IV.E
Planning and Control

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

CHAPTER 60

Queueing Models of Manufacturing and Service Systems

JOHN A. BUZACOTT
York University

J. GEORGE SHANTHIKUMAR
University of California at Berkeley

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

1. INTRODUCTION

The design and improvement of the performance of manufacturing and service systems requires that we have efficient ways by which we can (1) predict the performance of the systems and (2) identify the effects of key design parameters on the system performance. Manufacturing and service systems have to cope with a wide range of variability, uncertainty, and disturbances. Different customers require different tasks to be performed, people and machines can vary in their time to perform standardized tasks, machines can break down unexpectedly, repair can prove more complicated than anticipated. So we need approaches to predicting the performance that take into account this uncertainty and variability and also help us reduce their adverse impacts. Queueing models are particularly useful in describing variability and predicting its impact on performance. This chapter contains an overview of the key models that are relevant to analysis of manufacturing and service systems. The focus of these models is on predicting throughput, inventory levels, queue lengths, and service levels after allowing for disturbances such as machine breakdowns, human operator performance variability, and quality problems.

Queueing models can be used at the system-design stage to rapidly explore alternatives and see the sensitivity to parameter values. The models can also be of great value in assessing the performance of systems once they are installed because they enable the sources of loss of productivity to be

identified. Models provide understanding and insights of complex production systems and enable one to obtain answers to a variety of ''what-if'' questions with little effort.

Our focus in this chapter will be on manufacturing and service systems where each job or customer is distinct. Most service systems have to deal with the requirements of individual customers. In manufacturing, each job is distinct in the mechanical, electrical, and electronics industries making products such as cars, refrigerators, electric generators, and computers. Systems that process fluids, like those found in the chemical and metallurgical industries, will not be considered, although sometimes in these industries fluids are processed in distinct batches or packets and, if such a packet is taken as the unit of manufacture, then the system can be considered to process discrete jobs. For simplicity, we will call an item, part, subassembly, or assembly processed by a machine or workstation in manufacturing a *job.* In service applications, we will call each order or person a *customer.*

While each job or customer is distinct, different jobs or customers can be in all respects identical and in particular have the same processing requirements. If the system only processes one type of job or customer, then some aspects of its design and operation will be simplified because all jobs or customers can then be handled the same way. However, if the system processes many different types of jobs or customers, instructions for each type will be required and control will tend to be more complex. Particularly in service systems, it is often not known what the processing requirements of a customer will be until after the customer arrives and some diagnosis can be carried out. Even in manufacturing, quality problems can result in the processing requirements of a job changing after processing has begun. Models have to be able to represent this evolution of knowledge about processing requirements and how the information is used to modify instructions on what has to be done.

1.1. Models

Traditionally, manufacturing and service system designers relied on experience and rules of thumb in order to identify the effect of design parameters on the performance. However, the increasing cost and complexity of modern systems, and the often lengthy time required to bring them up to their designed performance targets, have resulted in designers using formal models of the system to assess performance.

1.1.1. Basic Approach to Modeling

The process of modeling involves the following steps:

1. *Identifying the issues to be addressed:* Ascertain the needs of the user. What decisions is the model required to support? These can range from the very specific, such as how large a specific storage location should be, to the somewhat vague, such as whether it is possible to identify when a particular way of operating the system is optimal.
2. *Learning about the system:* Identify the components of the system, such as the people, machines, material handling, storage and the data collection and control system. Determine the characteristics of the jobs or customers and the target volumes, quality, and cost. This step can involve close contact with the system designers and a review of any existing models to identify their capability and their shortcomings.
3. *Choosing a modeling approach:* Various types of modeling approaches can be used, ranging from formal mathematical models through computer simulations to the development of a ''toy'' system in which toy parts or people physically move from one toy machine to another. The choice of modeling approach is determined by the time and cost budget for model development and the anticipated way in which the model will be used.
4. *Developing and testing the model:* This step requires obtaining data on the parameters of the model, and often the lack of desirable data forces the model to be substantially simplified.
5. *Developing a model interface for the user:* If the model is to be of value in making decisions, it has to be provided with some interface so that it can be used by managers. This requires the modeler either to embody the model in a decision support system or to present the model and its implications in a way that managers can understand.
6. *Verifying and validating the model:* The model has to be checked to see that it is a reasonably correct representation of the reality it seeks to represent. Verification is the process of ensuring that the model results are correct for the assumptions made in developing the model. Validation is the process of ensuring that the model is an accurate representation of the real system. This may also involve convincing the user that the model is adequate for the decision he or she requires it to support.
7. *Experimenting with the model:* This requires exploring the impact of changes in model parameters and developing understanding of the factors influencing performance of the system so that the manager can be confident in the decisions made using the model.
8. *Presenting the results:* Using the model, the manager should come up with a recommended course of action. This recommendation may have to be presented to higher-level management

