarilly all to be utilized, whereas it may be possible to achieve the desired production level at a lower overall cost while not utilizing all available buffer slots. The implication is, of course, that available buffer slots not used are not part of the objective function.

The computational procedure is as follows (assuming for simplicity a balanced line with K stations):

1. Select $\sum_{i=2}^{K} N_i = N^* \leq N$.
2. With this N^* determine the value of G_1 for different buffer allocations which meet the throughput requirement that $X_K \geq X_0$.
3. Determine the minimum G_1 from those obtained in step 2, above. It should be noted that in step 2 the objective function, in fact, reduces to minimization of $C_h\overline{WIP} + \gamma(X_K - X_0)$ if $b_i = b$ for all $i = 2,\ldots,K$.
4. Search the next value of N^* and determine if an improvement can be made in the value of G_1 and continue until $\sum_{i=2}^{K} N_i = N^* \leq N$ is the minimum N^* which results in the minimum of the objective function G_1 while meeting the throughput constraint.

   Should there be a tie, i.e., when the minimum cost is achieved for two different values of N^*, break ties arbitrarily.

A deviation on the above balanced production line would be to assume a non-balanced production line, where w_i, the work-load at station i, is specified at each i, $i = 1,\ldots,K$, and is outside the control of the system designer. In this latter case, similar observations would lead to the same conclusion, viz., the cost objective function could be reduced to one of minimizing $\overline{WIP}$.

Below, two different examples are considered.

Example 1: $K = 5$ stations, $b_i = b = 1000$ FU, $i = 2,3,4,5$. $C_h = 2$ FU, $X_0 = 0.80$, $\gamma = 180{,}000$ FU. The value of γ was determined on the basis of a one shift system of 8 hours per day, 250 working days per annum and a maximum throughput of the last station of 30 units per hour. A nominal value of 3 FU for each unit of excess product produced was assumed. This nominal value would include such costs as material and labor costs, scrap values and disposal costs. A maximum of $N = 32$ buffer slots are available for allocation among the the four (4) intermediate buffers of the system.

The problem may now be formulated as follows:

$$\min G_1 = \min\left\{\sum_{i=2}^{5} 1{,}000N_i + 2\overline{WIP} + 180{,}000(X_K - 0.80)\right\}$$

s.t.
$X_K \geq 0.80$
$2 \leq N_i \leq 10,\ i = 2,3,4,5$
$\sum_{i=2}^{K} N_i \leq 32$
Each $N_i, i = 2,\ldots,5$ is an integer.

Numerical investigation of the above problem using the objective function, G_1, as specified above, leads to the following results: $N_2 = 10$, $N_3 = 3$, $N_4 = 7$ and $N_5 = 10$; $X_K = 0.806982$; $\overline{WIP} = 17.588877$. As expected, the BAP-B problem gives identical results, but it has a simpler objective function, i.e., $\min \overline{WIP}$.

Example 2: $K = 5$-stations balanced line,

$$b_2 = 10{,}000, b_3 = 1{,}000, b_4 = 1{,}500, b_5 = 1{,}400 \text{ FU}.$$
$$C_h = 2\text{FU}, X_0 = 0.80, \gamma = 180{,}000 \text{ FU},$$

as in example 1. Again, a maximum of $N = 32$ buffer slots are available for allocation among the four (4) intermediate buffers of the system.

The problem may now be formulated as follows:

$$\begin{aligned}\min G_1 = \min \{ & (10{,}000N_2 + 1{,}000N_3 + 1{,}500N_4 + 1{,}400N_5) \\ & + 2\overline{WIP} + 180{,}000(X_K - 0.80)\}\end{aligned}$$

s.t.
$X_K \geq 0.80$
$2 \leq N_i \leq 10,\ i = 2,3,4,5$
$\sum_{i=2}^{K} N_i \leq 32$
Each $N_i, i = 2,\ldots,K$ is an integer.

Numerical investigation of the above problem using the objective function, G_1, as specified above, leads to the following results: $N_2 = 3$, $N_3 = 10$, $N_4 = 7$ and $N_5 = 10$; $X_K = 0.814949$; $\overline{WIP} = 13.914076$.

If the design involves the allocation of work to each station, $w_i, i = 1,\ldots,K$, and the buffer allocation has already been decided, viz., $N_i, i = 2,\ldots,K$, are given, a minimization cost objective function and associated constraints may be formulated as follows:

$$\min G_2 = \min \left\{ C_h \overline{WIP} + \sum_{i=1}^{K} [\varepsilon_i | w_i - \overline{w_i}| + \eta_i (w_i - 1)\delta(w_i - 1)] \right\} \tag{7.5}$$

s.t.
$X_K \geq X_0$
$\sum_{i=1}^{K} w_i = K,$
$w_{i,\min} \leq w_i \leq w_{i,\max},$

$N_i,\ i = 2,\ldots,K$ are a feasible set of specified integer values which will allow for the achievement of the required throughput, X_0. $w_{i,\min}$ and $w_{i,\max}$ are, respectively, the lower and upper bounds of the w_i's, $i = 1,\ldots,K$, acceptable to the respective stations.

A set of feasible N_i's may be obtained from the solution of the BAP-C problem, i.e.,

$\min \sum_{i=2}^{K} N_i$

s.t.

$X_K \geq X_0$

$\sum_{i=2}^{K} N_i \leq N$,

$N_{i,\min} \leq N_i \leq N_{i,\max}$, $i = 2, \ldots, K$

Each $N_i, i = 2, \ldots, K$ is an integer.

Clearly, other feasible sets of N_i's could be specified knowing the results of this sub-problem.

$\delta(w_i - 1)$ is the Kronecker delta function defined as follows:

$$\delta(a) = \begin{cases} 1, & \text{if } a > 0 \\ 0, & \text{if } a \leq 0. \end{cases}$$

The term in the objective function involving the coefficient ε_i is a present worth value of costs associated with operating machines away from their natural design speeds, $\overline{w_i}, i = 1, \ldots, K$. The concept is that machines have a natural design speed and that running a machine above or below this design speed incurs a linear penalty. Although the format of the penalty in the above objective function involves an absolute magnitute expression, it is possible for analytical reasons to use a term such as $(w_i - \overline{w_i})^2$ in place of $|w_i - \overline{w_i}|$, particularly in the case of small deviations. If desired, the factor ε_i can be adjusted by simple numerical means to make the two expressions equivalent at a particular deviation level. The term involving the coefficient η_i captures the present worth value of the extra wages which in some circumstances may be paid to the operators arising out of the unequal work-load at the stations. The payment would be only applicable to those operators working in stations where the work-load was greater than the average normalized value of 1.

As may be seen from the above, it becomes increasingly more difficult to be precise about what costs to include in the objective function and the problem of the assessment of these costs is by no means trivial. Consequently, a different approach may be adopted, whereby, in conjunction with management, the designer specifies what is essentially a wish list of objectives and places weights on these objectives. For example, an objective function could be formed using the following:

- Minimize the total number of buffers.
- Maximize the service rates.
- Equalize the service times.
- Minimize the average work-in-process, $\overline{WIP}$.
- Minimize the deviation of the throughput from the target throughput.

The objective function in this case could be:

$$\min G_3 = \min \left\{ f_1 \sum_{i=2}^{K} N_i + f_2 \sum_{i=1}^{K} (w_{i,\max} - w_i) + f_3 \sum_{i=1}^{K} (w_i - 1)^2 + f_4 \overline{WIP} + f_5 (X_K - X_0) \right\} \tag{7.6}$$

s.t.
$X_K \geq X_0$,
$\sum_{i=1}^{K} w_i = K$,
where
N_i, $i = 2, \ldots, K$ are integer and $w_{i,\max}$ is the maximum w_i acceptable to station i, $i = 1, \ldots, K$.

The optimization problem could be formulated using the usual constraints. The values of weights $f_i, i = 1, \ldots, 5$, would be determined in consultation between the designers and the managers of the system with some regard given to the magnitude of the different costs involved using perhaps brainstorming or a Delphi approach. AHP, other paired comparison methods and other multicriterion decision methodologies would also have a place in this approach. It might be also noted that this approach could be formulated via a goal programming format.

Generally speaking, it must be remembered that the coefficient of the average WIP term would tend to be small relative to the coefficients of other terms in any of the above objective functions. This should not lead to the conclusion that minimizing $\overline{WIP}$ is never a sensible objective function in the design of production lines as has already been shown in the examples presented above. In all cases where cost considerations are involved, it would be very desirable for the designer to test the sensitivity of the design to changes in the costs figures used in the objective function.

References

1. Altiok, T. (1997), *Performance Analysis of Manufacturing Systems*, Springer-Verlag.
2. Altiok, T. and Stidham, S. Jr. (1983), The allocation of interstage buffer capacities in production lines, *AIIE Transactions*, Vol. 15, No. 4, pp. 292–299.
3. Beged Dov, A.G., Carmichael, C.D., Ferguson, S.T., Mitchel, I.E., and Strube, W.H. (1968), Production programming by revenue curve analysis, *Proceedings of the Second Winter Simulation Conference on Applications of Simulations*, pp. 53–57.
4. Blank, L. and Carrasco, H. (1985), The economics of new technology: System design and development methodology, *Annual International Industrial Engineering Conference Proceedings*, pp. 161–168.
5. Boer, C.R. and Metzler, V. (1986), Economic evaluation of advanced manufacturing by means of simulation, *Material Flow*, Vol. 3, pp. 215–224.
6. Boothroyd, G. (1982), Economics of Assembly Systems, *Journal of Manufacturing Systems*, Vol. 1, No. 1, pp. 111–126.
7. Canada, J.R. and Sullivan, W.G. (1990), Persistent pifalls and applicable approaches for justification of advanced manufacturing systems, *Engineering Costs and Production Economics*, Vol. 18, pp. 247–253.
8. Carrasco, H. and Blank, L. (1987), Prototype development of an investment tracking system for life cycle costing, *World Productivity Forum & 1987 International Industrial Engineering Conference Proceedings*, pp. 211–216.
9. Christy, P.D. and Kleindorfer, B.G. (1990), Simultaneous cost and production analysis of manufacturing systems, *Proceedings of the Winter Simulation Conference*, pp. 582–589.
10. Gustavson, R.E. (1983), Choosing manufacturing systems based on unit cost, *Proceedings of the 13th International Symposium on Industrial Robots and Robots*, Vol. 1, pp. 4-85–4-104.

11. Haider, S.W. and Blank, L.T. (1983), A role for computer simulation in the economic analysis of manufacturing systems, *Proceedings of the Winter Simulation Conference*, pp. 199–206.
12. Hedge, G.G., Ramamurthy, K., Tadikamalla, P.R., and Kekre, S. (1988), Capacity choice, work-in-process inventory and throughput: A simulation study, *Proceedings of the Winter Simulation Conference*, pp. 662–666.
13. Helber, S. (1999), *Performance Analysis of Flow Lines with Non-Linear Flow of Material*, Springer-Verlag.
14. Hutchinson, G.K. and Holland, J.R. (1982), The economic value of flexible automation, *Journal of Manufacturing Systems*, Vol. 1, No. 2, pp. 215–227.
15. Jensen, P.A., Pakath, R., and Wilson, J.R. (1988), Optimal buffer inventories for multistage production systems with failures, SMM 88-1, School of Industrial Engineering, Purdue University.
16. Jeong, K.C. and Kim, Y.-D. (2000), Heuristics for selecting machines and determining buffer capacities in assembly systems, *Computers & Industrial Engineering*, Vol. 38, pp. 341–360.
17. Karmarkar, U. (1987), Manufacturing configuration, capacity and mix decisions considering operational costs, *Journal of Manufacturing Systems*, Vol. 6, Part 4, pp. 315–324.
18. Meredith, J.R. and Suresh, N.C. (1986), Justification techniques for advanced manufacturing technologies, *International Journal of Production Research*, Vol. 24, No. 5, pp. 1043–1057.
19. Moerman, P.A. (1988), Economic evaluation of investments in new production technologies, *Engineering Costs and Production Economics*, Vol. 13, pp. 241–262.
20. Monga, A. and Zuo, M.J. (2001), Optimal design of series-parallel systems considering maintenance and salvage value, *Computers & Industrial Engineering*, Vol. 40, No. 4, pp. 323–337.
21. Noble, J.S. and Tanchoco, J.M.A. (1993), Design justification of manufacturing systems – A review, *The International Journal of Flexible Manufacturing Systems*, Vol. 5, pp. 5–25.
22. Smith, J.M. and Chikhale, N. (1994), Buffer allocation for a class of non-linear stochastic knapsack problems, Department of Industrial Engineering and Operations Research, University of Massachusetts, Amherst.
23. Smith, J.M. and Daskalaki, S. (1988), Buffer space allocation in automated assembly lines, *Operations Research*, Vol. 36, pp. 343–358.
24. Son, Y.K. (1991), A cost estimation model for advanced manufacturing systems, *International Journal of Production Research*, Vol. 29, No. 3, pp. 441–452.
25. Spinellis, D., Papadopoulos, C., and MacGregor Smith, J. (2000), Large production line optimization using simulated annealing, *International Journal of Production Research*, Vol. 38, No. 3, pp. 509–541.
26. Tempelmeier, H. (2003), Simultaneous buffer and work-load optimization for asynchronous flow production systems, In *Proceedings of the Fourth Aegean International Conference on the Analysis of Manufacturing Systems*, July, 1–4, 2003, Samos Island, Greece, pp. 31–39.
27. Yeralan, S., Dieck, A.J., and Darwin, R.F. (1986), Economically optimum maintenance, repair and buffering operations in manufacturing systems, *The Engineering Economist*, Vol. 31, No. 4, pp. 279–292.

A

Mathematical Fundamentals

In this appendix, the objective is to give an outline review of those areas of mathematics including probability and statistics which are essential background for a complete understanding of the material covered in the main text. It is assumed that many readers would have no need for this appendix. Of necessity, the treatment is rather brief and has an engineering rather than a pure mathematical orientation and with some emphasis on numerical examples.

A.1 Vectors and Matrices

A.1.1 Vectors

The definition of a vector is an ordered set of numbers which in applications to manufacturing systems are generally real numbers, e.g., vector $v = (v_1, v_2, v_3, v_4)$ is a vector with four components or elements and each component is a member of a set of real numbers. The ordering relates to the position of the components in the array. A numerical example would be $\mathbf{v} = (4,7,8,5)$ where the third component of the $[1 \times 4]$ vector is 8. The vector $\phi = (0,0,\ldots,0)$ is known as the null vector. Specific vector operations are shown numerically below.

Given $\alpha = 3$ and $\mathbf{v} = (v_1, v_2, v_3, v_4) = (3,5,8,2)$, then $\alpha\mathbf{v} = (9,15,24,6)$ is a $[1 \times 4]$ vector. The transpose of a $[1 \times n]$ row vector, $\mathbf{v}$, is a $[n \times 1]$ column vector, denoted by v^T, as illustrated below.

$$\mathbf{v} = (2,1,4,7), \quad \mathbf{v^T} = \begin{bmatrix} 2 \\ 1 \\ 4 \\ 7 \end{bmatrix}$$

If

$$\mathbf{c} = (4,3,2,1)$$

C. T. Papadopoulos et al., *Analysis and Design of Discrete Part Production Lines*, Springer Optimization and Its Applications,
DOI: 10.1007/978-0-387-89494-2_8, © Springer Science+Business Media, LLC 2009

and

$$\mathbf{d} = (5,6,9,12)$$

both $[1 \times 4]$ row vectors, then the sum of the vectors is

$$\mathbf{c} + \mathbf{d} = (9,9,11,13)$$

and the subtraction of the vectors is

$$\mathbf{c} - \mathbf{d} = (-1,-3,-7,-11).$$

A.1.2 Matrices

A matrix R of dimension $(m \times n)$ is a rectangular array of real numbers arranged in m rows and n columns as follows:

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix}$$

a $(m \times n)$ matrix.

If R and S are (3×3) matrices as given below:

$$R = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$S = \begin{bmatrix} 1 & 1 & 1 \\ 3 & 4 & 5 \\ 1 & 3 & 1 \end{bmatrix}$$

then the matrix $R+S$ is the following (3×3) matrix

$$R+S = \begin{bmatrix} 2 & 3 & 4 \\ 7 & 9 & 11 \\ 8 & 11 & 10 \end{bmatrix}$$

and the matrix $R-S$ is the following (3×3) matrix

$$R-S = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 1 & 1 \\ 6 & 5 & 8 \end{bmatrix}.$$

If $\alpha = 2$, then the matrix $T = \alpha R$ is given by

$$T = \alpha R = \begin{bmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \\ 14 & 16 & 18 \end{bmatrix}$$

The transpose A^T of matrix A of dimension $(m \times n)$ is a matrix of dimension $(n \times m)$ derived as follows:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

$$A^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \cdots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}.$$

For example, if A is the following (4×2) matrix:

$$A = \begin{bmatrix} 2 & 4 \\ 7 & 9 \\ 8 & 10 \\ 11 & -5 \end{bmatrix}$$

A^T is the (2×4) matrix:

$$A^T = \begin{bmatrix} 2 & 7 & 8 & 11 \\ 4 & 9 & 10 & -5 \end{bmatrix}.$$

The product CD of matrices C and D is defined only if the number of columns of C is equal to the number of rows of D. If C has dimension $(m \times r)$ and D has dimension $(r \times n)$, then CD has dimesnion $(m \times n)$ with the (i, j)th element of CD given by

$$\sum_{\kappa=1}^{n} C_{ik} D_{kj}.$$

Conceptually, the elements of CD are the inner or dot products of the appropriate row and column of C and D, respectively.

Given the (2×3) matrix, C:

$$C = \begin{bmatrix} 2 & 4 & 6 \\ 7 & 1 & 2 \end{bmatrix}$$

and the (3×4) matrix D:

$$D = \begin{bmatrix} 9 & 2 & 1 & 4 \\ 1 & 2 & 2 & 1 \\ 8 & 2 & 3 & 6 \end{bmatrix}$$

CD is the following (2×4) matrix:

$$CD = \begin{bmatrix} 70 & 24 & 28 & 48 \\ 80 & 20 & 15 & 41 \end{bmatrix}.$$

Note that DC given C and D above has no meaning.

Consider the first row of C as a row vector E and the second column of D as a column vector $\mathbf{F}$ where

$$E = (2,4,6)$$
$$\mathbf{F} = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix},$$

then the *inner or dot product* of the two vectors $E \cdot F = 24$ which is the (1,2) element of CD.

Matrix operations have the following general properties:

$$\begin{aligned} (C+D)+B &= C+(D+B) = C+D+B \\ C+B &= B+C \\ (C+D)B &= CB+DB \\ (CD)B &= C(DB) = CDB \\ (B+C)^T &= B^T+C^T \\ (AB)^T &= B^TA^T. \end{aligned}$$

If $C^T = C$, then C is said to be a symmetric matrix.

The identity matrix, I, is an $(n \times n)$ matrix as follows:

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

The elements of I are either 1 or 0 and the elements with value 1 are placed on what is termed the principal diagonal of the matrix.

Given a square matrix C of dimension $(n \times n)$ and another square matrix D of the same dimension, then, if

$$CD = I$$

D is said to be the inverse of C and C is said to be the inverse of D.

If C and D are as follows:

$$C = \begin{bmatrix} 1 & 5 \\ 4 & 2 \end{bmatrix}; \quad D = -\frac{1}{18}\begin{bmatrix} 2 & -5 \\ -4 & 1 \end{bmatrix}$$

$$CD = I; \quad DC = I$$

and so D is the inverse of C and C is the inverse of D.

There are a number of efficient computer packages available to determine the inverse of matrices. The inverse of a matrix may be determined by way of elementary transformations/operations as indicated below.