and the role of the model in aiding the decision explained. Alternatively, the model and the results of its use will be presented in a report or paper that, apart from describing the model itself, should explain what the model can do, what it cannot do, and how accurate are its predictions.

1.1.2. Types of Models

For systems processing discrete jobs or customers, there are three types of models in common use:

Physical models represent the real system by another physical system, in which jobs or customers move from one machine or service center to another and the machines or service centers perform processing operations on the jobs or customers. The major difference to the real system is that the model uses a different dimensional scale, so a large system will occupy a table top. Physical models can use toy-sized components, but they can be provided with a control system that employs the same logic as the real system. Physical models are excellent as a means of educating management and workers about the control of the system, but they do not lend themselves for assessing the long-run behavior of the system, as it is difficult to represent the statistical properties of events such as machine failures or worker absenteeism.

Simulation models represent the events that could occur as a system operates by a sequence of steps in a computer program (see Chapters 93–96). This means that the logical relationships that exist between events can be described in detail. The probabilistic nature of many events, such as machine failure, can be represented by sampling from a distribution representing the pattern of occurrence of the event, for example, the distribution of the time between machine failures. Thus, in order to represent the typical behavior of the system, it is necessary to run the simulation model for a sufficiently long time that all events can occur a reasonable number of times. Simulation models can be provided with an interactive graphic display to demonstrate the movement of jobs or customers. This can be of great value in communicating the assumptions of the model to engineers and others.

Analytical models describe the system using mathematical or symbolic relationships. These are then used to derive a formula or define an algorithm or computational procedure by which the performance measures of the system can be calculated. Analytical models can also be used to demonstrate properties of various operating rules and control strategies. Sometimes it is not possible, within a reasonable amount of computer time or space, to obtain the performance measure from the relationships describing the system without making further assumptions that modify these relationships. The resulting model is thus approximate rather than exact. Testing the approximation may then require a simulation model, so approximate models are useful only if they are easy to use and provide insight into what determines the system behavior.

1.1.3. Why Model?

Models can be developed for a variety of reasons, in particular:

- *Understanding:* The model is used in order to explain why and how. The model is intended to convey insight. Sometimes the model just indicates the direction of influence of some variable on performance, that is, as the variable increases in value does performance improve? Alternatively, the model can be quite complex but with the major function of explaining why the system behaves in certain ways.
- *Learning:* As well as providing insight, a model may be intended to teach managers or workers about the factors that determine performance. The model may omit many features of the real system and focus on those aspects that are considered crucial for those people responsible for effective operation of the system.
- *Improvement:* The model is used to improve system design and operation. Changes in parameters and rules can be explored, and factors critical for achieving performance targets can be identified. To make sure that conclusions drawn from the model will apply to the real system, such models pay particular emphasis to the adequacy with which they describe the system behavior.
- *Optimization models:* Given a model that predicts performance as a function of various parameters, an optimization model determines the optimal combination of these parameters. This usually means that the optimization problem is formulated as a mathematical programming problem, generally with a mixture of integer and continuous variables.
- *Decision making:* The model is to be used to aid decisions about either the design or operation of the system. The model has to be able to discriminate the effects of different courses of action and project their impact over time.

1.1.4. Requirements of Models

Models are based on certain assumptions about the system and its components and how the system is going to be operated. Then there are the assumptions about the nature of the disturbances that will

impact the system and the range of responses to these disturbances. The hierarchy of control and the flow of information also have to be represented in the model.

- *Complexity vs. simplicity:* Modeling involves compromises in deciding how much detail to represent. A large amount of detail means that the model should be a more precise representation of reality, but the disadvantages are that the model will be more difficult to verify and validate, be harder for users to understand, and take longer to develop. A simple model may not represent the system adequately and thus may give inaccurate predictions and omit key decisions or useful responses to disturbances.
- *Flexibility:* A model may be used to support decision making as the system evolves over time. That is, from initial concept through planning, detailed design, installation, and operation, there is a need for models to support decisions. While no one single model will support all decisions, it is desirable for a model to be useful at a number of different stages in system evolution. This means that the model should permit changes in the system modeled. Some of these changes may relate to the structure of the system, such as the number of machines or service representatives, the way in which jobs or customers move through the system, or the way in which the control hierarchy is set up. Other changes may relate to the values of parameters such as the frequency of machine failures or the demand rate. A model or a modeling approach has to be evaluated with respect to the ease of making both these sorts of changes.
- *Data requirements:* While there is often a great deal of data available in manufacturing or service, it is rare that the data are in the form required by the model. At the planning stage, there is often doubt as to the applicability of data collected from different systems in different environments, while when the system is operational, the data may well only apply over a limited range of operating conditions. Thus, a model should use the least amount of data required in order to make adequate predictions, and an important component of the validation of a model is assessing the sensitivity of the model to errors in the data.
- *Transparency:* Since the model has to be accepted by its users, it is desirable that the assumptions and procedures used in the model be reasonably transparent to others beside the model developer. The developer should be able to convince the user that the model is a reasonably accurate representation of reality.
- *Efficiency:* Models can consume significant resources, both in their development and in their use. Modeling approaches differ in their requirements on the knowledge, skill, and elapsed time required for development. Since most models will be implemented on a computer, such issues as running time and storage requirements can also be important.
- *User interface:* If a model is going to be of real value, it should be usable by managers rather than only by the model developer. A user interface is essential in order to guide the user in the correct use of the model, ensuring that it is clear what data should be provided and avoiding any ambiguity in interpreting the results.

1.2. Queueing Models

Queueing models are particularly useful for determining the following performance measures of a system:

- *Capacity or throughput:* This is the maximum rate at which the system can accept jobs or customers over some long time interval. We will use the symbol TH to denote it, and it is usually measured in jobs or customers per hour (or some other suitable time interval). Individual components of the system will, of course, have a higher short-term capacity than TH, but over the long run they will lose capacity because of machine breakdowns or worker absences. Capacity will also be lost because of interaction between the different parts of the system. For example, too few pallets will reduce work flow through a machine.
- *Flow Time or Lead Time:* The flow time, sometimes called the lead time in manufacturing, is the time from when a job or customer arrives at the system until the job or customer departs the system. It will be greater than the actual processing time because jobs are held in inventory buffers and customers wait in queues. Queueing models usually focus on determining the average flow time, but it is usually also possible to determine the variance and higher moments.
- *Inventories and queue lengths:* It is often important to know where inventories and queues are distributed through the system. The average total flow time and the average total queue length are connected by Little's law:

$$\bar{l} = \lambda \bar{w}$$

where $\bar{l}$ is the average queue length, $\bar{w}$ is the average flow time, and λ is the average rate at

which jobs or customers flow through the system. So given either queue length or flow time, the other performance measure can be readily determined.

- *Service level:* The service level can be measured in a variety of different ways, such as the fraction of demands met immediately or the average time to fill a customer demand. Service level is a particularly important performance measure when finished product inventories are kept and it is necessary to trade off the inventory investment with the penalties of delay in meeting customer demand.

1.2.1. Using Queueing Models

Queueing models are particularly useful in designing and improving system performance. In particular, they are of great value for addressing the following issues:

1. *Investigating alternative configurations:* There are usually alternative ways of allocating tasks to machines or people, and each alternative will result in different patterns of work flow. Queueing models are particularly valuable in rapidly exploring a wide range of alternatives and seeing how system performance is modified.
2. *Exploring the impact of parameters set by management:* Typically, management have to choose values for such parameters as inventory buffer capacities, the number of kanbans, or base stock levels. Queueing models enable their impact on performance measures such as throughput or service level to be found. If costs are available, then it is possible to determine the values that optimize performance.
3. *Comparing alternative scheduling and work-allocation rules:* Queueing models can be used to compare different scheduling rules. They can also be used to explore the way in which performance is changed as more information about jobs or customers is acquired and that information used to modify the routing and allocation of jobs or customers to machines or servers.
4. *Understanding the impact of variability:* Typically, less variability improves performance. But since reducing variability can be costly, it is desirable to know by how much performance is improved. Variability reduction is typically the aim of quality management efforts, and queueing models can help focus that effort.