To determine the inverse of C:

$$C = \begin{bmatrix} 1 & 5 \\ 4 & 2 \end{bmatrix}$$

$$\begin{array}{cccccl} 1 & 0 & 1 & 5 & (i) & \\ 0 & 1 & 4 & 2 & (ii) & \\ 1 & -2.5 & -9 & 0 & (iii) & = (i) - 2.5(ii) \\ -\frac{1}{9} & \frac{2.5}{9} & 1 & 0 & (iv) & = (iii)/(-9) \\ -\frac{4}{9} & \frac{10}{9} & 4 & 0 & (v) & = 4(iv) \\ \frac{4}{9} & -\frac{1}{9} & 0 & 2 & (vi) & = (ii) - (v) \\ \frac{2}{9} & -\frac{1}{18} & 0 & 1 & (vii) & = (vi)/2 \end{array} .$$

D from (iv) and (vii) is

$$D = \begin{bmatrix} -\frac{1}{9} & \frac{2.5}{9} \\ \frac{2}{9} & -\frac{1}{18} \end{bmatrix} = -\frac{1}{18}\begin{bmatrix} 2 & -5 \\ -4 & 1 \end{bmatrix}$$

which is the inverse of C, as shown above.

If a square matrix C has an inverse, the inverse may be shown to be unique. A square matrix which does not have an inverse is said to be singular. For example, the following (2×2) matrix C:

$$C = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$$

does not have an inverse. To show that this is the case by contradiction, assume that D is the inverse:

$$D = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

As $CD = I$ then

$$\begin{array}{r} a + 2c = 1 \quad (i) \\ b + 2d = 0 \quad (ii) \\ 2a + 4c = 0 \quad (iii) \\ 2a + 4c = 1 \quad (iv) \end{array}.$$

(i) and (iii) are incompatible as are (ii) and (iv) and so C does not have an inverse.

Determining the inverse of matrices of high order using classical methods is tedious; one such method involves a well-known mathematical process of obtaining the determinant of the square matrix and then deriving the determinants of the adjoint matrices. Interested readers are referred to standard textbooks on linear algebra. A simple example of the process involved is given below:

$$C = \begin{bmatrix} 1 & 5 \\ 4 & 2 \end{bmatrix}$$

The determinant of C, denoted by $||C||$, is given by

$$||C|| = 1 \times 2 - 5 \times 4 = -18.$$

Determinants of the adjoint matrices are obtained by deleting appropriate rows and columns of the parent matrix:

$$||C_{11}|| = 2; \quad ||C_{12}|| = 4; \quad ||C_{21}|| = 5; \quad ||C_{22}|| = 1$$

$$D = C^{-1} = -\frac{1}{18}\begin{bmatrix} (-1)^2 2 & (-1)^3 5 \\ (-1)^3 4 & (-1)^4 1 \end{bmatrix} = -\frac{1}{18}\begin{bmatrix} 2 & -5 \\ -4 & 1 \end{bmatrix}$$

as above.

In the analysis of queues it is sometimes useful to use the notation e^{At}, where,

$$e^{At} = I + At + \frac{1}{2!}A^2t^2 + \frac{1}{3!}A^3t^3 + \cdots$$

It may be easily shown that

$$\frac{d}{dt}\{e^{At}\} = Ae^{At} \quad or \quad [e^{At}]A$$

where A is a square matrix.

Readers are referred to standard texts on linear algebra to determine e^{At}. The following particular procedure may be used to determine e^{At}.

Given

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix},$$

let $[F(t)] = [e^{[A]t}]$ and $F(0) = I$. Knowing that $\frac{d}{dt}e^{[A]t} = [A]e^{[A]t}$ and taking the Laplace transforms of both sides

$$[s\widetilde{F}(s) - I] = [A][\widetilde{F}(s)]$$
$$[sI - A]\widetilde{F}(s) = I$$
$$[\widetilde{F}(s)] = [sI - A]^{-1}$$

where $[\widetilde{F}(s)]$ is the Laplace transform of $[F(t)]$.

$$\begin{aligned}[e^{At}] &= \mathscr{L}^{-1}[\widetilde{F}(s)] = \mathscr{L}^{-1}[sI - A]^{-1} \\ &= \mathscr{L}^{-1}\begin{bmatrix} \frac{1}{s-1} & \frac{2}{(s-1)(s-3)} \\ 0 & \frac{1}{s-3} \end{bmatrix} \\ &= \begin{bmatrix} e^t & e^{3t} - e^t \\ 0 & e^{3t} \end{bmatrix}.\end{aligned}$$

Note that

$$\frac{2}{(s-1)(s-3)} = \frac{1}{s-3} - \frac{1}{s-1}$$

and

$$\mathscr{L}^{-1}\left[\frac{1}{s-\alpha}\right] = e^{\alpha t}.$$

A.2 Probability

In probability theory, an experiment is a well-defined process, the observed outcome of which is not known in advance. The set of all possible outcomes is the sample space. Depending on the experiment, the sample space can consist of a countable number of outcomes, in which case, the sample space is described as discrete. If the sample space does not consist of a countable number of outcomes, it is described as continuous. A random variable is a function that assigns a value to every element of the sample space. Thus random variables are generally either discrete or continuous, but may be hybrid. Associated with each discrete value of a random variable is a probability mass function and associated with each continuous value of a random variable is a probability density function . An example of a discrete random variable would be the outcome of an experiment involving the determination of the sum of two throws of two fair dice. The sample space, here, would be the discrete integer

values from 2 to 12 and the associated probability mass function is given below, where T_1 is the random variable indicating the outcome of the first throw and T_2 is the random variable indicating the outcome of the second throw and X is the random variable which is the sum of T_1 and T_2.

The mean value or the expected value of any random variable is a measure of its central tendency and in the case of a discrete distribution is defined as follows

$$EX = \sum_{i=1}^{n} x_i p_i,$$

where x_i is a value of the discrete random variable and p_i is the associated probability mass function (probability of occurence).

In the above example

$$EX = \sum_{i=1}^{11} x_i p_i = 7.$$

There are 11 different values of the random variable $X = T_1 + T_2$ and the respective probabilities $p_i, i = 2, \ldots, 11$ are given in Table A.1.

The probability mass function of $X = T_1 + T_2$ is symmetric (see Table A.1). The reader should note that whereas the probability mass functions of T_1 and T_2 are both uniform (probabilities of all possible events in each case being 1/6 as the die is fair and the throws are independent), the probability mass function of the discrete random variable $X = T_1 + T_2$ has a triangular form. The process of deriving the distribution of a random variable which is the sum of other random variables is known as convolution. The example given above, in a very basic way, illustrates the fundamentals of the Central Limit Theorem in that a rudimentary form of a normal distribution has evolved from the convolution of two uniform probability mass functions.

The variance of a discrete random variable, X, is defined as

$$VarX = \sigma_X^2 = E(X - EX)^2 = \sum_{i=1}^{n} (x_i - EX)^2 p_i$$

Table A.1. Probability mass function

$X = T_1 + T_2$	Probability mass function
2	1/36
3	2/36
4	3/36
5	4/36
6	5/36
7	6/36
8	5/36
9	4/36
10	3/36
11	2/36
12	1/36

where σ_X is the standard deviation of the random variable X. For the above example, the variance of $X = T_1 + T_2$ is

$$VarX = \sigma_X^2 = \sum_{i=1}^{11}(x_i - EX)^2 p_i = \sum_{i=1}^{11} x_i^2 p_i - (EX)^2 = EX^2 - (EX)^2 = 35/6 = 5.833.$$

The reader will note that with respect to discrete random variables

$$\sum_{i=1}^{n} p_i = 1$$

and the corresponding results for the continuous random variables is

$$\int_{-\infty}^{\infty} f(x)dx = 1$$

where x is a value of X, a continuous random variable with associated probability density function $f(x)$. The reader will note that the range of x does not necessarily extend from $-\infty$ to $+\infty$. For example, if X is a random variable indicating service time, the values of X are strictly greater than or equal to zero. The variance is a measure of the spread of a random variable.

A measure of significance in production line analysis is the coefficient of variation (c.v.) which gives the dimenionless value of the standard deviation of a random variable in terms of the value of its mean and is defined as follows:

$$c.v.(X) = \frac{\text{standard deviation}}{\text{mean}} = \frac{\sqrt{\sigma_X^2}}{EX}.$$

The c.v. of $X = T_1 + T_2$ is 0.345.

To illustrate the use of probability density functions in association with continuous random variables, assume that the service time of an operation is *uniformly* distributed between a minimum of 1 minute and a maximum of 3 minutes as illustrated in Figure A.1.

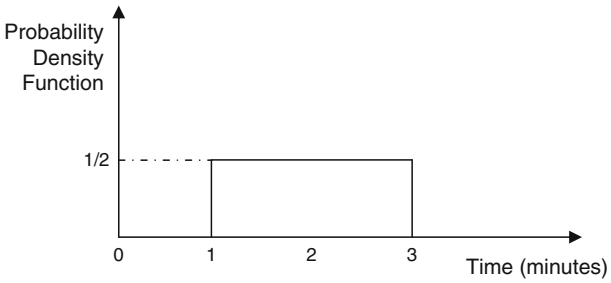


Fig. A.1. Continuous uniform distribution

Table A.2. Discrete probability distributions

Random variable name	Probability mass function	Mean	Variance	Range of variable	Parameter values
Bernoulli	$p^x(1-p)^{1-x}$	p	$p(1-p)$	$x=0,1$	$0<p<1$
Binomial	$\frac{n!}{x!(n-x)!}p^x(1-p)^{n-x}$	np	$np(1-p)$	$x=0,1,\ldots,n$	$0<p<1$ $n=1,2,\ldots$
Poisson	$e^{-\lambda}\,\frac{\lambda^x}{x!}$	λ	λ	$x=0,1,2,\ldots$	$\lambda>0$
Geometric	$p(1-p)^{x-1}$	$\frac{1}{p}$	$\frac{1-p}{p^2}$	$x=1,2,\ldots$	$0<p<1$

The probability density function (pdf), $f(t)$, is defined as follows

$$f(t)=\begin{cases}\frac{1}{2}, & \text{if } 1\le t\le 3;\\ 0. & \text{otherwise.}\end{cases}$$

Clearly, as expected

$$\int_1^3 f(t)dt=1.$$

The expected value of this random variable, T, is given by

$$ET=\int_1^3 tf(t)dt=\frac{1}{2}\int_1^3 tdt=2$$

and the variance of T is

$$VarT=\int_1^3 (t-2)^2 f(t)dt=\frac{1}{2}\int_1^3 (t-2)^2 dt=\frac{1}{3}.$$

The c.v.(T) is given by

$$c.v.(T)=\frac{\sqrt{\frac{1}{3}}}{2}=0.289$$

Next, a number of important probability distributions which are applicable to the analysis of production lines are considered. Table A.2 tabulates some of the discrete probability distributions and Table A.3 gives some of the continuous probability distributions.

A.2.1 Bernoulli trials

If a discrete random variable Y may only assume two values 0 or 1 with probabilities p and $q=1-p$, respectively, such a random variable is known as a *Bernoulli random variable* and each sample of the experiments is known as a Bernoulli trial.

Table A.3. Continuous probability distributions

Random variable name	Probability density function	Mean	Variance	Range of variable	Parameter values
Normal	$\frac{1}{\sigma(2\pi)^{1/2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	μ	σ^2	$-\infty < x < +\infty$	$-\infty < \mu < +\infty$ $\sigma > 0$
Exponential	$\lambda e^{-\lambda x}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$	$0 < x < \infty$	$\lambda > 0$
Erlang	$\frac{\lambda^\kappa}{(\kappa-1)!} x^{\kappa-1} e^{-\lambda x}$	$\frac{\kappa}{\lambda}$	$\frac{\kappa}{\lambda^2}$	$0 < x < \infty$	$\lambda > 0$ integer $\kappa > 0$
Gamma	$\frac{\lambda^n}{\Gamma(n)} x^{n-1} e^{-\lambda x}$	$\frac{n}{\lambda}$	$\frac{n}{\lambda^2}$	$0 < x < \infty$	$\lambda, n > 0$
Uniform	$\frac{1}{b-\alpha}$	$\frac{\alpha+b}{2}$	$\frac{(b-\alpha)^2}{12}$	$\alpha < x < b$	$-\infty < \alpha, b < +\infty$

Clearly, $EY = p$ and $VarY = p - p^2 = p(1-p) = pq$ and $c.v.^2(Y) = q/p$.

If there are n Bernoulli trials and X is the discrete random variable with values $x = 0, 1, \ldots, n$, then the probability mass function of X which is the sum of the values obtained over the n Bernoulli trials is given by:

$$p(x) = \binom{n}{x} p^x (1-p)^{n-x}, \quad x = 0, 1, \ldots, n$$

where

$$\binom{n}{x} = \frac{n!}{x!(n-x)!}.$$

This distribution is known as the *Binomial distribution*.

$$EX = np; \quad VarX = npq; \quad c.v.^2(X) = \frac{q}{np}.$$

Distributions (probability mass functions) of random variables related to the binomial distribution are the geometric distribution and the negative binomial distribution.

The *geometric distribution* is the distribution of the discrete random variable Y which is the number of the Bernoulli trials which occur up to and including the trial at which the random variable equals 1 for the first time. The mean and the variance of the geometric distribution are given in Table A.2 with associated squared coefficient of variation $c.v.^2(Y) = 1 - p = q$.

The binomial and geometric distributions are two-parameter distributions in that two parameters n and p are required to specify completely the distributions. A random variable R follows the negative binomial distribution, if R is the number of the Bernoulli trials which occur until the rth random variable equals 1 ($r \geq 1$). The reader

should note that the geometric distribution is a special case of the negative binomial distribution when $r = 1$. The negative binomial distribution is a three-parameter distribution. Details of the functional form, mean and variance are given in standard probability textbooks. For completeness, the reader might note that the hypergeometric distribution, a discrete distribution, often associated with sampling of lots of finite size, is a three-parameter distribution.

The *Poisson distribution* which may be derived analytically, under given hypotheses, has the following probability mass function:

$$p(x) = e^{-\lambda}\frac{\lambda^x}{x!}, \quad x = 0, 1, 2, \ldots$$

where X is a discrete random variable with values $x = 0, 1, 2, \ldots$ It is easy to show that $EX = \lambda$, $VarX = \lambda$ and $c.v.(X) = 1$. The Poisson distribution is often used to specify the distribution of arrivals in queueing systems. The expected number of arrivals λ is generally known as the arrival rate per unit time. In queueing theory if λ is the mean arrival rate, the following Poisson distribution as a function of time is generally used:

$$p(x,t) = e^{-\lambda t}\frac{(\lambda t)^x}{x!}, \quad x = 0, 1, 2, \ldots$$

where $p(x,t)$ is the probability of x arrivals in time t. In the derivation of the Poisson distribution, an assumption is made that the number of arrivals in time element δt is $\lambda\,\delta t$, λ a constant.

Perhaps the most important continuous probability density function (p.d.f.) is the *normal distribution*. This well-known distribution is used in a variety of applications in all disciplines. It is a two-parameter distribution with the mean and variance appearing explicitly in its functional form. The distribution has a number of very significant properties including, for example, that the sum (convolution) of two or more normally distributed random variables is itself normally distributed. Under certain very broad conditions, the sum of a large number of independent and arbitrarily distributed random variables may be shown to be approximately normally distributed (generalized central limit theorem). Further details about the normal distribution are available in most standard probability textbooks. An associated distribution of the normal distribution is the so-called lognormal distribution which is the distribution of a random variable whose natural logarithm follows a normal distribution. In contrast to the normal distribution, the *lognormal distribution* has applications where the product of nonnegative random variables is involved.

The *exponential distribution* of a continuous random variable T has the following functional form:

$$f(t) = \lambda e^{-\lambda t}, \quad t \geq 0.$$

The mean and variance of this distribution are, respectively:

$$ET = \int_0^\infty t f(t) dt = \frac{1}{\lambda}$$

$$Var\, T = ET^2 - (ET)^2 = \frac{2}{\lambda^2} - \left(\frac{1}{\lambda}\right)^2 = \frac{1}{\lambda^2}$$

and the coefficient of variation is

$$c.v.(T) = 1.$$

There is an interesting relationship between the discrete Poisson distribution and the continuous exponential distribution as shown below.

Given a Poisson process

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

let Ω be a random variable of the time between the events, then $f(\omega)\Delta\omega$, the probability of no event up to time ω and one event between ω and $\omega + \Delta\omega$, is given by

$$\begin{aligned} f(\omega)\Delta\omega &= e^{-\lambda\omega} \frac{(\lambda\omega)^0}{0!} \left[e^{-\lambda\Delta\omega} \frac{(\lambda\Delta\omega)^1}{1!} \right] \\ &= \lambda\Delta\omega\, e^{-\lambda\omega} \quad \text{to order } \Delta\omega. \end{aligned}$$

Therefore, $f(\omega) = \lambda e^{-\lambda\omega}, \omega \geq 0$, is the pdf of Ω. Thus, in a Poisson process the time between events follows an exponential distrribution with the same parameter λ. The converse is also true. To summarize, if arrivals into a process occur according to a Poisson distribution with arrival rate λ (mean value λ), the inter-arrival time is distributed according to an exponential distribution with mean value $1/\lambda$.

A.2.2 Memoryless property of the exponential distribution

Random variables distributed according to an exponential distribution exhibit an interesting property known as the "memoryless" property.

Given T, a continuous random variable distributed according to an exponential distribution, i.e., with a pdf

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

the cumulative distribution function (CDF), $F(t)$ of T is defined by

$$F(t) = Prob(T \leq t) = 1 - e^{-\lambda t}, \quad t \geq 0.$$

Assume $T > s$, a given value, determine the pdf of T given $T \geq s$.

Let A be the event $T \geq s$ and B be the event T being between t_1 and $t_1 + \Delta t_1$; $t_1 > s$. Then from a well-known result from the axioms of probability theory:

$$Prob(B|A) = \frac{P(B \cap A)}{P(A)}$$

where $\cap$ is the intersection operator and since event $B \cap A =$ event B,

$$Prob(B|A) = \frac{f(t_1)\,\Delta t_1}{e^{-\lambda s}} = \lambda\, e^{-\lambda (t_1 - s)}\,\Delta t_1, \quad t_1 \geq s.$$

In other words, if T is known to be greater than or equal to s, the pdf of $T|T \geq s$ is an exponential distribution, starting out at $t = s$, with parameter λ. Thus, if the time between successive occurrences of an event is exponentially distributed, then the time to the occurrence of the next event does not depend on how long ago the previous event occured. The time to the next occurrence of an event at any point in time under these circumstances is distributed according to an exponential distribution starting out at this point in time.

A.2.3 Relationship between the exponential distribution and the Poisson distribution

The reader may note the important relationship between the exponential distribution and the Poisson distribution: When the arrivals to a service system in a time interval $(0,t]$ follow the Poisson distribution with mean arrival rate λ units per unit time then the inter-arrival times between any two successive arrivals follow the exponential distribution with the same parameter, i.e., with mean inter-arrival time $1/\lambda$.

A convolution of κ independent exponentially distributed random variables is distributed according to an *Erlang distribution* with functional form as given in Table A.3. The squared coefficient of variation of an Erlang distribution with κ exponential phases is given by

$$c.v.^2(X) = \frac{\frac{\kappa}{\lambda^2}}{(\frac{\kappa}{\lambda})^2} = \frac{1}{\kappa}$$

so the c.v. of an Erlang distribution with $\kappa > 1$ is always less than 1.

A generalization of the Erlang distribution is the so-called *Gamma distribution* with its functional form given in Table A.3, where $\Gamma(n)$ is the gamma function given by

$$\Gamma(n) = \int_0^\infty x^{n-1}\, e^{-x}\, dx$$

where in general n is a positive real number. When n is integer, $\Gamma(n) = (n-1)!$ and in this case the gamma distribution reduces to the Erlang distribution. In general, it may be shown that $\Gamma(n) = (n-1)\Gamma(n-1)$ and $\Gamma(1) = 1$. Tables of the Gamma function are available over the interval $(0,1)$.

The $c.v.^2$ of the gamma distribution is $1/n$ and as $n > 0$, the c.v. of a gamma distribution may be any value greater than 0.

The reader will note that the Gamma and Erlang distributions like the normal and lognormal distributions are two-parameter distributions whereas the exponential distribution is a one-parameter distribution.

A.2.4 The Coxian distribution with two phases

The random variable Ω is said to be distributed according to the Coxian distribution with two phases, denoted by C_2, when it is equal to X with probability d_1 and equal to $X+Y$ with probability d_2, where X and Y are both random variables exponentially distributed with parameters μ_1 and μ_2, respectively, i.e.,

$$\Omega = \begin{cases} X, & \text{with probability } d_1 = 1-d \\ X+Y, & \text{with probability } d_2 = d. \end{cases}$$

where $d_1 + d_2 = 1$, with the d being the branching probability. Without loss of generality, it may be assumed that $\mu_1 > \mu_2$.