1.3. Modeling Manufacturing and Service Systems

Queueing theory is described in Chapter 83. To apply it to modeling manufacturing or service systems, it is necessary to develop an understanding of the key features of the system and how these translate into queueing models. Considerable expertise is necessary. Often, as the system becomes more complex, it is necessary to develop approximations, and these have to be tested by extensive simulations. So a number of software packages have been developed that incorporate tested queueing models and approximations. These packages have user interfaces that guide the user through the selection of the appropriate model and allow the user to specify parameters values.

Alternatively, most queueing models can be easily incorporated into spreadsheets because they are either formulas or fairly straightforward iterative calculations.

Before giving the details of typical models, it is useful to describe briefly how the main features of various manufacturing and service systems are viewed as queues.

1.3.1. Manufacturing Systems

1.3.1.1. Job Shops and Flow Lines The two traditional forms of organizing manufacturing systems are the job shop and the flow line. The *job shop* consists of a variety of different types of machines, some of which can perform operations on different types of jobs, although this may require some setup or changeover time between job types. Material handling is such that different types of jobs can visit machines in different sequences. In a queueing model, the job shop is viewed as a network of queues, with each machine or machine center regarded as a server. Job routing determines the movement of jobs between the different servers. Rather than tracking individual jobs in the queueing model, it is usual to represent all the different jobs by a small number of job types or classes. By contrast, the *flow line* requires all jobs to visit machines and work centers in the same sequence, thus simplifying material handling. The simple material handling, combined with the standard routing, makes it easier to control work flow and instruct machines and workers on their tasks and thus enables high volumes to be produced economically. The queueing model thus consists of a number of queues in series. However, buffer space between machines is often limited, resulting in blocking or starving of machines, and this has to be represented by the model.

1.3.1.2. Transfer Lines Many flow lines produce only a single type of product, and with increasing volume it becomes attractive to automate individual machines and replace human operators by automatic devices and machines. The *automatic transfer line* goes one step further. Not only are

the machines and the material handling from one machine to the next automated, but all machines are linked so that they begin their tasks simultaneously and thus material movement is synchronized. In this way, the number of jobs in process can be kept small and extremely high productivity is potentially possible. If any machine should fail, then the tight linkage means that the whole line stops. To overcome this, the line may be divided up into sections, with buffers between the sections. The queueing model has to capture this linkage and the impact of buffers.

1.3.1.3. Flexible Transfer Lines Initially, when automatic transfer lines came into widespread use in the 1950s, the instructions on how to perform the tasks at a machine were embodied in physical information storage devices such as cams, jigs, and other fixtures. This has been called ''hard automation,'' and obviously it made it difficult to change instructions. However, as electronic devices for storage and processing of information were developed, combined with numerical control (i.e., transducers that convert digital information into physical motion of tools), changing the instructions on performing tasks became simpler. This resulted in the development of automated *flexible transfer lines* and *flexible flow lines,* distinguished by whether synchronized movement of jobs was retained or not. If job movement is not synchronized, then it is essential to provide some storage space between machines, otherwise job movement will always be determined by the slowest machine. Flexible transfer lines and flexible flow lines can manufacture more variants of a product than traditional automatic transfer lines, but unless the variation between jobs is very small, there may still be a requirement to change tools between jobs. The queueing model now has to describe the movement of the different job types through the system.

1.3.1.4. Flexible Manufacturing Systems Transfer lines are essentially automated flow lines, and thus all jobs have to visit all the machines in the same sequence. If numerically controlled machines are combined with a material-handling system that enables different jobs to visit machines in different sequences, then the resulting system is known as a *flexible manufacturing system* (FMS). Such systems were first implemented in the 1970s for machining tasks, so FMS sometimes implies a flexible machining system. An FMS is essentially an automated job shop. It can produce a reasonable range of products, although initially this range was limited by the ability to store or deliver the required tools to individual machines. Usually parts are mounted on pallets and the number of pallets is limited, so the queueing model represents the system as a network of queues in which the total number of customers is limited by the number of pallets.

1.3.1.5. Flexible Assembly Systems Assembly tasks are difficult to automate economically unless the tasks have unusual characteristics such as size, weight, or temperature or chemical or radiation hazards. However, beginning in the clothing and shoe industries and then spreading to the electronic industry in the early 1980s, it was recognized that *flexible assembly systems* with automated job movement to assembly, inspection, and test stations, linked to automated job-identification systems, resulted in significant improvement in work flow and control in assembly systems, producing a variety of different jobs, even though many individual tasks are still performed by human operators. Some flexible assembly systems enable jobs to move between any pair of workstations, while others, such as those introduced in the mid-1980s by the automobile industry to replace the traditional assembly line, have a generally series structure but with paralleling of workstations and some feedback loops so that jobs can be readily reprocessed if they do not meet required quality standards. Again, the number of work carriers may be limited, and there may be requirements to maintain sequence of jobs as they go through parallel work stations. Flexible assembly systems sometimes store all work in progress centrally and assign it to work stations as they become available. Alternatively, jobs may circulate through the system on conveyors. Queueing models have to be able to represent job circulation, storage, and processing.

1.3.2. Service Systems

As yet, there is no established approach for categorizing the different configurations of service systems, although there are a variety of approaches proposed (Schmenner 1986, Silvestro et al. 1992). From the perspective of developing queueing models, perhaps the most useful approach is to focus on the range and features of the tasks assigned to the people providing service (Buzacott 2000).

1.3.2.1. Narrow-Range Tasks If the system is configured so that each individual server only performs a narrow range of tasks, then this usually means that a number of different servers are required in order to perform all the required tasks for a customer. This means that the system will often be similar to a manufacturing flow line and so can be represented by queues in series. Of course, if there is some flexibility in the sequence in which tasks are done, then the system becomes more like a manufacturing job shop and can be represented by a network of queues.

1.3.2.2. Broad-Range Tasks Alternatively, individual servers can be assigned a broad range of tasks. This means that they are often able to do all the tasks required by a specific customer. However, in order to cope with increasing volume of customers, many servers are required. The system can

then be represented as a number of servers in parallel. There are a variety of ways in which customers can be allocated to servers; for example, some servers may specialize on particular types of customers; alternatively, customers can be allocated to servers according to some allocation rule, such as allocate to free servers in order of customer arrivals or allocate according to a round-robin or cyclical rule (first arrival, to server 1, next to server 2, next to server 1, next to server 2, and so on).

1.3.2.3. Specialized Diagnosis In many service situations the tasks to be done on or for a customer are not known until the customer arrives and some diagnosis is carried out. As a result, it is sometimes desirable to have a first stage of service that determines the service tasks required and then allocates customers to specialized service providers capable of performing the required tasks. The appropriate queueing model now consists of a network of queues in which customer flow out of diagnosis to a specific specialized facility is random, with probability equal to the frequency of the facility being required by the diagnosis. In some situations, multiple diagnostic steps are required, depending on whether a step is able to determine the service requirements.

1.3.3. Supply Chains and Logistic Systems

A developing field for applying queueing models is in determining service levels and inventories in supply chains. For modeling purposes, the supply chain is viewed as a network of cells, where in each cell manufacturing, transport, inspection, and other functions are performed. Cells are connected by material flow from raw material through parts manufacture through assembly, test, and distribution through warehouses and retail outlets to customers. However, cells are also connected by information flow from customer orders through retailer orders, warehouse shipment instructions, production schedules, part requisitions, and raw material requests. Because material flow occurs only as a result of requisitions and orders, it is necessary to include in the queueing model not only the representation of material flow but also the representation of information flow and how the two interact. Certain types of coordination schemes for controlling work flow use advance information—for example, forecasts of receipts of customer orders are used in Material Requirements Planning—and it is then necessary to represent this in the model. We will illustrate the combined modeling of material flow and information flow by some simple models.

1.4. Description of Available Models

The remainder of this chapter will be devoted to a description of some of the available analytical models of manufacturing and service systems and supply chains. The discussion will emphasize analytical models that yield a formula-type result, as such results are most readily implementable. When no formula exists, we will outline the general principles of the computational algorithms. We will also illustrate a number of approximate approaches for determining performance.