The probability density function of C_2 is obtained by considering the two ways in which the random variable Ω will be in the interval t to $t+\Delta t$:

$$f_\Omega(t) = d_1\mu_1 e^{-\mu_1 t} + d_2 \frac{\mu_1\mu_2}{\mu_2-\mu_1}(e^{-\mu_1 t} - e^{-\mu_2 t}), \quad t \geq 0$$

as the probability density function of the convolution of exponentially distributed random variables X and Y with parameters μ_1 and μ_2, respectively, is given by

$$f_{X+Y}(t) = \frac{\mu_1\mu_2}{\mu_2-\mu_1}\left(e^{-\mu_1 t} - e^{-\mu_2 t}\right), \quad t \geq 0.$$

Another more general form of this probability density function is

$$f_\Omega(t) = \beta_1\mu_1 e^{-\mu_1 t} + \beta_2\mu_2 e^{-\mu_2 t}, \quad t \geq 0$$

where the coefficients β_1 and β_2 are given by

$$\begin{aligned} \beta_1 &= 1 - d - \frac{d\mu_2}{\mu_1-\mu_2} = 1 - \frac{d\mu_1}{\mu_1-\mu_2}, \\ \beta_2 &= \frac{d\mu_1}{\mu_1-\mu_2} = 1 - \beta_1. \end{aligned}$$

The cumulative distribution function of C_2 is

$$\begin{aligned} F_\Omega(t) &= 1 - d_1 e^{-\mu_1 t} - \frac{d_2}{\mu_2-\mu_1}(\mu_2 e^{-\mu_1 t} - \mu_1 e^{-\mu_2 t}), \quad t \geq 0 \\ &= 1 - e^{-\mu_1 t} - \frac{d_2\mu_1}{\mu_2-\mu_1}(e^{-\mu_1 t} - e^{-\mu_2 t}), \quad t \geq 0 \end{aligned}$$

and in a more general form, this may be written as follows

$$F_\Omega(t) = 1 - \beta_1 e^{-\mu_1 t} - \beta_2 e^{-\mu_2 t}, \quad t \geq 0.$$

The variance of this distribution is

$$Var[\Omega] = \frac{1}{\mu_1^2} + \frac{d_2}{\mu_2^2} + \frac{d_1 d_2}{\mu_2^2}.$$

If $d_1 = 1$ or ($d_2 = 1$ and $\mu_1 = \mu_2$), the exponential or Erlang distribution with two phases of service, denoted by E_2, are obtained, respectively.

The first three moments, $\psi_1,\ \psi_2,\ \psi_3$, of the Coxian distribution with two phases of service, C_2, are given by (see Papadopoulos et al., 1993):

$$\psi_1 = E[\Omega] = \frac{1}{\mu_1} + \frac{d_2}{\mu_2} \tag{A.1}$$

$$\psi_2 = \frac{2}{\mu_1^2} + \frac{2d_2(\mu_1 + \mu_2)}{\mu_1 \mu_2^2} \tag{A.2}$$

$$\psi_3 = \frac{6}{\mu_1^3} + \frac{6d_2(\mu_1^2 + \mu_1\mu_2 + \mu_2^2)}{\mu_1^2 \mu_2^3}. \tag{A.3}$$

The squared coefficient of variation is

$$c.v.^2(\Omega) = \frac{\mu_2^2 + d_2(1 + d_1)\mu_1^2}{(\mu_2 + d_2\mu_1)^2}.$$

The C_2 distribution may be used to approximate any general distribution by matching their first three moments. For a general distribution with its first three momentsknown, $\psi_1,\ \psi_2$ and ψ_3, the Coxian-2 parameters may be derived from equations A.1, A.2 and A.3. If $\mu_1 + \mu_2 = \nu$ and $\mu_1\mu_2 = \xi$, then:

$$\nu = \frac{6\psi_1\psi_2 - 2\psi_3}{3\psi_2^2 - 2\psi_1\psi_3} \tag{A.4}$$

$$\xi = \frac{12\psi_1^2 - 6\psi_2}{3\psi_2^2 - 2\psi_1\psi_3} \tag{A.5}$$

$$\mu_1 = \frac{1}{2}\left[\nu + \sqrt{\nu^2 - 4\xi}\right], \quad \mu_2 = \frac{1}{2}\left[\nu - \sqrt{\nu^2 - 4\xi}\right] \tag{A.6}$$

$$d_2 = \mu_1^{-1}\mu_2(\psi_1\mu_1 - 1), \quad \mu_1 > 0. \tag{A.7}$$

For the Coxian-2 parameters to have real values, the coefficient of variation of the general distribution must be greater than or equal to 1 and also the condition $\psi_3/\psi_1^3 > 1.5(c.v.^2 + 1)^2$ must be satisfied. Although the C_2 distribution has three independent parameters, it is ideal for a two-moment approximation of general distributions with a squared coefficient of variation greater than or equal to 0.5.

When $0.5 \leq c.v.^2 \leq 1$, the parameters of the C_2 distribution may be shown to be

$$\mu_1 = \frac{1}{\psi_1 c.v.^2}, \quad \mu_2 = \frac{2}{\psi_1}, \quad d_2 = 2(1 - c.v.^2). \tag{A.8}$$

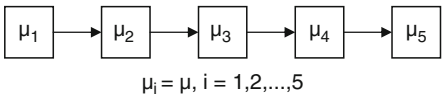


Fig. A.2. Five-stage Erlang distribution, $E_{k=5}$

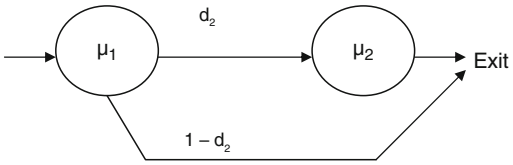
A.2.5 Phase-type distributions

The virtues of having the Markovian property in queueing problems is well known. In many cases it may be reasonable to assume Poisson arrivals but service time distributions may not be exponential. To overcome this latter difficulty, there is a rich literature related to the so-called phase method. The Erlang distribution discussed above may be considered as a convolution of k identical exponential distributions and may be depicted as in Figure A.2.

The individual stages themselves may have no physical interpretation and the concept is that the customer leaves the system having passed through all k stages. Considerable development of the phase method approach is possible, for example, instead of having identical exponential distributions of time spent in each stage it would be possible to have a convolution of non-identical distributions in series (sometimes called the generalized Erlang distribution). A further simple extension would be to have exponential distributions in parallel with different means and service time of a particular customer chosen at random. Such an arrangement would lead to a hyper-exponential distribution with k parallel channels. Early work with respect to the phase method stressed the advantage of being able to model distributions with squared coefficients of variation other than 1 (which is the pure single exponential case). For example, the coefficient of variation of Erlang is between 0 and 1((0,1]) and for the hyper-exponential distribution the squared coefficient of variation is greater than 1 provided that all the μ_i's are not equal. However, modeling distributions just to match the squared coefficient of variation is a relatively weak match as it is possible to have many different shapes of distributions with the same squared coefficient of variation. This observation has led to further developments of the phase method where, for example, one could consider (i) a convolution of identical or non-identical Erlang distributions in series format or (ii) a parallel series arrangement of identical or non-identical exponential distributions. A particular phase type distribution is the Coxian distribution with two phases of service, already described above, and it may be of interest to the reader to consider its derivation via the use of transition probability matrices in contrast to the direct derivation used or at least implied above.

Consider the following system as depicted in Figure A.3:

A unit enters the system at the first station. After completion of service at this station, the unit may either exit the system with probability $1 - d_2$ or enter the second station with probability d_2, $0 \leq d_2 \leq 1$. Both stations have exponentially distributed service times with parameters μ_1, μ_2, respectively. Arrivals to the system are assumed to follow a Poisson distribution with mean arrival rate λ.



There are 3 states as follows:

State 1: being served in station 1
State 2: being served in station 2
State 3: Exit from the system (absorbing state)

Fig. A.3. A two-phase Coxian distribution, C_2

The transition probability matrix, $T(\Delta t)$, from states at time t to states at time $t+\Delta t$ (to the order Δt) is given below:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1-\mu_1\Delta t & d_2\mu_1\Delta t & (1-d_2)\mu_1\Delta t \\ 0 & 1-\mu_2\Delta t & \mu_2\Delta t \\ 0 & 0 & 1 \end{pmatrix} \end{array}.$$

Note that state 3 is an absorbing state.

Let $f_\Omega(t)dt = P[t \le t \le t+dt]$ and t is the time to exit from the system. If $\Pi(t)$ is the probability vector giving the probabilities of being in each of the non-absorbing states at time t (states 1 and 2, above), then it holds, in general:

$$\frac{d}{dt}\Pi(t) = \Pi(t)[\Theta]$$

where

$$\Theta = \begin{bmatrix} -\mu_1 & d_2\mu_1 \\ 0 & -\mu_2 \end{bmatrix}$$

derived from $T(\Delta t)$, above. $T(\Delta t)$ is a 3×3 matrix but here only a 2×2 matrix is needed. $[\Theta]$ is obtained from $T(\Delta t)$ by deleting the third row and third column (corresponding to the absorbing state 3), subtracting 1 from the diagonal elements and deleting the Δt factors from all terms.

Taking Laplace transforms:

$$s\Pi(s) - \Pi(0) = \Pi(s)[\Theta]$$

where Π_0 is a row vector giving the probabilities of being in each non-absorbing state (1 or 2, above) at time $t=0$. Assume $\Pi(0) = (1,0)$, therefore

$$\Pi(s)[sI - [\Theta]] = \Pi(0)$$

$$\Pi(s) = \Pi(0)[sI - [\Theta]]^{-1}$$

Now $f_\Omega(t)dt = \Pi(t)\{A\}dt$, where $\{A\}$ is a column vector giving the probability of being absorbed from each nonabsorbing state and

$$\{A\} = \left\{ \begin{array}{c} (1-d_2)\mu_1 \\ \mu_2 \end{array} \right\}$$

obtained from $T(\Delta t)$, above.

$$\begin{aligned} F(s) &= \Pi(s)\{A\} \\ &= \Pi(0)[sI - [\Theta]]^{-1}\{A\} \end{aligned}$$

where $F(s)$ is the Laplace transform of $f(t)$. Then,

$$f(t) = \Pi(0)[e^{\Theta t}]\{A\}, \quad t \geq 0.$$

Let $e^{\Theta t} = [G(t)]$, then

$$\frac{dG(t)}{dt} = [G(t)][\Theta].$$

Taking Laplace transforms

$$s[G(s)] - I = [G(s)][\Theta]$$

$$\begin{aligned} [G(s)] &= [sI - \Theta]^{-1} \\ &= \left[\begin{array}{cc} \frac{1}{s+\mu_1} & \frac{d_2\mu_1}{(s+\mu_1)(s+\mu_2)} \\ 0 & \frac{1}{s+\mu_2} \end{array} \right]. \end{aligned}$$

By inversion

$$[G(t)] = \left[\begin{array}{cc} e^{-\mu_1 t} & -\frac{d_2\mu_1}{(\mu_1-\mu_2)}e^{-\mu_1 t} + \frac{d_2\mu_1}{(\mu_1-\mu_2)}e^{-\mu_2 t} \\ 0 & e^{-\mu_2 t} \end{array} \right] = e^{[\Theta]t}.$$

Note that

$$\frac{d_2\mu_1}{(s+\mu_1)(s+\mu_2)} = -\frac{d_2\mu_1}{(s+\mu_1)(\mu_1-\mu_2)} + \frac{d_2\mu_1}{(s+\mu_2)(\mu_1-\mu_2)}.$$

Then, given $\Pi(0) = (1,0)$,

$$f(t) = \{10\}[e^{[\Theta]t}] \left\{ \begin{array}{c} (1-d_2)\mu_1 \\ \mu_2 \end{array} \right\}$$

which on substitution reduces to

$$f(t) = \left[1 - \frac{d_2\mu_1}{\mu_1-\mu_2}\right]\mu_1 e^{-\mu_1 t} + \frac{d_2\mu_1}{\mu_1-\mu_2}\mu_2 e^{-\mu_2 t}, \quad t \geq 0,$$

which may easily be converted to the expression for $f_\Omega(t)$, given above.

A.3 Discrete Markov Processes (Markov Chains)

Usually, production machines may be described as residing in a discrete and identifiable state, e.g., operating satisfactorily (up state) or in repair (down state). The analysis of the behavior of systems of such machines involves the determination of the state of the system (each machine) at selected times (technically known as stages). These stages or times may be represented by either discrete or continuous variables. If discrete, the actual time between each stage may be regular or irregular. If the states and stages of the system are both discrete and the future states of the system are characterized by a lack of memory, i.e., the state of the system at stage $t + 1$ depends only on the state of the system at stage t and not on the history of the system up to stage t, the underlying process is described as a discrete Markov process or equivalently as a Markov chain.

Although, in general, it is satisfactory to describe the states of machines as discrete and distinct, it may be desirable to consider the time variable as either discrete or continuous. There is a rich literature on discrete state Markov processes with discrete or continuous time stages. Here, by way of an example a simple discrete state, discrete stage Markov process (Markov chain) is considered:

A machine as shown in Figure A.4 may be in one of three states, viz., operating satisfactorily, state 1; operating in a derated condition, state 2; and broken down, state 3. Assume the stages of the system are discrete and regular.

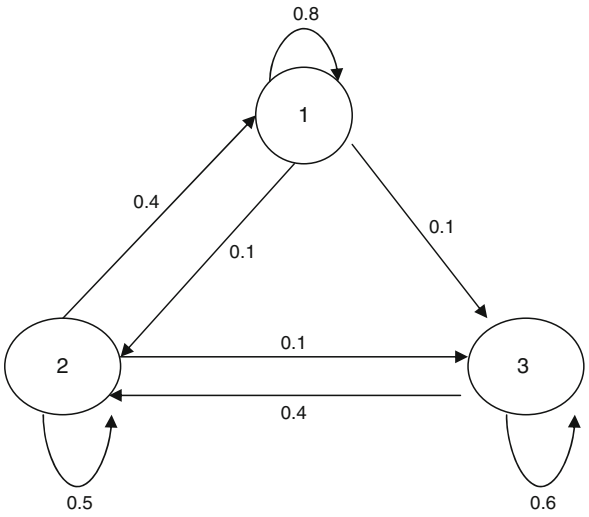


Fig. A.4. State space diagram of a three-state machine system

The system may be described by the following transition probability matrix T:

$$T = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.4 & 0.5 & 0.1 \\ 0 & 0.4 & 0.6 \end{pmatrix} \end{array} = [p_{ij}].$$

Note that the sum of the elements in each row of T add to 1 and all elements are probability masses, where $0 \leq p_{ij} \leq 1, \;\; i, j = 1, 2, 3$.

If the system is in state 2 at stage t, for example, it will be in state 1 at stage $t+1$ with probability 0.4, it will remain in state 2 with probability 0.5, while it will move to state 3 with probability 0.1.

An interesting question is to track the behavior of the machine system over time. For example, if the system starts out in state 2, where will it be after three stages (periods)? Answering such questions involves matrix multiplication as follows:

Initial state of system:

$$\{P\} = \{P_1^0, P_2^0, P_3^0\}$$

as a probability vector. After one stage, the state of the system is:

$$\{P^1\} = \{P^0\}\, T.$$

After n stages, the state of the system is

$$\{P^n\} = \{P^0\}\, [T]^n.$$

So, for example, the state of the system which if it starts out in state 2 initially will be as follows after three stages:

$$\{P^0\} = \{0, 1, 0\}$$

$$[T]^3 = \begin{bmatrix} 0.612 & 0.213 & 0.175 \\ 0.548 & 0.277 & 0.175 \\ 0.304 & 0.396 & 0.300 \end{bmatrix}.$$

The state of the system at stage 3 will be:

$$\{P^3\} = \{P^0\}\, [T]^3 = \{0.548, 0.277, 0.175\}.$$

Note $\{P^3\}$ is a probability vector.

Some transition probability matrices have the following property:

$$\lim_{n \to \infty} \{P^n\} = \{P\} \quad \text{exists},$$

such that

$$\{P\} = \{P\}\, [T],$$

where $\{P\}$ is termed the steady state probability vector.

Continuing the example:

$$\{P\} = \{P_1, P_2, P_3\}$$

and $\{P\} = \{P\}[T]$ leads to the following three simultaneous, but not independent, linear equations:

$$\begin{aligned} P_1 &= 0.8P_1 + 0.4P_2 \\ P_2 &= 0.1P_1 + 0.5P_2 + 0.4P_3 \\ P_3 &= 0.1P_1 + 0.1P_2 + 0.6P_3. \end{aligned}$$

To obtain the values of P_1, P_2 and P_3 it is necessary to include the normalizing condition:

$$P_1 + P_2 + P_3 = 1.$$

The solution to the four equations, above, involving P_1, P_2 and P_3 is

$$\{P\} = \{P_1, P_2, P_3\} = \left\{ \frac{8}{15}, \frac{4}{15}, \frac{3}{15} \right\}.$$

An interpretation of this result is that in steady state or in the long term, in every 15 stages the machine is operating satisfactorily in 8 of these stages, is operating in the derated state in 4 of these stages, while it is broken down in 3 of these stages.

In Markov chains, a state is said to be an absorbing state if it has the property that once the system enters the absorbing state it remains in that state for all subsequent stages of the system. The concept of the absorbing state is very useful in determining the mean number of periods a system remains outside a particular state.

To illustrate the use of the concept of absorbing states, consider the above example machine system with a view to determining the mean (expected) number of stages to the first breakdown of the system (state 3) starting out in any of the other two states (state 1 or state 2, now called "transient" states).

Define $[T^*]$, obtained from $[T]$ as follows:

$$T^* = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.4 & 0.5 & 0.1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Note that in T^*, state 3 is an absorbing state and that T^* may be partitioned as follows:

$$T^* = \left[\begin{array}{c|c} A & B \\ \hline 0 & I \end{array}\right]$$

where $A, B, 0$ and I have dimensions 2×2, 2×1, 1×2 and 1×1, respectively. The matrix 0 has all elements equal to 0 while I is an identity matrix. This general structure of T^* will hold even if there are more than one absorbing states and the dimension of T^* is greater than 3×3.

In the following analysis a matrix F, the fundamental matrix, plays a central role:

$$F = [I - A]^{-1}.$$

A as noted above represents the transient states prior to being absorbed.

$$F = I + A + A^2 + \cdots$$

gives the expected number of stages the system stays in each transient (non-absorbing) state before the system reaches the absorbing state(s). This may be shown from a consideration of the structure of $[T^*], [T^*]^2, [T^*]^3, \ldots, [T^*]^n$ as follows:

$$\begin{aligned}
[T^*] &= \left[\begin{array}{c|c} A & B \\ \hline 0 & I \end{array}\right] \\
[T^*]^2 &= \left[\begin{array}{c|c} [A]^2 & AB + B \\ \hline 0 & I \end{array}\right] \\
[T^*]^3 &= \left[\begin{array}{c|c} [A]^3 & A^2B + AB + B \\ \hline 0 & I \end{array}\right] \\
&\vdots \\
[T^*]^n &= \left[\begin{array}{c|c} [A]^n & [A^{n-1} + \cdots + A + I]\, B \\ \hline 0 & I \end{array}\right].
\end{aligned}$$

In the above example

$$F = [I - A]^{-1} = \begin{bmatrix} 8.33 & 1.67 \\ 6.67 & 3.33 \end{bmatrix}.$$

So, starting out in state 1, the system would on average spend 8.33 stages in state 1 and 1.67 stages in state 2 or a total of 10.0 stages before failure for the first time. Thus, the mean time to failure (complete breakdown) of the system starting out in state 1 is 10 stages.

The above result could be obtained in a more efficient manner as follows:

Let EP_i, $i = 1, 2$ be the expected number of stages before reaching state 3 having set out in state i, $i = 1, 2$, respectively,

$$\begin{aligned}
EP_1 &= 1 + 0.8EP_1 + 0.1EP_2 \\
EP_2 &= 1 + 0.4EP_1 + 0.5EP_2
\end{aligned}$$

from which $EP_1 = 10$; $EP_2 = 10$.