1.4.1. Assumptions of Models

Every model is based on certain assumptions about the composition of the system, the work flow through it, and the availability of resources, such as the number of operators, the number of repair crews, and the way in which these resources are allocated when there are competing demands for them. These assumptions describe the essential features of the system. Then there are further assumptions that relate to the nature of disturbances to the system operation, such as the distribution of time between successive failures of a machine, the distribution of time to repair a machine, the distribution of time to perform a task, and the pattern of occurrence of defective parts. In some models, the results depend critically on these distributions, while in others the mean of the distribution is all that is required. Usually it is easier to derive results for certain distributions than for others; in particular, it is possible to derive explicit formulas for the performance of some systems when the associated distributions are exponential or geometric, but not for other distributions. Fortunately, in many cases the results are not particularly sensitive to the form of the distribution or, at least, they give the right general shape of the relationship of the system design parameters to the system performance measures. Thus, the results can be used to give general insight. Almost all analytical models make the following assumptions:

1. Distributions are stationary (i.e., process parameters do not change with time or cumulative output).
2. Successive events of a particular type occur at intervals that are independent of each other (i.e., there is no serial correlation).
3. Events at one machine or service center are independent of events at another machine.

These assumptions are often not strictly correct in reality. However, it would be difficult to derive results that would relax them. In most real systems, when the assumptions fail, the system would be

considered out of control and managerial action would be taken to restore control (e.g., a significant worsening in quality with time would demand managerial intervention).

2. SINGLE-STAGE SYSTEMS

In a single-stage system all work required by a job or a customer is done by one service facility. However, this facility may consist of a number of servers or machines in parallel, and system performance will depend on how jobs or customers are allocated to the machines or servers.

We distinguish between two situations. One is that where no work is done prior to a job or customer's arrival. The second is where jobs are done prior to arrival of a demand, or work is done on anticipated customer requirements prior to the customer arriving.

2.1. Make-to-Order Manufacturing or Service with No Tasks Done Prior to Customer Arrival

2.1.1. Single-Server System

The system is then equivalent to a single-server queueing system for which numerous results are available (see sections 4.2 to 4.5 of Chapter 83). With Poisson arrivals, the performance measures are usually obtainable from fairly simple formulae. However, for general arrival times and general service times, it is usually necessary to resort to bounds and approximations, although such results are often remarkably accurate when the server utilization is reasonably high (see Daley et al. (1992) for a comprehensive description of various bounds and approximations). Suppose that the mean time between arrivals is $1/\lambda$ and the squared coefficient of variation (scv = variance/mean^2) of the time between arrivals is C_a^2. The service time of a job has mean $1/\mu$ and scv of C_S^2. Let $\rho = \lambda/\mu$. Then a good general upper bound on the number of jobs in the system is

$$E[N] \leq \frac{\rho(2 - \rho)C_a^2 + \rho^2 C_S^2}{2(1 - \rho)} + \rho \tag{1}$$

while for interarrival times that are DMRL, i.e., decreasing mean residual life, or in other words, as the time since the last arrival increases, the expected time to the next arrival becomes less, a lower bound is

$$\frac{\rho C_a^2 - \rho(1 - \rho) + \rho^2 C_S^2}{2(1 - \rho)} + \rho \leq E[N] \tag{2}$$

Note that the difference between these upper and lower bounds is $\rho(C_a^2 + 1)/2 < \rho$, since with DMRL arrivals $C_a^2 < 1$.

An approximation for the mean number of jobs in the system, $\hat{n}$, that gives good results for relatively high utilizations is the following:

$$\hat{n} = \left\{\frac{\rho^2(1 + C_S^2)}{1 + \rho^2 C_S^2}\right\}\left\{\frac{(C_a^2 + \rho^2 C_S^2)}{2(1 - \rho)}\right\} + \rho \tag{3}$$

At low utilizations, none of the approximations or bounds that just use information on the mean and variance of the interarrival times and service times are particularly good when the criterion is the percentage error. This is because the performance of the queue tends to be determined by the burstiness of arrivals, a property not captured by the second moment.

For some situations, where all that is required is to get insight into the impact of variability in interarrival times and service times at high utilizations, it is possible to use the heavy traffic result

$$\lim_{\rho \to 1} 2(1 - \rho)E[N] = C_a^2 + C_S^2 \tag{4}$$

2.1.2. Multiple Servers

2.1.2.1. Identical Servers Suppose there are c parallel servers with identical capabilities. The mean time between arrival of jobs is $1/\lambda$ and the mean time to serve a job is $1/\mu$. Jobs wait in a single queue and the first job in the queue is allocated to the first free server. Let $\rho = \lambda/c\mu$. Then with general arrivals and general service time distributions there are no exact results. One approach is to approximate the system by a $G/G/1$ queue and then modify the performance measures by the relationship between $M/M/c$ and $M/M/1$ results. When the multiple server system is represented by a $G/G/1$ queue, the arrival process at the queueing system is unchanged but the service time of the

single server equivalent to the c parallel servers is scaled by a factor of $1/c$. That is, if the service time at any one of the c servers had mean $\bar{s}$, variance σ_S^2, and squared coefficient of variation (scv) $C_S^2 = \sigma_S^2/\bar{s}^2$, the equivalent single server has service time with mean $\bar{s}/c$, variance σ_S^2/c^2, and scv $= C_S^2$. Then the number of jobs in service and waiting can be approximated by

$$\hat{n}_{G/G/c} = \frac{E[L]_{M/M/c}}{E[L]_{M/\mathbf{M}/1}} (\hat{n}_{G/G/1} - \rho) + c\rho \tag{5}$$

where $\hat{n}_{G/G/1}$ is an approximation for the number of jobs in service and waiting in the approximating $G/G/1$ system. $E[L]_{M/M/c}$ is given by a well-known queueing theory result

$$E[L]_{M/M/c} = \frac{(c\rho)^c}{c!} \left(\frac{\rho}{(1-\rho)^2} \right) p(0)$$

where $p(0) = 1/\{\Sigma_{k=0}^{c-1}(c\rho)^k/k! + (c\rho)^c/(1-\rho)c!\}$. Also note that $E[L]_{M/M/1} = \rho^2/(1-\rho)^2$.

2.1.2.2. Nonidentical servers Assume that all jobs have essentially the same work content. However, machines or servers differ in the time that it takes for them to perform the required work. Suppose that there are c servers or machines and the time required to perform the required work for a job or customer by server j, $j = 1,2, \ldots, c$, is a random variable S_j with mean $\bar{s}_j$.

2.1.2.3. Throughput Then the following table shows the throughput for a number of different rules for allocating jobs to machines or customers to servers.

Allocation rule	TH
First free server	$\Sigma_{j=1}^{c} 1/\bar{s}_j$
To server j with probability p_j	$\min_j 1/p_j\bar{s}_j$
Round robin or cyclic	$\min_j c/\bar{s}_j$

Note that these results depend only on the mean of the S_j. Also note that the first free server rule gives the greatest throughput, although the random allocation can reach the same throughput if p_j is chosen, so $\bar{s}_j p_j = 1/\Sigma_{u=1}^{c} 1/\bar{s}_u$. If, however, $p_j = 1/c$ for all j, then random allocation and cyclic allocation give the same throughput. In service systems, it is usual that servers differ in their capabilities even though they perform the same tasks. It can be seen that these differences will have significant impact on throughput unless it is feasible to use the first-free-server rule.

Suppose the time between arrivals has mean $1/\lambda$ and squared coefficient of variation (scv) C_a^2. With the cyclic allocation rule, the time between arrivals at a given server will have mean c/λ and scv of C_a^2/c. With random allocation, the time between arrivals at server j has mean $1/p_j\lambda_j$ and scv of $1 - p_j + p_j C_a^2$. Thus, if $p_j = 1/c$, the mean time between arrivals at a server are the same but the random allocation has higher scv of arrivals and hence will have longer queues.

2.2. Make-to-Stock Manufacturing Systems or Service Systems where Work Is Done in Advance of Customer Arrival

Assume throughout this subsection that there is a single server.

2.2.1. Exponential Service Time, Poisson Demands

Suppose the target stock level is set as z. Then, as soon as a demand arrives, it is satisfied by an item taken from inventory. A job is then released to the machine or server to begin manufacturing or serving the replenishment. Mean service time of a job is $\bar{s} = 1/\mu$.

2.2.1.1. Backlogged Demands In a system where unmet demands are backlogged, it follows that jobs will arrive at the machine in exactly the same way as demand arrives. Hence the queue length at the machine is the same as the queue length in a make-to-order system, with the same pattern or arrivals of demands and the same service time distribution at machines. However, if N is the length of the queue, then the inventory in the output store will be $\max\{z - N,0\}$ while the size of the backlog will be $\max\{0,N - z\}$. Hence it follows that the expected backlog $E[B]$ is given by

$$E[B] = \frac{\rho^{z+1}}{1 - \rho} \tag{6}$$

and the expected delay in meeting a demand is $\bar{s}(\rho^z/(1 - \rho))$. The probability a demand cannot be met immediately will be ρ^z, so the service level, the fraction of demand met from stock, is $1 - \rho^z$.