A.4 Data Plotting

In practice, it may be necessary to assess if a particular data set comes from a selected underlying distribution. For example, the inter-arrival times of a process have been obtained from an actual observation of the process. The issue is how one would test

if it was reasonable to assume that the underlying distribution would be, for example, exponential. Basically, there are two different approaches. One is to plot the data on appropriate probability paper and assess usually by eye if the data is close enough to be on a straight line. Probability paper exists for a number of distributions including normal, exponential and gamma. An issue arises in plotting experimental data as to how to transfer the data on to the plotting coordinates. There are a number of different procedures in existence and the reader is referred to Shapiro (1980) and Montgomery (1996), among others. One of the benefits of using probability plots is that the parameters of the distribution in question may be read off the plot or derived by simple calculations. The other approach is to use statistical methods such as the chi-squared or the Kolmogorov/Smirnov goodness of fit tests. In goodness of fit tests the null hypothesis is that the candidate distribution is correct. Hence, there is a strong bias in favor of whatever distribution is chosen. For this reason the modeler testing physical data should have in mind appropriate distributions to test. In some work, it might be more appropriate to use the data and to estimate moments of the underlying distribution and from these moments to develop appropriate parameters for selected distributions as discussed earlier in relation to the Coxian distribution.

A.5 Well-Known Results of Queueing Theory

Here important results for single-station queueing systems are given.

Kendall's notation for single station queues, $A/B/C : D/E/F$, where A is a descriptor of the statistics of the arrival into the single-station system, B is a descriptor of the service time distribution of each of the servers in the system, C refers to the number of servers, D is a descriptor of the queueing discipline, i.e., how the arriving units are called into the service, E is a specification of the overall size of the system which is a limit to the number of those waiting for service plus those being served and F is a descriptor of the population from which the arriving units come, specifies fully any single-station queueing system. In queueing theory the terms units and customers are used interchangeably and there is a basic assumption that customers are not allowed to wait if there is a free server available. The queueing discipline affects the waiting time of classes of customers arriving into the system. There are two types of solutions to queueing systems, one called time-dependent and the other steady-state. The steady-state solution is a limit as time goes to infinity of the time-dependent solution, which of course is the full solution. Many of the analytical results of queueing theory are derived on the assumption that the arrival process is a Poisson process and that the distributions of service times of servers are independent identically distributed (i.i.d.) exponential distribution. Because of the relationship between the Poisson and exponential distributions and the well-known Markov process theory, it is normal to let $A = M$ and $B = M$ if the arrival process is Poisson and the service time distribution is exponential.

Table A.4 illustrates the types of characteristics of a single-station queueing system that have been developed analytically.

Table A.4. Characteristics of single-station queueing systems

Description	*Notation*
Probability that there are n units in the system at time t	$P_n(t)$
Probability that there are n units in the system in steady state	P_n
The expected number of units in the system in steady state	L_S
The expected number of units waiting for service in steady state	L_q
The expected time spent in the system	W_S
The expected time spent waiting for service	W_q
Probability density function (pdf) of the total time spent in the system under a given queueing discipline	$f_{W_S}(t)$
pdf of the time spent waiting for service under a given queueing discipline	$f_{W_q}(t)$

A.5.1 $M/M/1$: First-Come First-Served (FCFS)$/\infty/\infty$ queue

$$P_n(t) = e^{-(\lambda+\mu)t}\left[\rho^{(n-i)/2} I_{n-i}(2\sqrt{\lambda\mu}t) + \rho^{(n-i-1)/2} I_{n+i+1}(2\sqrt{\lambda\mu}t) + (1-\rho)\rho^n \sum_{j=n+i+2}^{\infty} \rho^{-j/2} I_j(2\sqrt{\lambda\mu}t)\right], \quad (n \geq 1),$$

where $\rho = \lambda/\mu$, λ is the mean arrival rate and μ is the mean service rate, and $I_k(x)$ is the modified Bessel function of first kind and k^{th} order.

For $\rho < 1$:

$$\begin{aligned}
P_n &= (1-\rho)\rho^n \\
L_S &= \frac{\rho}{1-\rho} \\
L_q &= \frac{\rho^2}{1-\rho} \\
W_q &= \frac{\rho/\mu}{1-\rho} \\
W_S &= \frac{1/\mu}{1-\rho} \\
f_{W_S}(t) &= \mu(1-\rho)e^{-\mu(1-\rho)t}, \quad t \geq 0 \\
f_{W_q}(t) &= (1-\rho)\delta(t) + \lambda(1-\rho)e^{-\mu(1-\rho)t}, \quad t \geq 0
\end{aligned}$$

where $\delta(t)$ is the unit impulse function occurring at time $t = 0$ called the **Dirac delta function**.

A.5.2 $M/M/1$: FCFS/N/∞ queue

$$
\begin{aligned}
P_n &= \left\{ \begin{array}{ll} \frac{1}{N+1}, & \rho = 1 \\ \frac{1-\rho}{1-\rho^{N+1}}\rho^n, & \rho \neq 1 \end{array} \right\}, \quad 0 \leq n \leq N, \quad \rho = \frac{\lambda}{\mu}, \\
L_S &= \frac{\rho\left[1-(N+1)\rho^N + N\rho^{N+1}\right]}{(1-\rho^{N+1})(1-\rho)}, \quad \rho \neq 1, \\
L_S &= \frac{N}{2}, \quad \rho = 1, \\
L_q &= L_S - \frac{\rho\,(1-\rho^N)}{1-\rho^{N+1}}, \quad \rho \neq 1, \\
L_q &= L_S - \frac{N}{N+1} = \frac{N}{2}\frac{N-1}{N+1}, \quad \rho = 1, \\
W_S &= L_S \frac{1}{\lambda\,(1-P_N)}, \\
W_q &= L_q \frac{1}{\lambda\,(1-P_N)}.
\end{aligned}
$$

A.5.3 $M/M/c$: FCFS/∞/∞ queue

$$
P_n = \left\{ \begin{array}{l} \frac{(c\rho)^n}{n!}P_0, \; n \leq c \\ \frac{c^c\,\rho^n}{c!}P_0, \; n \geq c, \end{array} \right.
$$

where $\rho = \lambda/c\mu < 1$ is the utilization factor and

$$
\begin{aligned}
P_0 &= \left\{ \sum_{j=0}^{c-1} \frac{(\rho c)^j}{j!} + \frac{(\rho c)^c}{c!}\left(\frac{1}{1-\rho}\right) \right\}^{-1}. \\
L_q &= \frac{(\rho c)^c}{c!}\frac{\rho}{(1-\rho)^2}P_0, \\
L_S &= L_q + \rho c, \\
f_{W_q}(t) &= \left\{ 1 - \frac{(\rho c)^c}{c!(1-\rho)} \right\}\delta(t) + \\
&\quad + \frac{(c\mu-\lambda)(\rho c)^c}{c!(1-\rho)}P_0 e^{-(c\mu-\lambda)t}, \quad t \geq 0, \\
W_q &= \frac{(\rho c)^c}{c!c\mu(1-\rho)^2}P_0,
\end{aligned}
$$

$$f_{W_S}(t) = \frac{1}{\lambda-(c-1)\mu}\left[\mu e^{-\mu t}(\lambda - c\mu + \mu A) - (1-A)(\lambda - c\mu)\mu e^{-(c\mu-\lambda)t}\right], \quad t \geq 0,$$

$$A = 1 - \frac{(\rho c)^c}{c!(1-\rho)}P_0,$$

$$W_S = W_q + \frac{1}{\mu}.$$

A.5.4 $M/M/c$: FCFS/N/∞ queue

$$P_n = \begin{cases} \frac{(c\rho)^n}{n!}P_0, & 1 \leq n \leq c-1 \\ \frac{c^c(\rho)^n}{c!}P_0, & c \leq n \leq N, \end{cases}$$

where $\rho = \lambda/c\mu < 1$ and

$$P_0 = \left\{\sum_{j=0}^{c-1}\frac{(\rho c)^j}{j!} + \frac{c^c}{c!}\sum_{j=c}^{N}\rho^j\right\}^{-1}.$$

$$L_q = \begin{cases} \frac{(\rho c)^c}{c!}\frac{\rho}{(1-\rho)^2}P_0\left\{1-\rho^{N-c+1}-(1-\rho)(N-c+1)\rho^{N-c}\right\}, & \rho \neq 1; \\ \frac{c^c}{c!}P_0\frac{(N-c)(N-c+1)}{2}, & \rho = 1, \end{cases}$$

$$\begin{aligned} L_S &= L_q + \rho c(1-p_N), \\ \rho(1-P_N) &= \text{utilization factor}, \\ \lambda_e &= 1-P_N \ \text{(the mean effective arrival rate)} \\ W_q &= L_q\frac{1}{\lambda\,(1-P_N)}, \\ W_S &= L_S\frac{1}{\lambda\,(1-P_N)}. \end{aligned}$$

A.5.5 $M/M/c$: FCFS/c/∞ queue

$$P_n = \frac{\frac{(\rho c)^n}{n!}}{\sum_{i=0}^{c}\frac{(\rho c)^i}{i!}}, \quad \rho = \lambda/c\mu, \quad n = 0,1,\ldots,c,$$

$$P_c = \frac{\frac{(\rho c)^c}{c!}}{\sum_{i=0}^{c}\frac{(\rho c)^i}{i!}}, \quad n = c,$$

which is the well-known *Erlang's loss formula* and corresponds to the probability of a full system in the steady state.

A.5.6 $M/M/\infty$ queue

This is the self-service queueing system where the number of servers is equal to the number of units in the system.

$$\begin{aligned} P_n &= \frac{1}{n!}\rho^n e^{-\rho}, \quad \rho = \frac{\lambda}{\mu} \\ L_S &= \rho \\ L_q &= 0 \\ W_q &= 0 \\ W_S &= \frac{1}{\mu}. \end{aligned}$$

The formulae above hold for the $M/G/\infty$ queue, where G is a general service time distribution too, except of course the last one, which applies to the exponential service time distribution.

A.5.7 $M/M/c$: FCFS$/K/K$–The finite source queue

In this model, the population from which the arrivals come is finite, say of size K. Classically known as *the machine interference problem*, it is concerned with the modeling of the repair of a bank of machines of size K with $c \leq K$ repair persons

$$P_n = \begin{cases} \binom{K}{n} (\frac{\lambda}{\mu})^n P_0, & 0 \leq n < c \\ \binom{K}{n} \frac{n!c^c}{c!} (\frac{\lambda}{c\mu})^n P_0, & c \leq n \leq K, \end{cases}$$

where,

$$P_0 = \left[\sum_{\nu=0}^{c-1} \binom{K}{\nu} \left(\frac{\lambda}{\mu}\right)^\nu + \sum_{\nu=c}^{K} \binom{K}{\nu} \frac{\nu! c^c}{c!} \left(\frac{\lambda}{c\mu}\right)^\nu \right]^{-1}$$

and

$$\binom{K}{\nu} = \frac{K!}{\nu!(K-\nu)!}. \tag{A.9}$$

Using the following notation:

- L_S equals the average number of machines 'down' in the system,
- L_q equals the average number of machines 'down' that are waiting in the queue to be repaired,
- $L_{d,r}$ equals the average number of machines 'down' that are under repair, and
- L_o equals the average number of machines that are operating,

then these measures, all relating to steady-state conditions, are given by the following expressions:

$$L_S = K - \frac{\mu}{\lambda}\left[c - \sum_{v=0}^{c-1}(c-v)P_v\right]$$

$$L_q = K - (1+\frac{\mu}{\lambda})\left[c - \sum_{v=0}^{c-1}(c-v)P_v\right]$$

$$L_{d,r} = L_S - L_q = c - \sum_{v=0}^{c-1}(c-v)P_v$$

$$L_o = K - L_S = \frac{\mu}{\lambda}\left[c - \sum_{v=0}^{c-1}(c-v)P_v\right]$$

$$\frac{\lambda(K-L_S)}{c\mu} = \frac{\lambda L_o}{c\mu} = \text{utilization factor.}$$

The total service capacity of the system is $c\mu$ and the mean effective arrival rate is $\lambda L_o = \lambda(K - L_S)$. Little's formula may be used to determine the average time spent in the system, W_S, and the average time spent waiting for repair, W_q.

Little's formulae, $L_S = \lambda_e W_S$ and $L_q = \lambda_e W_q$, where λ_e is the mean effective arrival rate into the system, which differs from λ if there are constraints preventing an arrival unit entering the system, e.g., if there is a maximum size N of the system. This formula may be shown to apply to most queueing systems that are organized so that a server is never idle if there is a unit waiting for service. Generally speaking, Little's formulae may be used in the analysis of manufacturing systems.

Analytical results are also available for queues where the arrivals and/or service times do not follow a Poisson distribution and an exponential distribution, respectively. The reader is referred to the specialists' textbooks on queueing theory for further information.

A.5.8 Queueing networks

A queueing network is a network of service stations each of which has at least one server and with storage capacity of finite size greater than or equal to zero between the service stations. This storage capacity is generally referred to as inter-station buffer capacity and its function is to allow queues to form before the associated service stations. Units in general may enter the queueing system at any particular station, if necessary wait for service, leave that station after service and go through the network along a route which may not be the same for other arriving units. It is possible for units to return to stations at which they were previously served, to leave the network at some point or to remain in the network indefinitely. Fundamental work on queueing networks was undertaken by Erlang and Jackson.

In queueing networks with K stations there are two fundamental concepts, namely, the probability of arriving from outside the network to station i, $i = 1$,

$2,\ldots,K$, and the probability that a unit which has completed its service at station i will go immediately to station j. The former probability, in the earlier work was assumed to follow a Poisson process with mean rate $\lambda_{i,e}$, $i = 1,2,\ldots,K$, and the latter, the routing probability, is given by $q_{i,j}$, $i,j = 0,\ldots,K$, where K is the total number of stations in the network and i or $j = 0$ indicates arrival from outside or departure from the system, respectively. Open networks are those networks which have contact with the outside, i.e., $\lambda_{i,e} \neq 0$ for all i. Closed networks, on the other hand, have no contact with the outside and so $\lambda_{i,e} = 0$ for all i and $q_{i,0} = 0$ for all i, i.e., no units are allowed to enter from outside the system or to leave the system. A cyclical queue is a closed queueing network in which the units follow a path from station i through the network and back to station i and they repeat this route ad infinitum. Series or tandem queues are open networks with arrivals only at the first station, processing of all items in a specific order and exit from the system from the last station with some probability to return to the first station via a process called feedback. Once at the first station, the unit goes through all stations in turn again.

Jackson devised results for open queueing networks where there is in effect no constraint in relation to the flow of units arising out of a shortage of inter-station buffer space. A sufficient condition for such a netwok would be that there is an infinite buffer capacity between stations. For such open Jackson networks, Jackson derived well-known product-form solutions for the joint probabilities. In relation to closed Jackson networks, Gordon and Newell derived a formula for the joint probabilities and Buzen provided an efficient way of determining the normalization coefficient.

Example 1

A two-station series model with capacity of the intermediate buffer equal to 1.

Consider the system depicted in Figure A.5. Items arrive at the first station according to a Poisson distribution with mean arrival rate equal to λ units per unit time. Items get service first at the first station and then move on to the second station via the intermediate buffer. The general rule is that an item is served if a server is free to give service. It should be noted that station 2 is never blocked and station 1 can be blocked when an item has finished its service at the first station, the intermediate

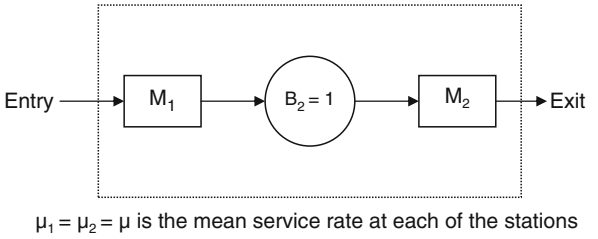


Fig. A.5. A two-station series queueing network with two identical exponential stations and an intermediate buffer of capacity 1

Table A.5. The transition matrix of the queueing network model of example 1

	(0,0,0)	(0,0,1)	(0,1,1)	(*b*,1,1)	(1,0,0)	(1,0,1)	(1,1,1)
(0,0,0)	$1-\lambda\Delta t$	0	0	0	$\lambda\Delta t$	0	0
(0,0,1)	$\mu\Delta t$	$1-\mu\Delta t-\lambda\Delta t$	0	0	0	$\lambda\Delta t$	0
(0,1,1)	0	$\mu\Delta t$	$1-\mu\Delta t-\lambda\Delta t$	0	0	0	$\lambda\Delta t$
(*b*,1,1)	0	0	$\mu\Delta t$	$1-\mu\Delta t$	0	0	0
(1,0,0)	0	$\mu\Delta t$	0	0	$1-\mu\Delta t$	0	0
(1,0,1)	0	0	$\mu\Delta t$	0	$\mu\Delta t$	$1-2\mu\Delta t$	0
(1,1,1)	0	0	0	$\mu\Delta t$	0	$\mu\Delta t$	$1-2\mu\Delta t$

buffer is full and the second station is busy. Both stations are assumed to be perfectly reliable and service times at the stations are identical and exponentially distributed, i.e., the average service rates are the same, viz., $\mu_1 = \mu_2 = \mu$. There is no waiting space in front of the first station and so, during the period the first station is blocked, all incoming items are lost to the system.

The possible states of this system are labeled (a,B,c), where a represents the state of the first station, B represents the state of the buffer and c represents the state of the second station. a may have three values: 0, 1 and b, where 0 indicates that the station is free, 1 indicates that the station is busy and b indicates that the station is blocked. Likewise, c can take two values: 0 and 1, whereas B may take two values, indicating the number of units in the buffer, i.e., 0 or 1. There are 7 feasible states as shown in the transition matrix given in Table A.5. The transition probabilities are obtained noting that a service will be completed in time interval $(t,t+\Delta t)$ with probability $\mu\Delta t$ and an arrival will enter the system if station 1 is not blocked in time $(t,t+\Delta t)$ with probability $\lambda\Delta t$. Using the usual assumptions of arrival and service completion, the transition matrix given in Table A.5 from states at time t to states at time $t+\Delta t$ (to the order Δt) may be determined.

The steady-state probabilities of the 7 states of the system may be derived from the solution of a system of 7 linear simultaneous equations, the following:

$$
\begin{aligned}
P_{000}(t+\Delta t) &= P_{000}(t)[1-\lambda\Delta t] + P_{001}(t)\mu\Delta t \\
P_{001}(t+\Delta t) &= P_{001}(t)[1-(\mu+\lambda)\Delta t] + P_{011}(t)\mu\Delta t + P_{100}(t)\mu\Delta t \\
P_{011}(t+\Delta t) &= P_{011}(t)[1-(\mu+\lambda)\Delta t] + P_{b11}(t)\mu\Delta t + P_{101}(t)\mu\Delta t \\
P_{b11}(t+\Delta t) &= P_{b11}(t)[1-\mu\Delta t] + P_{111}(t)\mu\Delta t \\
P_{100}(t+\Delta t) &= P_{000}(t)\lambda\Delta t + P_{100}(t)[1-\mu\Delta t] + P_{101}(t)\mu\Delta t \\
P_{101}(t+\Delta t) &= P_{001}(t)\lambda\Delta t + P_{101}(t)[1-2\mu\Delta t] + P_{111}(t)\mu\Delta t \\
P_{111}(t+\Delta t) &= P_{011}(t)\lambda\Delta t + P_{111}(t)[1-2\mu\Delta t].
\end{aligned}
$$

These equations lead to the following set of steady-state equations:

$$
\begin{aligned}
\lambda P_{000} &= \mu P_{001} \\
(\lambda+\mu)P_{001} &= \mu P_{011} + \mu P_{100} \\
(\lambda+\mu)P_{011} &= \mu P_{b11} + \mu P_{101} \\
\mu P_{b11} &= \mu P_{111}
\end{aligned}
$$

$$\begin{aligned}
\mu P_{100} &= \lambda P_{000} + \mu P_{101} \\
2\mu P_{101} &= \lambda P_{001} + \mu P_{111} \\
2\mu P_{111} &= \lambda P_{011}.
\end{aligned}$$

By using the boundary equation that the sum of the probabilities of all 7 states equals one, viz.,

$$\sum_{\forall a,B,c} P_{aBc} = 1$$

in conjunction with any six out of the above 7 equations, one may derive:

$$\begin{aligned}
P_{000} &= (4\alpha^3 + \alpha^2)A \\
P_{001} &= (4\alpha^2 + \alpha)A \\
P_{011} &= (2\alpha)A \\
P_{100} &= (4\alpha^2 + 3\alpha + 1)A \\
P_{b11} &= A \\
P_{101} &= (1 + 2\alpha)A \\
P_{111} &= A \\
\sum_{\forall a,B,c} P_{aBc} &= [4 + 8\alpha + 9\alpha^2 + 4\alpha^3]A = 1
\end{aligned}$$

where

$$A = [4 + 8\alpha + 9\alpha^2 + 4\alpha^3]^{-1}$$

and

$$\alpha = \frac{\mu}{\lambda}.$$

The throughput of the system, X_K, may be calculated as follows:

$$\begin{aligned}
X_K &= \mu\, P[\text{the last station is busy}] \\
&= \mu\, [P_{001} + P_{011} + P_{b11} + P_{101} + P_{111}] \\
&= \mu\, [(4\alpha^2 + \alpha)A + (2\alpha)A + A + (1 + 2\alpha)A + A] \\
&= \mu\, A\, [4\alpha^2 + 5\alpha + 3].
\end{aligned}$$

The average queue length of the system, L_S, may be derived as follows:

$$\begin{aligned}
L_S &= 0P_{000} + 1(P_{001} + P_{100}) + 2(P_{011} + P_{101}) + 3(P_{b11} + P_{111}) \\
&= (4\alpha^2 + \alpha)A + (4\alpha^2 + 3\alpha + 1)A + 2(2\alpha)A \\
&\quad + 2(1 + 2\alpha)A + 3A + 3A \\
&= A[8\alpha^2 + 12\alpha + 9].
\end{aligned}$$

The mean effective arrival rate to the system, λ_e, is given by:

$$\begin{aligned}\lambda_e &= \lambda P[\text{the first station is idle}]\\ &= \lambda[P_{000}+P_{001}+P_{011}]\\ &= \lambda[(4\alpha^3+\alpha^2)A+(4\alpha^2+\alpha)A+(2\alpha)A]\\ &= \lambda A[4\alpha^3+5\alpha^2+3\alpha]\\ &= \mu A[4\alpha^2+5\alpha+3]\end{aligned}$$

and the average waiting time in the system, W_S, may be obtained from Little's formula:

$$\begin{aligned}W_S &= \frac{L_S}{\lambda_e}\\ &= \frac{A[8\alpha^2+12\alpha+9]}{\mu A[4\alpha^2+5\alpha+3]}\\ &= \frac{8\alpha^2+12\alpha+9}{\mu[4\alpha^2+5\alpha+3]}.\end{aligned}$$

It may be also noticed that the average waiting time in the system consists of the following times:

$$W_S = \frac{1}{\mu}+W_b+W_B+\frac{1}{\mu} = \frac{2}{\mu}+W_b+W_B$$

where W_b and W_B are the average blocking time and the average waiting time at the intermediate buffer, respectively.

The average waiting time in the queue, W_q, consists of two elements, viz., the waiting time while an item is blocked at the first station, W_b, and the waiting time at the intermediate buffer, W_B. Applying Little's formula, one may obtain:

$$W_q = W_b+W_B = \frac{L_q}{\lambda_e} = \frac{1(P_{011}+P_{111})+2P_{b11}}{\lambda_e}.$$

Example 2: A production machine subject to failure and repair

Consider a machine with a mean service (production) rate of μ units per unit time which is subject to failure and repair at rates of β and r per unit time, respectively. The failure and repair mechanism is assumed to follow a Markov process. There are two states in the system, viz., 0 when the machine is operating and 1 when the machine is down.

Let $P_i(t)$ equal the probability of the system being in state i at time t, $i = 0,1$. Then, the equations of state of the system may easily be derived using the Markov properties and are as follows:

$$\begin{aligned}\frac{dP_0(t)}{dt}+\beta P_0(t) &= rP_1(t)\\ \frac{dP_1(t)}{dt}+rP_1(t) &= \beta P_0(t).\end{aligned}$$

Using the normalizing condition $P_0(t) + P_1(t) = 1$, with the initial condition $P_0(0) = 1$, leads to

$$P_0(t) = \frac{r}{\beta + r} + \frac{\beta}{\beta + r} e^{-(\beta + r)t}$$
$$P_1(t) = \frac{\beta}{\beta + r}[1 - e^{-(\beta + r)t}]$$

and the steady-state solutions are as follows:

$$P_0 = \frac{r}{\beta + r}$$
$$P_1 = \frac{\beta}{\beta + r}.$$

It may be noted that these steady-state results could be obtained by considering that the machine operates for a mean time of $1/\beta$ and is down for a mean time of $1/r$ during repair.

The steady-state availability, A, of the system is given by

$$A = \frac{MTTF}{MTTF + MTTR} = \frac{\frac{1}{\beta}}{\frac{1}{\beta} + \frac{1}{r}} = \frac{r}{\beta + r}$$

where $MTTF$ is the mean time to failure and $MTTR$ is the mean time to repair.

So, as μ is the mean service rate of the machine, the average steady-state output, X, would be

$$X = \mu A = \mu \left(\frac{MTTF}{MTTF + MTTR} \right).$$

References

1. Altiok, T. (1997), *Performance Analysis of Manufacturing Systems*, Springer-Verlag.
2. Barnett, S. (1994), *Matrices: Methods and Applications*, Oxford Applied Mathematics and Computing Science Series.
3. Bellman, R. (1960), *Matrix Analysis*, McGraw-Hill.
4. Billinton, R. and Allan, R.N. (1983), *Reliability Evaluation of Engineering Systems: Concepts and Techniques*, Longman Scientific & Technical.
5. Brogan, W.L. (1974), *Modern Control Theory*, Quantum Publishers.
6. Cramer, H. (1946), *Mathematical Methods of Statistics*, Princeton University Press.
7. Feller, W. (1957), *An Introduction to Probability Theory and Its Applications*, Vol. I, 2nd Edition, John Wiley & Sons.
8. Feller, W. (1966), *An Introduction to Probability Theory and Its Applications*, Vol. II, 2nd Edition, John Wiley & Sons.
9. Gross, D. and Harris, C.M. (1985), *Fundamentals of Queueing Theory*, Second Edition, John Wiley & Sons.
10. Hahn, G.J. and Shapiro, S.S. (1967), *Statistical Models in Engineering*, John Wiley & Sons.

11. Horn, R.A. and Johnson, C.J. (1991), *Topics in Matrix Analysis*, Cambridge University Press.
12. Kemeny, J.G. and Snell, J.L. (1960), *Finite Markov Chains*, Van Nostrand.
13. Kleinrock, L. (1975), *Queueing Systems, Vol. I: Theory*, John Wiley & Sons.
14. Meyer, P.L. (1965), *An Introduction to Probability Theory and Its Applications*, Addison-Wesley.
15. Montgomery, C.D. (1996), *Introduction to Statistical Quality Control*, John Wiley & Sons.
16. Mood, A.M. and Graybill, F. (1963), *Introduction to the Theory of Statistics*, McGraw-Hill.
17. Neuts, M.F. (1981), *Matrix-Geometric Solutions in Stochastic Models – An Algorithmic Approach*, John Hopkins University Press.
18. Neville, A.M. and Kennedy, J.B. (1964), *Basic Statistical Methods for Engineers and Scientists*, International Textbook Co.
19. Papadopoulos, H.T., Heavey, C. and Browne, J. (1993), *Queueing Theory in Manufacturing Systems Analysis and Design*, Chapman & Hall.
20. Parzen, E. (1960), *Modern Probability Theory and Its Applications*, John Wiley & Sons.
21. Ross, S.M. (1985), *Introduction to Probability Models*, John Wiley & Sons.
22. Shapiro, S.S. (1980), *How to Test Normality and Other Distributional Assumptions*, Vol. 3, *The ASQC Basic References in Quality Control: Statistical Techniques*, ASQC, Milwaukee, WI.

B

Algorithms/Procedures Details and Guide to Use

The following material appears at the website associated with this book.

The details of the algorithms available at the website associated with this book are given below:

1. Abbreviation is the name by which the algorithm/procedure is named at the web-site associated with this book
2. Author name
3. Coder name
4. The type of algorithm or procedure used, e.g., evaluative/predictive or generative/optimization
5. Description of system to which the algorithm/procedure may be applied including size restrictions, if any
6. Output of the algorithm/procedure
7. Reference

Only one algorithm is capable of handling open loop (unsaturated) serial production lines (EXPAN).

C. T. Papadopoulos et al., *Analysis and Design of Discrete Part Production Lines*, Springer Optimization and Its Applications,
DOI: 10.1007/978-0-387-89494-2_9, © Springer Science+Business Media, LLC 2009

B.1 Markovian

Abbreviation: MARKOV
Author: Cathal Heavey, University of Limerick, Ireland
Coder: Cathal Heavey
Algorithm: Evaluative/Predictive
Description: Given a detailed specification of a reliable or unreliable production line with single machines at each station with service and repair times distributed according to an Erlang-k ($k \geq 1$) distribution and the times to failure following an exponential distribution. Intermediate buffers of finite capacity are allowed between any two successive stations of the saturated line. With current computer capabilities the algorithm is able to handle systems with up to 300,000 states/equations in reasonable time.
Output: Exact throughput of the specified production line
Reference: Heavey, Papadopoulos and Browne (1993)

B.2 Decomposition-1

Abbreviation: DECO-1
Author: Yves Dallery (Ecole Centrale Paris) and Yannick Frein (Institut Polytechnique de Grenoble, France)
Coder: Michael Vidalis (University of the Aegean, Greece)
Algorithm: Evaluative/Predictive
Description: The algorithm is capable of handling any size of serial single machine station reliable saturated production lines with exponential service times and intermediate buffers of finite capacity using the decomposition approach.
Output: Throughput of the specified production line
Reference: Dallery and Frein (1993), among other papers

B.3 Expansion

Abbreviation: EXPAN
Author: Laoucine Kerbache and James MacGregor Smith
Coder: Suchant Jain and James MacGregor Smith
Algorithm: Evaluative/Predictive
Description: The algorithm is capable of handling unsaturated reliable serial production lines with parallel machines at each station with finite intermediate buffers using a decomposition methodology.
Output: Throughput of the specified production line
Reference: Kerbache and MacGregor Smith (1987) and Jain and MacGregor Smith (1994)

B.4 Aggregation

Abbreviation: AGGRE
Author: Jonh-Tae Lim, Semyon Meerkov and Ferudun Top
Coder: Jonh-Tae Lim, Semyon Meerkov and Ferudun Top
Algorithm: Evaluative/Predictive
Description: The algorithm is capable of handling asymptotically reliable saturated transfer lines (with the machines having identical cycle times) of any size using the aggregation approach and involving forward and backward loops to obtain convergence.
Output: Throughput of the specified transfer line
Reference: Jonh-Tae Lim, Semyon Meerkov and Ferudun Top (1990)

B.5 Decomposition-2

Abbreviation: DECO-2
Author: Alexandros Diamantidis (Aristotle University of Thessaloniki, Greece)
Coder: Alexandros Diamantidis
Algorithm: Evaluative/Predictive
Description: The algorithm is capable of handling saturated long lines (with over 1000 stations in series) with exponential service times, parallel identical machines at each station and finite intermediate buffers using a decomposition methodology.
Output: Throughput of the specified production line. *Note 1:* For the number of stations, $K = 2$, the algorithm gives the exact equations and numerical results of the two-station production line with parallel machines at each station. *Note 2:* For the number of parallel machines at each station, $s_i = 1,\ i = 1,2,\ldots,K$, the algorithm gives the same equations and numerical results as those originally developed by Gershwin (1987, 1994).
Reference: Diamantidis, Papadopoulos and Heavey (2007)

B.6 Two-Level Work-Load Allocation

Abbreviation: TLWLA
Author: John Buzacott and George J. Shanthikumar
Coder: Michael Vidalis and Alexandros Diamantidis
Algorithm: Stand-alone Optimization
Description: It is a self-contained algorithm which develops an approximate two-level work-load allocation for saturated production lines with single machine reliable stations and specified identical or non-identical buffer sizes.
Output: Throughput and two-level work-load approximation of the specified production line
Reference: Buzacott and Shanthikumar (1993)

B.7 Simulated Annealing

Abbreviation: SA
Author: Diomidis Spinellis (Athens University of Economics and Business) and Chrissoleon Papadopoulos (Aristotle University of Thessaloniki, Greece)
Coder: Diomidis Spinellis
Algorithm: Generative/Optimization
Description: It is an optimizing search algorithm based on the methodology of simulated annealing which communicates with appropriate evaluative/predictive algorithm(s) to solve large production lines.
Output: Work-load-, Buffer-, and Server-allocations, in single or double or triple combinations
Reference: Spinellis and Papadopoulos (2000a)

B.8 Genetic Algorithm

Abbreviation: GA
Author: Diomidis Spinellis and Chrissoleon Papadopoulos
Coder: Fanis Karagiannis and Diomidis Spinellis
Algorithm: Generative/Optimization
Description: It is an optimizing search algorithm based on the methodology of genetic programming which communicates with appropriate evaluative/predictive algorithm(s) to solve large production lines.
Output: Work-load-, Buffer-, and Server-allocations, in single or double or triple combinations
Reference: Papadopoulos and Karagiannis (2001) and Spinellis and Papadopoulos (2000b)

B.9 Complete Enumeration

Abbreviation: CE
Author: Michael Vidalis and Chrissoleon Papadopoulos
Coder: Michael Vidalis and Diomidis Spinellis
Algorithm: Generative/Optimization
Description: It is an optimizing search algorithm based on enumeration which communicates with appropriate evaluative/predictive algorithm(s) to solve only small production lines with constraints with respect to total number of buffer slots and total number of servers.
Output: Buffer- and Server-allocations, in single or double combinations
Reference: enumeration, CE—

B.10 Buffer Allocation

Abbreviation: BA
Author: Chrissoleon Papadopoulos and Michael Vidalis
Coder: Michael Vidalis and Diomidis Spinellis
Algorithm: Stand-alone optimization
Description: It is a self-contained algorithm which initially specifies a near optimal buffer allocation and being directly connected to the Markovian algorithm develops via the Hooke and Jeeves search mechanism the optimal buffer allocation and the associated optimal throughput. It solves small reliable or unreliable production lines.
Output: Buffer allocation and throughput of the specified production line
Reference: Papadopoulos and Vidalis (2001a)

The authors would be very pleased to hear from researchers or practitioners who wish to have an algorithm/procedure developed by them to be included at the website. Hopefully in time a very comprehensive set of algorithms/procedures for the analysis/design of serial production lines would become available for all to use. This could well be the first step to having at the website a set of algorithms/procedures which have been found to be of value in design and analysis of general manufacturing systems.

References

1. Buzacott, J.A. and Shanthikumar, J.G. (1993), *Stochastic Models of Manufacturing Systems*, Prentice Hall.
2. Dallery, Y. and Frein, Y. (1993), On decomposition methods for tandem queueing networks with blocking, *Operations Research*, Vol. 41, No. 2, pp. 386–399.
3. Diamantidis, A.C., Papadopoulos, C.T., and Heavey, C. (2007), Approximate analysis of serial flow lines with multiple parallel-machine stations, *IIE Transactions*, Vol. 39, issue 4, pp. 361–375.

4. Gershwin, S.B. (1987), An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking, *Operations Research*, Vol. 35, pp. 291–305.
5. Gershwin, S.B. (1994), *Manufacturing Systems Engineering*, Prentice Hall.
6. Heavey, C., Papadopoulos, H.T., and Browne, J. (1993), The throughput rate of multistation unreliable production lines, *European Journal of Operational Research*, Vol. 68, pp. 69–89.
7. Jain, S. and Smith, J.M. (1994), Open finite queueing networks with $M/M/C/K$ parallel servers, *Computers & Operations Research*, Vol. 21, No. 3, pp. 297–317.
8. Kerbache, L. and MacGregor Smith, J. (1987), The generalized expansion method for open finite queueing networks, *European Journal of Operational Research*, Vol. 32, pp. 448–461.
9. Lim, J.-T., Meerkov, S.M., and Top, F. (1990), Homogeneous, asymptotically reliable serial production lines: Theory and a case study, *IEEE Transactions on Automatic Control*, Vol. 35, No. 5, pp. 524–534.
10. Papadopoulos, C.T. and Karagiannis, T.I. (2001), A genetic algorithm approach for the buffer allocation problem in unreliable production lines, *International Journal of Operations and Quantitative Management*, Vol. 7, No. 1, pp. 23–35.
11. Papadopoulos, H.T. and Vidalis, M.I. (2001a), A heuristic algorithm for the buffer allocation in unreliable unbalanced production lines, *Computers & Industrial Engineering*, Vol. 41, pp. 261–277.
12. Spinellis, D.D. and Papadopoulos, C.T. (2000a), A simulated annealing approach for buffer allocation in reliable production lines, *Annals of Operations Research*, Vol. 93, pp. 373–384.
13. Spinellis, D.D. and Papadopoulos, C.T. (2000b), Stochastic algorithms for buffer allocation in reliable production lines, *Mathematical Problems in Engineering*, Vol. 5, pp. 441–458.

C

Glossary

C.1 General Acronyms

Symbol	*Meaning*
ABC	ABC analysis in inventory/stock control
ABC	Actvity-based costing
Arena	A simulation package
CDIM	Customer-driven intelligent manufacturing
DSS	Decision support system
eM-plant	A simulation package
PL	Production line
CI	Confidence interval
FIFO	First-In, First-Out
FCFS	First-Come, First-Served
LIFO	Last-In, First-Out
FMS	Flexible manufacturing system
FMC	Flexible manufacturing cell
FAS	Flexible assembly system
GT	Group technology
CAD	Computer-aided design
CAM	Computer-aided manufacturing
CAE	Computer-aided engineering
CNC	Computer numerically controlled
NC	Numerically controlled
CIM	Computer-integrated manufacturing
JIT	Just-In-Time
TQM	Total quality management
WIP	Work-In-Process or Work-In-Progress
$\overline{WIP}$	Average WIP
WF	Workforce

(*continued*)

C. T. Papadopoulos et al., *Analysis and Design of Discrete Part Production Lines*, Springer Optimization and Its Applications,
DOI: 10.1007/978-0-387-89494-2_10,

General Acronyms — (Continued)

Symbol	*Meaning*
MRP	Materials requirements planning
BAS	Blocking after service
BBS	Blocking before service
WAP	Work-load allocation problem
BAP	Buffer allocation problem
SAP	Server allocation problem
W + S	Simultaneous work-load and server allocation
W + B	Simultaneous work-load and buffer allocation
S + B	Simultaneous server and buffer allocation
W + S + B	Simultaneous work-load and server and buffer allocation
PARTAN	The steepest ascent method of parallel tangents
SOR	Successive over relaxation factor or method
DP	Dynamic programming
SA	Simulated annealing
GA	Genetic algorithms
TS	Tabu search algorithm
w.r.t.	With respect to
r.v.	Random variable
$c.v.(X)$	Coefficient of variation of the r.v. X

C.2 Production Lines

Symbol	*Meaning*
K	Number of stations in a production line
B_i	Buffer i, $i = 2,3,\ldots,K$ placed before station i in a K-station production line
B_i	Capacity of buffer B_i, $i = 2,3,\ldots,K$
N	Total number of buffer slots to be allocated among the $K-1$ intermediate buffers, B_i, $i = 2,3,\ldots,K$ of a K-station production line
N_i	Number of buffer slots allocated to buffer B_i, $i = 2,3,\ldots,K$ $(0 \leq N_i \leq B_i)$ in the buffer allocation problem
WS_i or M_i	Work-station i in a K-station line. This may be single- or multi-machine work-station
μ_i	Mean service or processing rate of station i
$w_i = 1/\mu_i$	Mean service or processing time (work-load) of station i
β_i	Mean failure rate of station i
$1/\beta_i = MTTF$	mean time to failure (MTTF) of station i
r_i	Mean repair rate of station i
$1/r_i = MTTR$	Mean repair time or mean time to repair (MTTR) of station i

(*continued*)

Production Lines — (Continued)

Symbol	*Meaning*
A_i	Availability of station i
X_K	Throughput or mean production (output) rate of a K-station production line
$1/X_K$	Mean production time of a K-station production line
X_i	Throughput or mean production (output) rate of the ith station of a production line
ρ_i	Utilization of the ith work-station of a production line
e_i	Efficiency or mean effective service rate of the ith station of a production line (equal to $\mu_i A_i$)

C.3 Decomposition Approach

Symbol	*Meaning*
L	Original production line that is decomposed in the decomposition approach
L_i	Sub-line i, $i = 1,2,\ldots,K-1$, in the decomposition approach
M_i^u	The part of the original line, L, upstream buffer B_{i+1}. It is a pseudo work-station for $i = 2,\ldots,K-1$. For $i = 1$ it holds: $M_1^u = M_1$
M_{i-1}^d	The part of the original line, L, downstream buffer B_i. It is a pseudo work-station for $i = 2,\ldots,K-1$. Special case: It holds: $M_K^d = M_K$
w_i	The mean service time (work-load) of station i, $i = 1,2,\ldots,K$, in the original line, L ($w_i = 1/\mu_i$)
w_i^d	The sum of the mean service time and the possible blocking time at station i in the original line, L ($w_i^d = 1/\mu_i^d$)
w_{i-1}^u	The sum of the mean service time at station $i-1$ and the possible starvation time of station $i-1$, $i = 2,\ldots,K$ in the original line, L ($w_{i-1}^u = 1/\mu_{i-1}^u$)
μ_i	The mean service rate of station i, $i = 1,2,\ldots,K$ in the original line, L
μ_i^u	The mean service (processing) rate of the upstream station of buffer B_{i+1}, $i = 1,\ldots,K-1$ in the decomposition approach
μ_i^d	The mean service (processing) rate of the downstream station of buffer B_i, $i = 2,\ldots,K$ in the decomposition approach
p_i^{bl}	The blocking probability of a station i
p_i^{st}	The starvation probability of a station i
p_i^{bl}	The blocking probability of sub-line L_i, $i = 1,\ldots,K-1$
p_{i-1}^{st}	The starvation probability of sub-line L_{i-1}, $i = 2,\ldots,K$
X_{DECO}	Throughput or mean production (output) rate of a K-station production line obtained from application of the decomposition method
X_{SIM}	Simulated throughput or mean production (output) rate of a K-station production line obtained from application of a simulation package

C.4 Markovian Model

Symbol	*Meaning*
λ	Mean arrival rate
$N(t) =$ $[N_1(t), N_2(t)]$	A two-dimensional stochastic process in the context of a queueing network (q.n.)
$N_1(t)$	The number of jobs queued up in front of the first station of the q.n.
$N_2(t)$	The state of the sub-network of the q.n. at time t
QBD	Quasi birth and death process
$\mathbf{e}$	A $(m \times 1)$ column-vector with all elements equal to 1
P	Steady-state probability
$A = A_0 + A_1 + A_2$	The conservative matrix in the stochastic process model
n_i	Status of buffer i in the Markovian model
s_i	Status of station i in the Markovian model
m_K^B	Number of states in the sub-network of a K-station line with identical buffers, each of capacity B
$m_K^{B_2,\ldots,B_K}$	Number of states in the sub-network of a K-station line with non-identical buffers, with buffer capacities $B_2, \ldots, B_K$
P_i	The number of phases of the service time distribution of the ith station in the Markovian model
R_i	The number of phases of the repair time distribution of the ith station in the Markovian model
$m_{K,P}^{B,R}$	The number of states in the sub-network with K-stations, each buffer having the same capacity B, each service time distribution having P phases and each repair time distribution having R phases in the Markovian model
$m_{K,P_1,P_2,..,P_K}^{B_2,..,B_K,R_1,R_2,..,R_K}$	The number of states in the sub-network of a K-station system with buffer capacities $B_2, \ldots, B_K$. The number of phases of each station's service time distribution is equal to $P_1, P_2, \ldots, P_K$ phases and the number of phases of each station's repair time distribution is equal to $R_1, R_2, \ldots, R_K$ in the Markovian model

C.5 Expansion Method

Symbol	*Meaning*
h	The holding node established in the expansion method
Λ	External Poisson arrival rate to the network
λ_j	Poisson arrival rate to node j
$\tilde{\lambda}_j$	Effective arrival rate to node j

(*continued*)

Expansion Method — (Continued)

Symbol	*Meaning*
μ_j	Exponential mean service rate at node j
$\tilde{\mu}_j$	Effective service rate at node j due to blocking
p_K	Blocking probability of finite queue of size K
p'_K	Feedback blocking probability in the expansion method
p_0^j	Unconditional probability that there is no unit in the service channel at node j (either being served or being held after service)
X	Throughput (mean production rate)

C.6 Aggregation Method

Symbol	*Meaning*
Λ_i	The loss parameter of the ith machine
q_i	$= 1 - \varepsilon\,\Lambda_i, \quad \varepsilon << 1$
	The probability machine i produces a part during a time slot/period
Λ_i^f	The loss parameter of the ith machine in the forward aggregation
Λ_i^b	The loss parameter of the ith machine in the backward aggregation
X_K^f	The throughput of the K-machine line in the forward aggregation
X_K^b	The throughput of the K-machine line in the backward aggregation

C.7 Design Problems

Symbol	*Meaning*
$PI(\mu_1,\ldots,\mu_K,N_1,\ldots,N_K)$	Performance index
MARKO	Markovian algorithm
DECO	Decomposition algorithm
EXPAN	Expansion algorithm
SA	Simulated annealing algorithm
GA	Genetic algorithm
CE	Complete enumeration
RE	Reduced enumeration
OBA	Optimal buffer allocation
OSA	Optimal server allocation
LBAS	Linear buffer allocation scheme
w	$:= (w_1, w_2, \ldots, w_K) =$ The mean service times (work-load) vector in WAP

(*continued*)

Design Problems — (Continued)

Symbol	*Meaning*
$\mathbf{s}$	$:= (S_1, S_2, \ldots, S_K) =$ The servers vector in SAP
$\mathbf{n}$	$:= (N_2, N_3, \ldots, N_K) =$ The buffers vector in BAP
$\mathbf{g}$	$:= (g_2, g_3, \ldots, g_K) =$ The gradient vector in the gradient method
$\lfloor x \rfloor$	The floor function, denoting the largest integer less than or equal to x
$(a \ldots b)$	An open interval containing all values from a to b excluding the two endpoints a and b
$[a \ldots b]$	A closed interval containing all values from a to b including the two endpoints a and b
$[a \ldots b)$	A half closed interval containing all values from a to b including a but excluding b
$(a \ldots b]$	A half closed interval containing all values from a to b excluding a but including b
$[I]$	Buffer classes of first generation $(I = 0, \ldots, N)$. It consists of all buffer allocations with the first element of the buffer vector equal to I
$[I,J]$	Buffer classes of second generation, $(I = 0, \ldots, N,\ J = 0, \ldots, N+1-I)$, and so on. It consists of all buffer allocations with the first two elements of the buffer vector equal to I and J, resp.

C.8 Cost Considerations

Symbol	*Meaning*
AHP	Analytical hierarchical processes
R	Selling price of a unit of the product
C	Product unit cost
C_h	Inventory unit holding cost
I	Interest annual rate
FU	Financial units
b_i	A net present value coefficient associated with each buffer slot
$P.W.F.^*$	Present worth factor
$P.W.V.$	Present worth value
$\delta(a)$	The Kronecker delta function defined by: $\delta(a) = \begin{cases} 1, \text{ if } a > 0 \\ 0, \text{ if } a \leq 0. \end{cases}$
$F_i,\ i = 1,2,3$	Profit maximization objective functions
$G_j,\ j = 1,2,3$	Cost minimization objective functions

C.9 Mathematical Fundamentals

Symbol	*Meaning*
I	The identity matrix of dimension $n \times n$
A^T	The transpose matrix of dimension $n \times m$ of matrix A of dimension $m \times n$
$[\widetilde{F}(s)]$	The Laplace transform of the function $[F(t)]$
EX	The mean value or expected value of the r.v. X
$VarX$	The variance of the r.v. X
E_k	Erlang distribution with k phases
C_2	Coxian distribution with two phases
$\psi_i,\ i = 1,2,3$	The first three moments of a probability distribution
T	The transition probability matrix in Markov chains
$F = [I - A]^{-1}$	The fundamental matrix in Markov chains
QN	Queueing network
q.s.	Queueing system
$A/B/c : D/E/F$	Kendall's notation of a queueing system (q.s.), where:
A	A descriptor of the statistics of the arrival into the q.s.
If $A = M$	The arrival process is Poisson
If $B = M$	The service time is an exponential distribution
If $A = B = D$	Both the inter-arrival and the service time distributions are deterministic
If $A = GI$	General independent arrival process
If $B = G$	General service time distribution
B	A descriptor of the service time distribution of each of the servers of the q.s.
c	The number of servers of the q.s.
D	A descriptor of the queueing discipline
E	The overall size of the q.s.
F	A descriptor of the population from which the arriving units to the q.s. come
i.i.d.	Independent identically distributed
p.d.f.	Probability density function
C.D.F.	Cumulative distribution function
$P_n(t)$	Probability that there are n units in the q.s. at time t
P_n	Probability that there are n units in the q.s. in steady state
L_S	The expected number of units in the q.s. in steady state
L_q	The expected number of units waiting for service in steady state
W_S	The expected time spent in the q.s.
W_q	The expected time spent waiting for service
$f_{W_S}(t)$	p.d.f. of the total time spent in the q.s. under a given queueing discipline

(*continued*)

Mathematical Fundamentals — (Continued)

Symbol	*Meaning*
$f_{W_q}(t)$	p.d.f. of the time spent waiting for service under a given queueing discipline
λ_e	Mean effective arrival rate
$I_k(x)$	The modified Bessel function of first kind and kth order
λ	Mean arrival rate
μ	Mean service rate
$\delta(t)$	The Dirac delta function in the q.s. $M/M/1 : FCFS/\infty/\infty$
P_c	The Erlang's loss formula in the q.s. $M/M/c : FCFS/c/\infty$
L_S	$:= \lambda_e W_S$, Little's formula
L_q	$:= \lambda_e W_q$, Little's formula

C.10 Accompanying Algorithms and Procedures

Symbol	*Meaning*
MARKOV	Markovian algorithm
DECO-1	Decomposition algorithm for solving reliable exponential single-machine station production lines
EXPAN	Expansion algorithm
AGGRE	Aggregation algorithm
DECO-2	Decomposition algorithm for solving reliable exponential parallel-machine station production lines
TLWLA	Two-level work-load allocation algorithm
SA	Simulated annealing algorithm
GA	Genetic algorithm
CE	Complete enumeration
BA	Buffer allocation procedure in unreliable production lines

D

Conference Participants: Presenters and Attendees

These are the lists of participants:Presenters and attendees at the five Hellenic (Aegean and Ionian) International Conferences on Analyis, Design and Optimization of Manufacturing Systems. The conferences were held in Greece in the islands of Samos, Tinos, Tinos, Samos, and Zakynthos in 1997, 1999, 2001, 2003, and 2005, respectively.

D.1 Conference Participants: Presenters

L/I	*Participant Name*	Affiliation
1	ALLON G.	Technion, Haifa, Israel
2	ALTIOK Tayfur	Rutgers University, USA
3	ASKIN Ronald	University of Arizona, USA
4	Avsar Zeynep Muge	Middle East Technical University, Turkey
5	AXSATER Sven	Lund University, Sweden
6	BATZIAS F.A.	University of Piraeus, Greece
7	BENJAAFAR Saif	University of Minnesota, USA
8	BILALIS Nikos	Tecnhical University of Crete, Greece
9	BOHORIS Giorgos	University of Piraeus, Greece
10	BROUNS Gido	Eindhoven University of Technology, The Netherlands
11	BUITENHEK R.	Technical University of Twente, Holland
12	BURMAN Mitchell	Analytics, Inc., USA
13	BUZACOTT John	York University, Toronto, Canada
14	CARAMANIS Michael	Boston University, USA
15	COLLEDANI Marcello	Politecnico di Milano, Italy

(*continued*)

C. T. Papadopoulos et al., *Analysis and Design of Discrete Part Production Lines*, Springer Optimization and Its Applications,
DOI: 10.1007/978-0-387-89494-2_11,

Conference Participants: Presenters — Continued

L/I	*Participant Name*	Affiliation
16	DALLERY Yves	Ecole Centrale, Paris and Groupe HEC, France
17	DASKALAKI Sophia	University of Patras, Greece
18	DIAMANTIDIS Alexandros	Aristotle University of Thessaloniki, Greece
19	Di MASCOLO Maria	Laboratoire d' Automatique de Grenoble, France
20	DOGRU Mustafa Kemal	Eindhoven Technical University, The Netherlands
21	DOUNIAS Georgios	University of the Aegean, Greece
22	DUENYAS Izak	University of Michigan
23	EMIRIS Dimitrios	University of Piraeus, Greece
24	FADILOGLU Murat	Bilkent University
25	FURMANS Kai	IFL - University of Karlsruhe, Germany
26	GEORGIADIS Patroklos	Aristotle University of Thessaloniki, Greece
27	GERAGHTY John	Dublin City University, Ireland
28	GERSHWIN Stan	MIT, USA
29	GOH Thiam Hock Jason	National University of Singapore
30	GRUBBSTROM Robert	Department of Production Economics, Linkoping Institute of Technology, Sweden
31	GUPTA Omprakash	Nirma Institute of Management, India
32	GURKAN Gul	Tilburg University, Holland
33	HAGHNEVIS Moeed	University of Tehran, Iran
34	HEAVEY Cathal	University of Limerick, Ireland
35	HELBER Stephan	University of Hannover, Germany
36	HONGLER Max Oliver	EPFL, Switzerland
37	HU Jian-Quiang	Boston University, USA
38	IOANNIDIS Stratos	University of the Aegean, Greece
39	IOANNOU Georgios	Athens University of Economics and Business, Greece
40	ISHIKURA Hiroki	Osaka Gakuin University, Japan
41	JAFARI Mohsen	Rutgers University, USA
42	KARAESMEN Fikri	Koc University, Turkey
43	KERBACHE Laoucine	HEC School of Management, France
44	KHODADADEGAN Yasaman	University of Tehran, Iran
45	KNIKER T.	Analytics Inc., USA
46	KOCK Ad	Eindhoven University of Technology, The Netherlands
47	KOUIKOGLOU Vassilis	Technical University of Crete, Greece
48	KOUKOUMIALOS S.	University of Thessaly, Grrece

(*continued*)

Conference Participants: Presenters — Continued

L/I	*Participant Name*	Affiliation
49	KRISHNAMURTHY Ananth	Rensselaer Polytechnic Institute, Troy, New York, USA
50	KUHN Heinrich	Catholic University of Eichstaett, Germany
51	KYRIAKIDIS Nondas	University of the Aegean, Greece
52	LAIOS Labros	University of Piraeus, Greece
53	LEVANTESI Raniero	Politecnico di Milano, Italy
54	LI Jingshan	
55	LIBEROPOULOS Giorgos	University of Thessaly, Greece
56	LOU Sheldon	CSU San Marcos
57	LOUKIS E.	Univeristy of the Aegean, Greece
58	MACGREGOR SMITH James	University of Massachusetts, Amherst, USA
59	MAKRIS Spilios	Telcordia Technologies, USA
60	MAKROKANIS Georgios	Neoset, Greece
61	MALHAME Roland	Ecole Polytechnique de Montreal and GERAD, Canada
62	MATTA Andrea	Politecnico di Milano, Italy
63	MEERKOV Semyon	University of Mishigan, USA
64	MITTMAN Stefanos	Intracom, Greece
65	MOCANU Stephane	Laboratoire d'Automatique de Grenoble, France
66	MODESTOU Thomas	BSHG, Greece
67	MOUSTAKIS Vassilis	Tecnhical University of Crete, Greece
68	NENES George	Aristotle University of Thessaloniki, Greece
69	NONAKA Youichi	Hitachi, Japan
70	O'KELLY M.E.J.	National University of Ireland, Galway, Ireland
71	PANAGIOTOU N.	NTUA, Greece
72	PAPACHRISTOS Sotiris	University of Ioanina, Greece
73	PAPADOPOULOS Chrissolleon T.	Aristotle University of Thessaloniki, Thessaloniki, Greece
74	PAPPIS Kostas	University of Piraeus, Greece
75	PARDALOS Panos	University of Florida, USA
76	PERROS Harry	North Carolina State University, USA
77	PETERING Matthew	National University of Singapore, Singapore
78	POLLOCK Stephen	University of Michigan, USA
79	PRASTACOS Gregory	Athens University of Economics and Business, Greece
80	RAI Sudhendu	Xerox Corporation, USA
81	RAVIV Tal	Technion, Haifa, Israel

(*continued*)

Conference Participants: Presenters — Continued

L/I	*Participant Name*	Affiliation
82	SAWIK Tadeusz	AGH University of Science and Technology, Poland
83	SBITI Naval	University Mohammed V-Agdal, Maroc
84	SELCUK Baris	Eindhoven University of Technology, The Netherlands
85	SHANTHIKUMAR, J. George	University of California, Berkeley, USA
86	SKINTZI Georgia	Athens University of Economics and Business, Greece
87	SOFIANOPOULOU Stella	University of Piraeus, Greece
88	SOLOMOS Lefteris	Delta Dairy S.A., Greece
89	SPINELLIS Diomidis	Athens University of Economics and Business, Greece
90	TAGARAS Georgios	Aristotle University of Thessaloniki, Greece
91	TAN Baris	Koc University, Turkey
92	TAN Tarkan	Eindhoven University of Technology, The Netherlands
93	TATSIOPOULOS Ilias	NTUA, Greece
94	TEMPELMEIER Horst	University of Cologne, Germany
95	TOLIO Tullio	Politecnico di Milano, Italy
96	TRIANTAPHYLLOU Evangelos	Louisiana State University, USA
97	TSAKONAS Athanassios	University of the Aegean, Greece
98	VAN DER WAL Jan	Eindhoven University of Technology, The Netherlands
99	VAN HOUTUM Gert-Jan	Eindhoven University of Technology, The Netherlands
100	VAN NYEN Pieter	Eindhoven University of Technology, The Netherlands
101	VAN OOIJEN Henny	Eindhoven University of Technology, The Netherlands
102	VAN VUUREN Marcel	Eindhoven University of Technology, The Netherlands
103	VAN WOENSEL Tom	Eindhoven University of Technology, The Netherlands
104	VANDAELE Nico	University of Antwerp, Belgium
105	VIDALIS Michalis	University of the Aegean, Greece
106	VLACHOS Dimitrios	Aristotle University of Thessaloniki, Greece
107	VOUROS George	University of the Aegean, Greece

(*continued*)

Conference Participants: Presenters — Continued

L/I	*Participant Name*	Affiliation
108	WINANDS Erik	Eindhoven University of Technology, The Netherlands
109	XANTHAS Alekos	Tecnhical University of Crete, Greece
110	YAO David	Columbia University, USA
111	YERALAN Sencer	University of Florida, Gainesville, USA
112	ZAZANIS Michael	Athens University of Economics and Business, Greece
113	ZIJM W.H.M.	University of Twente, The Netherlands
114	ZIKOPOULOS Christos	Aristotle University of Thessaloniki, Greece
115	ZOGRAFOS Kostas	Athens University of Economics, Greece

D.2 Conference Participants: Attendees

L/I	*Participant Name*	Affiliation
1	ADAMIS Ioannis	INTRACOM, Greece
2	ALAYALIS Thanos	DELTA, Greece
3	ANDROUTSOPOULOS K.	Athens University of Economics and Business, Greece
4	APERGI A.	University of Piraeus, Greece
5	ARAVOSSITAS Giorgos	Univeristy of the Aegean, Greece
6	BALASKAS P.	University of Piraeus, Greece
7	BEKOS N.	University of Piraeus, Greece
8	BELESIOTOU F.	University of Piraeus, Greece
9	BLESSIOS N.	University of Piraeus, Greece
10	CARAMANI A.	University of the Aegean, Greece
11	CHALIKAS N.	University of Piraeus, Greece
12	CHRISTOFIDES Tony	University College, Galway, Ireland
13	CHRISTOPOULOS K.	University of Piraeus, Greece
14	DARZENTAS Jenny	University of the Aegean, Greece
15	DARZENTAS John	University of the Aegean, Greece
16	DASKALOPOULOS Konstantinos	INTRACOM, Greece
17	DESSOUKY Mohamed	University of Southern California, USA
18	DIMOPOULOS Ilias	Business Consultant, Theesaloniki, Greece
19	DIMOPOULOU Helen	GSRT, Greece
20	EMERY Helen	Rigel Corporation, USA

(*continued*)

Conference Participants: Attendees — Continued

L/I	*Participant Name*	Affiliation
21	FILIPPIDIS D.	Kraft Foods Hellas, Greece
22	FLESSAS George	University of the Aegean, Greece
23	FOLVEN Gary	Kluwer Academic Publishers, USA
24	FONIADAKI A.	University of Piraeus, Greece
25	FRANTZESKAKIS Kyriakos	ELVAL, Greece
26	GALANI V.	BSHG, Greece
27	GALIATSATOS Christos	Coca Cola HBC, Greece
28	GAMBROGIANNI C.	Univeristy of Piraeus, Greece
29	GANAS G.	University of Ioannina, Greece
30	GAONKAR R.	National University of Singapore, Singapore
31	GEORGIADIS Efstathios	BSHG, Greece
32	GERARDOU A.	BSHG, Greece
33	GERSHWIN ALLOU Francis	USA
34	GERSHWIN J.	USA
35	GIASTA E.	University of Piraeus, Greece
36	GIONI, S.	University of Piraeus, Greece
37	GLAROS A.	BSHG, Greece
38	GLYKAS Michalis	University of the Aegean, Greece
39	GOTSIAS A.	University of the Aegean, Greece
40	HADJISAVVAS Nikolaos	University of the Aegean, Greece
41	ILIADIS A.	Athens 2004 Olympic Games, Greece
42	KAKOS A.	University of Piraeus, Greece
43	KAKOUDAKIS J.	University of Piraeus, Greece
44	KALLIANOS Theodoros	European Commission
45	KAPSIS George	Metaxa S.A.
46	KARAGIANNIS F.	University of the Aegean, Greece
47	KARAKAPILIDIS N.	University of Patras, Greece
48	KARAKOSTAS Christos	Eurometal, Greece
49	KARAKOSTAS Georgios	University of the Aegean, Greece
50	KARAPATAKIS S.	Greece
51	KARAYANNIS George	Perefetti Van Melle, Greece
52	KATRAMADOS Y.	University of Piraeus, Greece
53	KATRAVAS D.	BSHG, Greece
54	KATSIKAS Socratis	University of the Aegean , Greece
55	KAYAVA Irene	Univeristy of the Aegean, Greece
56	KEHAGIAS J.	University of the Aegean, Greece
57	KLOUTSINIOTI A.	University of Piraeus, Greece
58	KONTORIS P.	University of Piraeus, Greece
59	KOUSSENIDIS Dimitrios	Aristotle University of Thessaloniki, Greece

(*continued*)

Conference Participants: Attendees — Continued

L/I	*Participant Name*	Affiliation
60	KOUTSOPODIOTIS T.	University of Piraeus, Greece
61	LABIRIS K.	University of Piraeus, Greece
62	LAPSATIS George	BSHG, Greece
63	LEKKAS Themis	University of the Aegean, Greece
64	LETSIOS Y.	University of Piraeus, Greece
65	LIBEROPOULOS K.	University of the Aegean, Greece
66	LITINAS Nikos	University of the Aegean, Greece
67	LORETZATOS J.	University of Piraeus, Greece
68	LOUKAKIS E.	Aristotle University of Thessaloniki, Greece
69	MAGINA Chryssanthi	Univeristy of the Aegean, Greece
70	MALKOTZOGLOU A.	University of Piraeus, Greece
71	MALTSINIOTIS K.	INTRACOM
72	MANIADAKIS E.	Hellenic Scientific, Greece
73	MANOLAKOS T.	BSP ABE, Greece
74	MARKOULAKI Effie	University of Piraeus, Greece
75	MAVRAKAKIS Michalis	INTRACOM, Greece
76	MERETAKI Sofia	BRAVO S.A., Greece
77	MERIKAS A.	University of the Aegean, Greece
78	METAXAS N.	Philkeram - Johnson S.A., Greece
79	MIDDLE John	Loughborough University, United Kingdom
80	MOSCHOURIS Socratis	University of Piraeus, Greece
81	NEGAKIS C.	Aristotle University of Thessaloniki, Greece
82	NIKOLAOU N.	University of Piraeus, Greece
83	PANOUSSOPOULOU P.	Univeristy of the Aegean, Greece
84	PANTAZIS P.	BSHG, Greece
85	PAPADAKIS E.	University of Piraeus, Greece
86	PAPADOPOULOS V.	University of Piraeus, Greece
87	PAPADOPOULOU N.	University of Macedonia, Greece
88	PAPOUTSI C.	University of the Aegean, Greece
89	PARIKAKIS George	Univeristy of the Aegean, Greece
90	PASCHALIDIS Apostolos	Ministry of Education, Greece
91	PATSILINAKOS T.	University of Piraeus, Greece
92	PAVLIS N.	University of Piraeus, Greece
93	PAVLOU Eleni	Univeristy of the Aegean, Greece
94	PIRGIANAKI E.	University of the Aegean, Greece
95	PYROMALLIS Spiros	Ministry of Education, Greece
96	RAFTOPOULOS John	Athenian Brewery, Greece
97	REKLITIS E.	University of the Aegean, Greece
98	REVELAKIS M.	Coca Cola HBC, Greece

(*continued*)

Conference Participants: Attendees — Continued

L/I	*Participant Name*	Affiliation
99	ROGDAKI E.	University of the Aegean, Greece
100	SAINI A.	University of Piraeus, Greece
101	SANTRI M.	Neoset, Greece
102	SAPIDIS N.	University of the Aegean, Greece
103	SARANTELIS A.	Athens 2004 Olympic Games, Greece
104	SARDIS Frantzeskos	Univeristy of the Aegean, Greece
105	SIMITZIS Theodoros	Unilever Hellas, Greece
106	SKENTOS Nikos	Univeristy of the Aegean, Greece
107	SOCRATOUS P.	University of Piraeus, Greece
108	SOTIROPOULOU I.	BSHG, Greece
109	SPYROY Thomas	University of the Aegean, Greece
110	STAMATIS Dimitris	Univeristy of the Aegean, Greece
111	STASSIS I.	University of Piraeus, Greece
112	TARTA S.	University of Piraeus, Greece
113	TATARAKIS Antonios	Oracle Hellas A.E., Greece
114	TETOROU P.	University of Piraeus, Greece
115	THEODORAS D.	University of Piraeus, Greece
116	THEODOSIOU K.	University of the Aegean, Greece
117	THILAKERATHNE C.	University of the Aegean, Greece
118	TOPOUZAS A.	Neoset, Greece
119	TRIANTAPHYLLOY Georgios	ELVAL, Greece
120	TRIANTZI Z.	University of Piraeus, Greece
121	TRIMERITI K.	University of the Aegean, Greece
122	TSABAS G.	University of Piraeus, Greece
123	TSAFARAS P.	BSHG, Greece
124	TSAGARIS Christos	University of the Aegean, Greece
125	TSAVDARIDOU Z.	University of the Aegean, Greece
126	TSEKREKOS E.	Ikonomikos Tachidromos Journal, Greece
127	TSIOTRAS Georgios	University of Macedonia, Thessaloniki, Greece
128	TSOULFAS G.	University of Piraeus, Greece
129	TZARATZOUNI N.	University of Piraeus, Greece
130	VALASSIADIS T.	Philkeram - Johnson S.A., Greece
131	VOUTSINAS T.	University of Piraeus, Greece
132	VRYSSAGOTIS Vassilis	Univeristy of the Aegean, Greece
133	XIDIAS D.	University of Piraeus, Greece
134	ZACHAROPOULOU C.	Aristotle University of Thessaloniki, Greece

(*continued*)

Conference Participants: Attendees — Continued

L/I	*Participant Name*	Affiliation
135	ZORZOS Iakovos	Univeristy of the Aegean, Greece
136	ZOTOS E.	University of Piraeus, Greece

E

Simulation Model of a Reliable Production Line

E.1 Description of the Production Line

Consider a production line consisting of four stations S_1, S_2, S_3, S_4 and three intermediate buffers B_2, B_3, B_4 shown in Figure E.1. At each station there are identical machines: 3 machines at station 1, 2 machines at stations 2 and 3, and 3 machines at station 4. The service times at each station are assumed to be exponential distributed with service rates $\mu_i,\ i = 1,2,3,4$ (may be identical or different). Between the stations there are three buffers to reduce the starvation and blocking phenomena. The capacities of the buffers are 4, 2, and 4 slots in buffers $B_{i+1},\ i = 1,2,3$, respectively. Due to the finiteness of the buffers, blocking may occur. On completion of the service at a machine at station i, the job tries to enter the next buffer B_{i+1} or station $i+1$. If the buffer B_{i+1} is full, the job is forced to stay at the machine in station i until a space becomes available at buffer B_{i+1} and then that machine can initiate the service on the next job, when available. The system (PL) is saturated, i.e., in front of the first station there is adequate raw material so that the first station is never starved. Also the last (fourth) station is never blocked.

E.2 The Model of the System

The model of the system has been constructed using Arena 3.0 simulation software. As may be seen from Figure E.1, it contains a collection of modeling constructs (called modules). Particularly there are one Arrive module, four Server modules, three Resource modules, one Simulate module, one Statistics module, one Depart module and three Animate modules.

Each module represents a part of the original system (PL) and gives information about the evolution of the system.

E.2.1 The Arrive module

The Arrive module represents the generation of the entities (jobs) that enter the system. On doubleclicking on the Arrive module the following screen shown in

C. T. Papadopoulos et al., *Analysis and Design of Discrete Part Production Lines*, Springer Optimization and Its Applications,
DOI: 10.1007/978-0-387-89494-2_12,

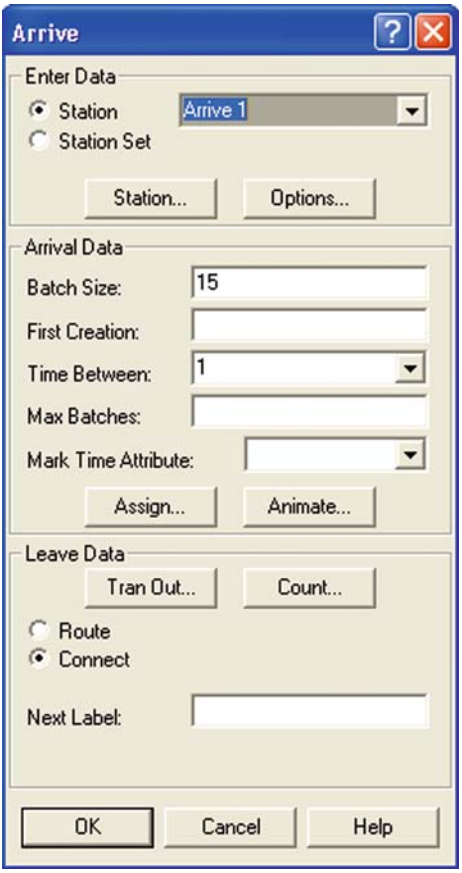


Fig. E.1. A production line with four stations with parallel machines at each station and intermediate buffers

Figure E.2 will appear. The entities (jobs) are generated in batches of 15 jobs every 1 time unit (minute) and enter the queue of station 1. So a large number of jobs are waiting for service in front of station 1 (saturation of station 1).

E.2.2 The Server modules

The Server modules represent the stations of the original system. On doubleclicking on a Server module, e.g., station 1, the screen shown in Figure E.3 will appear.

In the dialog box are the name of the station, the capacity of the station (i.e., the number of identical machines, 3 in this case), information about the distribution of service time (the exponential distribution with mean service time equal to 1 time unit (minute)), and information about the flow of the jobs (connection to the next station). There is also information about blocking. Clicking on the options button of the server 2 (station 2) dialog, the next screen appears as shown in Figure E.4

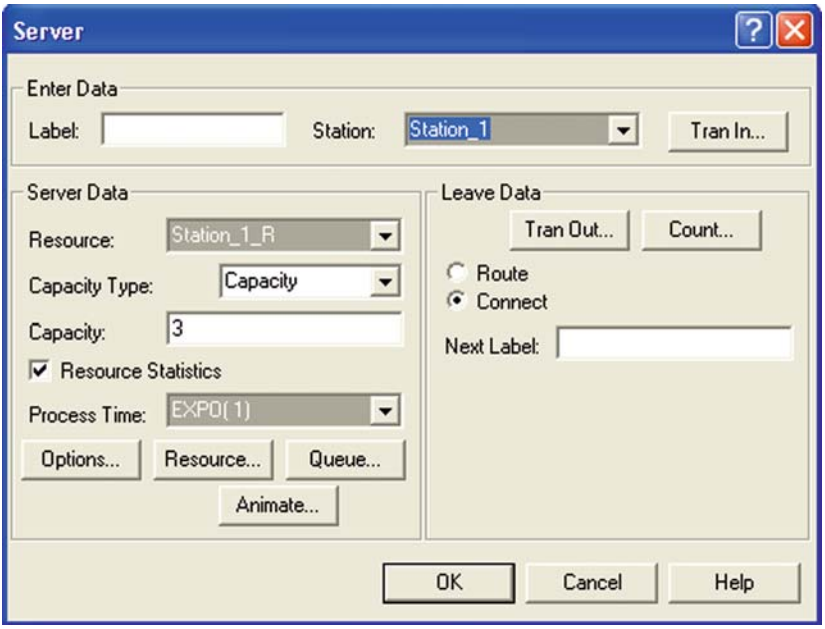


Fig. E.2. The Arrive module dialog box

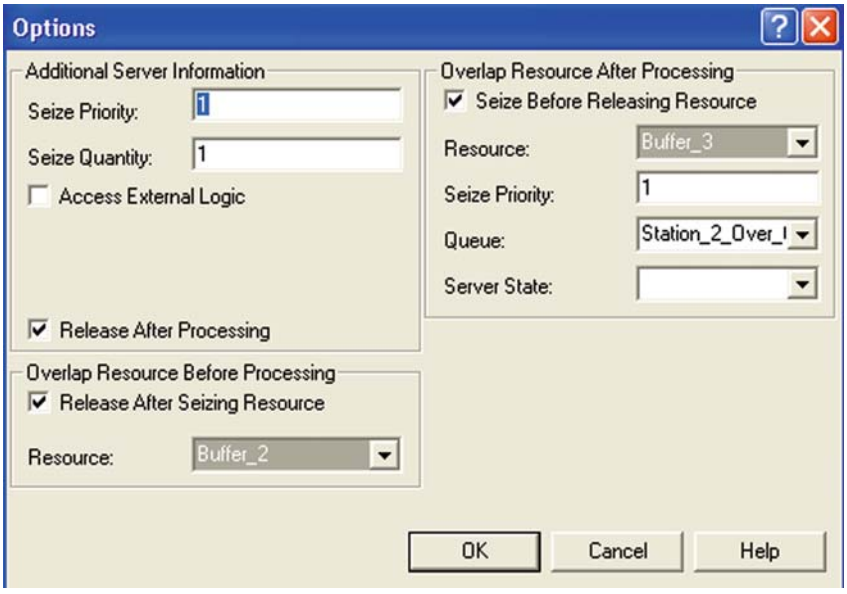


Fig. E.3. The Server module dialog box

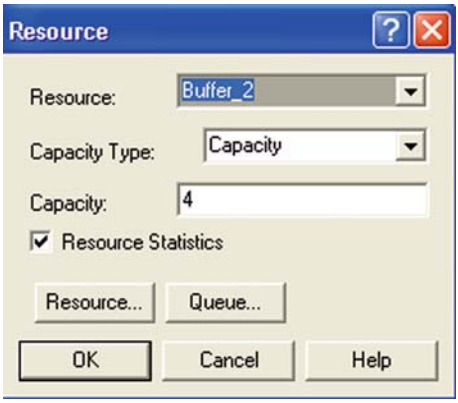


Fig. E.4. The Options dialog box

Here there is information order about the job process before the service of a job at station 2. As one can see at the left bottom corner there is an order to release one unit of buffer B_2 (in front of server 2) as soon as the job seizes a machine at station S_2 (a random selection between the available machines). After completion of service of a job at station 2 there is a request to seize a slot unit at the next buffer (i.e., B_3) before freeing the machine at station S_2. If B_3 is full, the jobs remain at station 2 (blocking after service) until a space becomes free at B_3. Then a machine in S_2 becomes free – a random selection between the blocked machines.

E.2.3 The Resource modules

The Resource modules represent the buffers at the original system. On doubleclicking on a Resource module, the screen shown in Figure E.5 will appear. In the dialog box are the name of the buffer and the capacity of the buffer, i.e., the number of buffer slots.

E.2.4 The Depart module

The Depart module represents jobs leaving the system. On doubleclicking on the Depart module, the screen shown in Figure E.6 will appear.

In the count area of the dialog box, the individual counter button is selected to obtain the total number of jobs that have passed through this module, with counter name: No_of_Jobs. This is what creates the number above the icon for the Depart module (initially at 0) which will clock up as jobs pass through this module.

E.2.5 The Simulate module

The Simulate module does not represent any part of the original system but gives information about the length of simulation time run, the number of replications, the

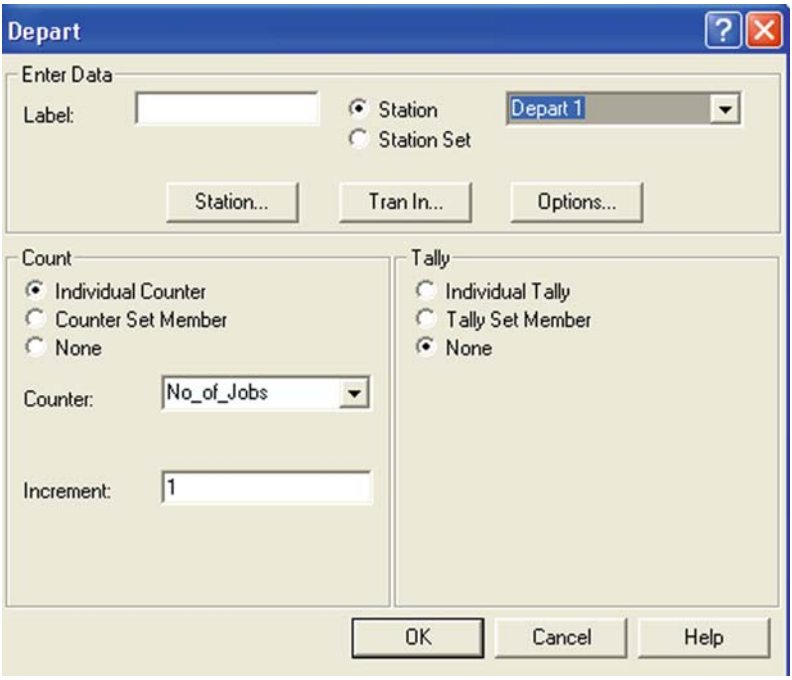


Fig. E.5. The Resource module dialog box

length of warm-up period, etc. In this case, as shown in Figure E.7, the model is to be run for 10 replications each run having a length of 50,000 time units (minutes) and a warm-up period of length 100 minutes. The warm-up period for each run is needed to ensure that the queue in front of station 1 is always full.

Running the model for one replication only by clicking the run button in the run toolbar, and clicking *Yes* on the message to see the results we obtain the numerical summary results listed in Tables E.1 and E.2. The performance measures of the system are listed in Table E.3.

E.2.6 The Statistics module

The Statistics module defines additional statistics to be collected as well as specifying which data are to be saved to files. To create a confidence interval (CI = 95%) for the main performance measure (Throughput), we need to run the model for more than one replication (specifically for 10 replications) and to use the Statistics module to save statistical data. Doubleclicking on the Statistics module, the screen shown in Figure E.8 appears.

On clicking in the Statistics module the Edit button in the counter area, the screen shown in Figure E.9 appears.

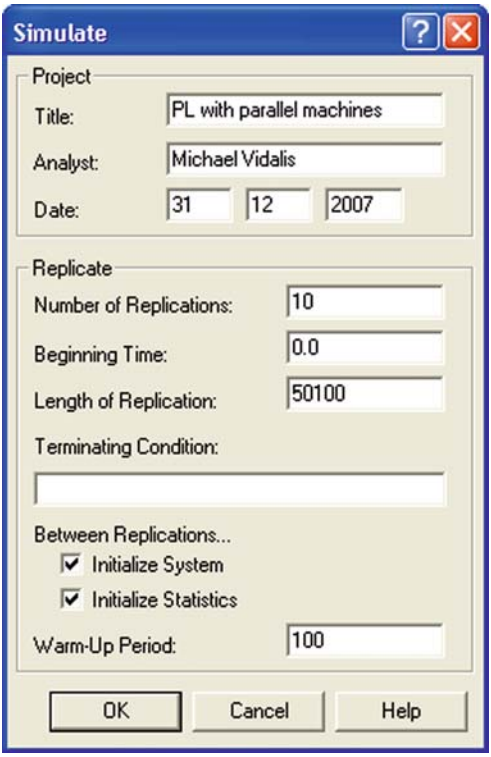


Fig. E.6. The Depart module dialog box

The result from this option is that the value of the counter No_of_Jobs for each replication is saved at Throughput.DAT. This information is used by the Output Analyzer of Arena to create the confidence interval as shown in Figure E.10.

Statistical analysis

Statistical analysis may be used to estimate some characteristic of a large population, too large to enumerate completely. Consider, for example, the throughput of a production line during a particular time period. Under the usual assumptions in regard to the input to the line and the service time of the servers, the throughput is a random variable with a particular but often unknown distribution. A run of a simulation model may be considered to be an experiment involving the taking of a random sample of some variable of interest, in this case the throughput. If the experiment is run a number of times n with throughput values equal to $X_1, X_2, \ldots, X_n$, then

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i, \tag{E.1}$$

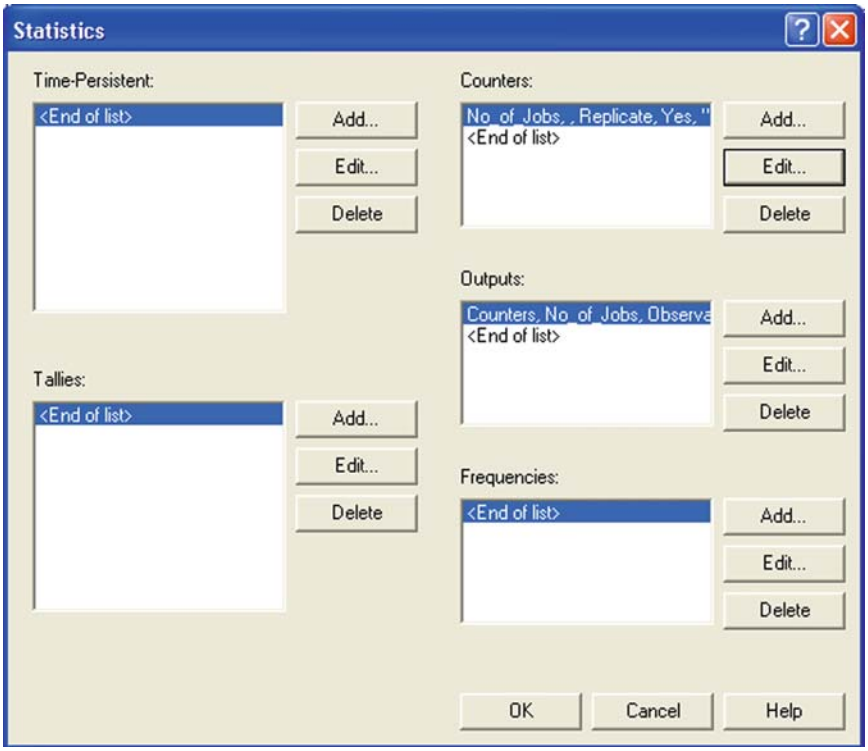


Fig. E.7. The Simulate module

Table E.1. Simulation results: Continuous variables

Identifier	Average	Half Width	Minimum	Maximum	Observations
Station_1_Queue Time	11.555	0.06817	4.7310	22.969	82959
Station_2_Queue Time	2.2720	0.01739	0.0000	9.4101	82959
Station_1_Queue Time	0.60904	0.01023	0.0000	7.5403	82957
Station_1_Queue Time	0.16356	0.00735	0.0000	5.7756	82956

is the sample mean

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2, \tag{E.2}$$

is the sample variance, where the set $X_1, X_2, \ldots, X_n$ is defined as the sample. It is well known that

$$E(\overline{X}) = \mu, \tag{E.3}$$

$$Var(\overline{X}) = \frac{\sigma^2}{n}, \tag{E.4}$$

Table E.2. Simulation results: Discrete variables

Identifier	Average	Half Width	Minimum	Maximum	Final Value
# in Station_1_Queue	19.171	0.00685	13.000	20.000	20.000
# in Station_2_Queue	3.7697	0.01331	0.000	4.000	4.000
# in Station_3_Queue	1.0105	0.02009	0.000	2.000	2.000
# in Station_4_Queue	0.2714	0.01599	0.000	4.000	0.000
Buffer_2 Busy	3.7697	0.01331	0.0000	4.000	4.000
Buffer_3 Busy	1.0105	0.02009	0.0000	2.000	2.000
Buffer_4 Busy	0.27137	0.01599	0.0000	4.000	0.000
Station_1 Busy	3.0000	0.00000	2.0000	3.000	3.000
Station_2 Busy	1.9955	9.6872E-04	0.0000	2.000	2.000
Station_3 Busy	1.6698	0.01114	0.0000	2.000	2.000
Station_4 Busy	1.6588	0.01656	0.0000	3.000	1.000
Station_1 Available	3.00	N/A	3.0000	3.000	3.000
Station_2 Available	2.00	N/A	2.0000	2.000	2.000
Station_3 Available	2.00	N/A	2.0000	2.000	2.000
Station_4 Available	3.00	N/A	3.0000	3.000	3.000
Buffer_4 Available	4.00	N/A	4.0000	4.000	4.000
Buffer_3 Available	2.00	N/A	2.0000	2.000	2.000
Buffer_2 Available	4.00	N/A	4.0000	4.000	4.000

Table E.3. Simulation results: Performance measures

Measure	Value
Throughput	82956 jobs/50000 minutes = 1.65912 jobs/min
System efficiency	1.65912/3 = 55.30%
Utilization of S_1	3.00/3 machines = 100%
Utilization of S_2	1.9955/2 machines = 99.775%
Utilization of S_3	1.6698/2 machines = 83.49%
Utilization of S_4	1.6588/3 machines = 55.293%
Average level of B_2	3.7697 jobs
Average level of B_3	1.0105 jobs
Average level of B_4	0.2714 jobs
Waiting time at B_2	2.2720 minutes
Waiting time at B_3	0.60904 minutes
Waiting time at B_4	0.16356 minutes

where E and Var are the expected value and variance operators, respectively, and μ and σ^2 are the mean and variance of the underlying distribution of X, the throughput. If the underlying distribution of X is normal, $X \sim N(\mu, \sigma)$, then the distribution of $\overline{X}$ is normal, $\overline{X} \sim N(\mu, \sigma/\sqrt{n})$.

$$E(s^2) = \sigma^2 \tag{E.5}$$

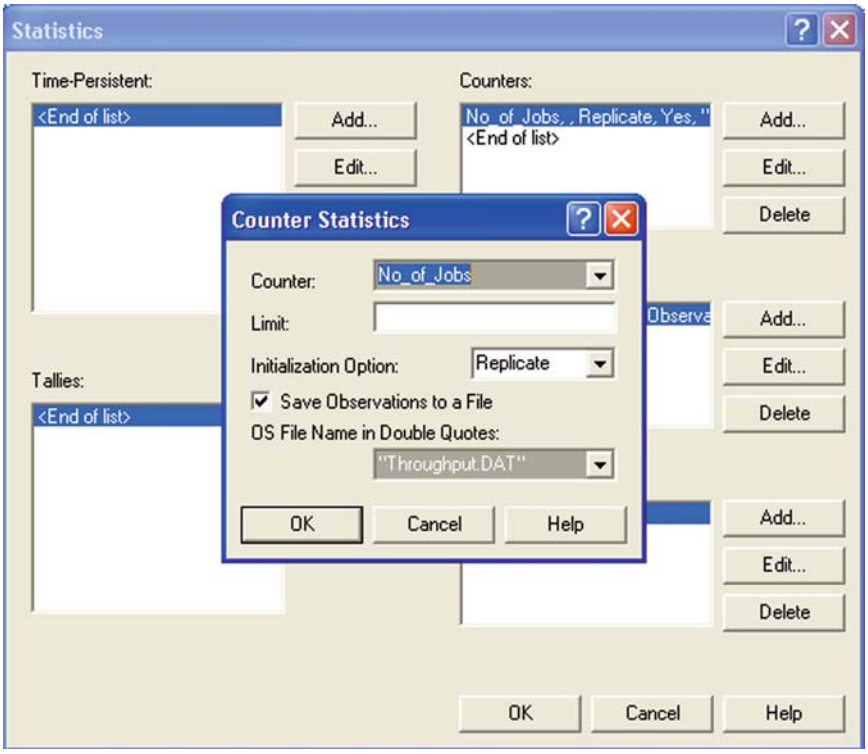


Fig. E.8. The Statistics module dialog box

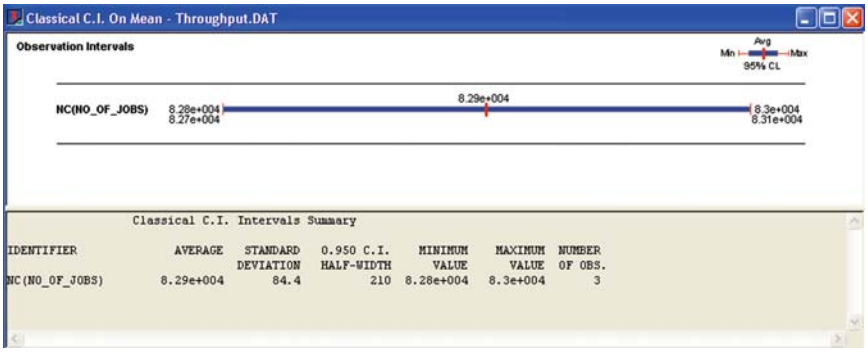


Fig. E.9. Saving the value of counter No_of_Jobs into file Throughput.DAT

where σ^2 is the variance of the underlying distribution. Should the underlying distribution of X be normal, $X \sim N(\mu, \sigma)$, then the statistic $(n-1)s^2/\sigma^2$ has a chi-squared distribution with $(n-1)$ degrees of freedom, denoted by χ^2_{n-1}.

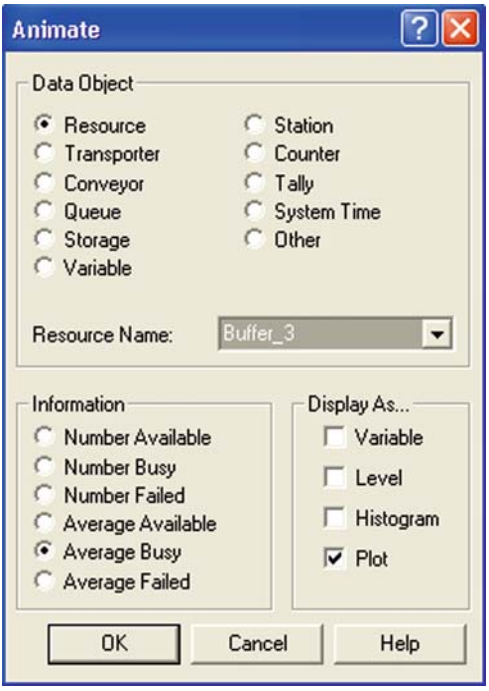


Fig. E.10. The confidence interval (CI = 95%) of throughput

In simulation experiments, it is usually necessary to estimate the parameters of the underlying distributions such as the mean and variance by using the output rate from the experiments. One approach is to use a point estimate to give a single numerical value for the parameter of interest. A statistic acting as a point estimate is said to be unbiased if the expected value of the point estimate is equal to the value of the parameter of interest. Clearly from equations (E.1), (E.2), (E.3) and (E.4) above, $\overline{X}$ and s^2 are unbiased estimates of the mean and variance, respectively, of the underlying distribution of X, the throughput. The variance of an estimator is an important characteristic in that, for example, if two unbiased estimators were available, the estimator with the lower variance is preferable in that the point estimate using that estimator is more likely to be closer to the parameter being estimated. The lower-variance estimator is said to be more 'efficient.' A further and related property is the concept of the 'consistency' of an estimator. Basically, the idea is that if an estimator is consistent, as the number in the sample, n, increases the estimator improves in some sense, for example, the variance becomes smaller.

In simulation experiments, good point estimates of parameters are generally available. However, some quantitative information about the variance of the estimator used is required if the analyst is to have confidence in the estimate. A more formal approach is to develop a 'confidence interval' for the parameter of interest. The objective of determining a confidence interval is to form an interval with

specified end points that will contain the parameter of interest with a pre-specified probability level. Clearly, the parameter of the underlying distribution has a fixed value, i.e., it is not itself a random variable, but the estimate used is a random variable. To illustrate the development of a confidence interval, consider the following simple example.

Assume $X \sim N(\mu, \sigma)$ with σ known and μ fixed but unknown, given $\overline{X} = \sum_{i=1}^{n}$ where $X_i,\ i = 1, 2, \ldots, n$ is a set of sample values of X. Now $\overline{X} \sim N(\mu, \sigma/\sqrt{n})$. Therefore,

$$Prob\left(-z_{\alpha/2} \leq \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} \leq z_{\alpha/2}\right) = (1 - \alpha),$$

where $z_{\alpha/2}$ is obtained from the standard normal tables, $Z \sim N(0, 1)$ and

$$Prob\left(-z_{\alpha/2} \leq Z \leq z_{\alpha/2}\right) = (1 - \alpha).$$

From

$$Prob\left(-z_{\alpha/2} \leq \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} \leq z_{\alpha/2}\right) = (1 - \alpha),$$

by algebraic manipulation:

$$Prob\left(\overline{X} - \frac{\sigma}{\sqrt{n}} z_{\alpha/2} \leq \mu \leq \overline{X} + \frac{\sigma}{\sqrt{n}} z_{\alpha/2}\right) = (1 - \alpha).$$

Thus the interval $\overline{X} - \frac{\sigma}{\sqrt{n}} z_{\alpha/2}$ to $\overline{X} + \frac{\sigma}{\sqrt{n}} z_{\alpha/2}$ is said to be the $(1 - \alpha)$ confidence interval for μ.

There were two constraints in the development of the above confidence interval for μ, the mean of the underlying distribution, viz., X was distributed according to a normal distribution and σ^2, the variance of X was known. Generally speaking, if the sample size is large, say $n \geq 30$, the central limit theorem may be used to assume that $\overline{X}$ follows a normal distribution.

As a working rule, if the number of sample, n, is 30 or more and even if the underlying distribution of X is unknown and σ, the variance of the underlying distribution is unknown, the following is a $(1 - \alpha)$ confidence interval for the mean, μ, of the underlying distribution:

$$\overline{X} \pm z_{\alpha/2} \frac{\sigma}{\sqrt{n}}; \quad \sigma^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \overline{X})^2$$

where $z_{\alpha/2}$ is obtained from $N(0, 1)$ tables as indicated above. However, if n, the sample size is less than 30 and the underlying distribution of X can be assumed to be normal but the value of σ, the variance of the underlying distribution is unknown, the following $(1 - \alpha)$ confidence interval for μ obtains:

$$\overline{X} \pm t_{n-1,\alpha/2} \frac{\sigma}{\sqrt{n}}; \quad \sigma^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \overline{X})^2$$

where $t_{n-1,\alpha/2}$ is the value from the t distribution providing an area of $\alpha/2$ in the upper tail of the t distribution, with $n-1$ degrees of freedom.

It should be noted that the term 'Half Width' in the Arena tables refers to half the confidence interval, e.g., $t_{n-1,\alpha/2}$ in the third case discussed above. In reference to the simulation experiments discussed above, Figure E.10 gives the confidence interval $(CI = 95\%; \alpha = 0.05)$ of the mean value of the throughput.

The average of 10 observations of the throughput is given in Figure E.10 as $8.32e+004$ or 83200 units, s, the sample standard deviation being 310. As n, the sample size is less than 30 and σ^2, the variance of the throughput is unknown, the following 95% confidence interval for the mean of the throughput applies, provided it may be assumed that the underlying distribution of the throughput is normal:

$$\overline{X} \pm t_{9,0.025} \frac{s}{\sqrt{n}}$$

$t_{9,0.025} = 2.26$ from t-tables.

Thus the confidence interval for the mean value of the throughput is

$$83200 \pm 2.26 \left(\frac{310}{\sqrt{10}} \right) = 83200 \pm 222.$$

Note that Figure E.10 gives a 'Half Width' value of 222. The information on 'Half Width' in the Arena dialogs may be used to give what might be described as 'reasonable' upper and lower bounds for the throughput. It must be noted that some round-off error is present in the Arena calculations. For example, the number of observations in the queue time tableau (Table E.1) of the first replication of the above model is given to five significant figures, whereas in the confidence interval dialog (Figure E.10) the throughput is given to three significant figures only. Using the data given in Figure E.10, the following range of values of throughput per minute may be obtained:

Upper and lower limits (95% confidence interval):

Upper Limit: $= 83400/50000 = 1.668$ jobs/minute.
Lower Limit: $= 82900/50000 = 1.658$ jobs/minute.
Mean Value: $= 83200/50000 = 1.664$ jobs/minute.

It might be noted that because of the round-off error procedures in Arena the mean value of the distribution in this case is not exactly equal to the average of the upper limit and lower limit values.

E.2.7 The Animate modules

The Animate modules give temporary information (dynamic plots during the simulation running) of selected parts of the system. Here the Animate module gives information about the average level of buffer B_3. In Figure E.11, a snapshot of the average level of buffer B_3 up to a specific time (time = 1000 minutes) is shown. One can see from this picture that the average level of buffer B_3 is 1 job.

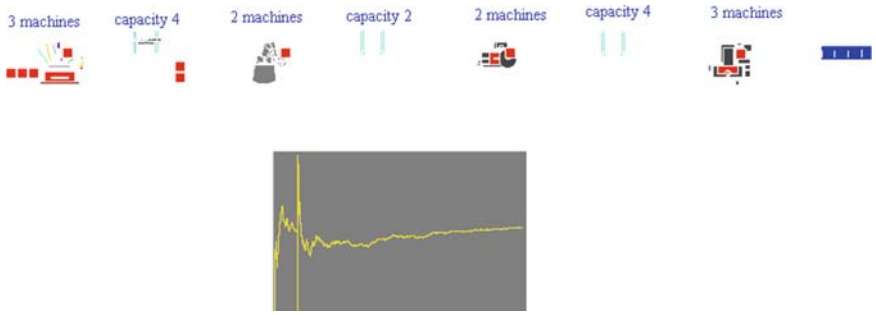


Fig. E.11. A snapshot of the evolution of the average level of buffer B_3 up to time equal to 1000 minutes in a production line with 4 stations with parallel machines at each station and intermediate buffers

References

1. Kelton, W.D., Sadowski, R.P., and Sadowski, D.A. (1998), *Simulation with Arena*, McGraw-Hill.
2. Anderson, D.R., Sweeney, D.J., and Williams, T.A. (1997), *Statistics for Business and Economics*, Fourth Edition, West Publishing Company.

Subject Index

Author Index