2.2.1.2. Lost Sales Alternatively, suppose that demands that cannot be met from stock are lost. Now SL, the fraction of demand met from stock, is given by

$$\text{SL} = \frac{1 - \rho^z}{1 - \rho^{z+1}} \tag{7}$$

2.2.2. Single Machine with Interruptible Demand (Stopped Arrival Queue)

Suppose now that as soon as the store is empty, the demand process is turned off. It is turned on again once there is an item in the store. The arrival and service of jobs at the machine is now equivalent to a *stopped arrival* queue, that is, a queue in which the arrival process turns off once the queue length reaches z. Now the service level is defined by the ratio of the number of demands met to the number that would have been generated if the arrival process were never turned off. This is equal to the fraction of time that the inventory level in the store is greater than zero. With exponential arrivals and exponential service time, the service level is the same as the lost sales case. However, when the interarrival process is general, then the results are somewhat different. Consider a $G/G/1$ queue with arrivals equal to the arrival process of demands and service equal to the machine service time. Let $\hat{n}$ be an approximation for the queue length in this queue and ρ its traffic intensity. Then define σ by $\sigma = (\hat{n} - \rho)/\hat{n}$. If $\rho > 1$, then define $\sigma = 1/\sigma_R$ where $\sigma_R = (\hat{n}_R - \rho_R)/\hat{n}_R$ and $\hat{n}_R$ is the average queue length in a $G/G/1$ queue with arrivals having a distribution equal to the service process and service distribution equal to the interarrival distribution in the original system. ρ_R is defined as the ratio of the mean interarrival time of demands to the mean service time (i.e., $\rho_R = 1/\rho$). Then a good approximation for the service level in this system is given by

$$\text{SL} = \frac{1 - \rho\sigma^{z-1}}{1 - \rho^2\sigma^{z-1}}, \qquad \rho \neq 1$$

$$\text{SL} = \frac{1 + (z - 1)\nu}{2 + (z - 1)\nu}, \qquad \rho = 1 \tag{8}$$

where $\nu = \lim_{\rho \to 1} d\sigma/d\rho$. For example, if the heavy traffic approximation (4) is used, $\nu = 2/(C_a^2 + C_s^2)$ and so when $\rho = 1$,

$$\text{SL} = \frac{C_a^2 + C_s^2 + 2(z - 1)}{2(C_a^2 + C_s^2 + z - 1)} \tag{8}$$

2.2.3. Multiclass Backlogged Demand

Suppose now that there are r types of items produced on a common single machine. The demand rate for type i, $i = 1, \ldots, r$, is λ_i. The service time distribution is identical for all types and has an exponential distribution with mean $1/\mu$. Suppose that there is a target stock level of z_i for type i. Then define $\hat{\rho}_i$ by $\hat{\rho}_i = \lambda_i/(\mu - \Sigma_{j \neq i}\lambda_j)$. The service level for type i is then given by

$$\text{SL}_i = 1 - \hat{\rho}_i^{z_i} \tag{10}$$

2.2.4. Multiclass Lost Sales

Again there are r classes with the target stock z_i for class i, $i = 1,2, \ldots, r$. Demand rate for class i is λ_i while all types are produced on a single machine with the same service time distribution. It is possible to show that the probability of observing the number of jobs of each type in the machine queue of $n_1, n_2, \ldots, n_i, \ldots, n_r$ is given by

$$p(\mathbf{n}) = p(n_1, n_2, \ldots, n_i, \ldots, n_r) = G(\mathbf{Z})^{-1} \binom{\Sigma_{i=1}^r n_i}{n_1, \ldots, n_r} \prod_{i=1}^r \rho_i^{n_i}, \quad 0 \leq n_i \leq z_i;\ i = 1, \ldots, r \tag{11}$$

where the normalizing constant $G(Z)$ is determined by

$$G(\mathbf{Z}) = G(z_1, z_2, \ldots, z_r) = \sum_{n_1=0}^{z_1} \cdots \sum_{n_r=0}^{z_r} \binom{\Sigma_{i=1}^{r} n_i}{n_1, \ldots, n_r} \prod_{i=1}^{r} \rho_i^{n_i} \tag{12}$$

Note that $(\Sigma_{i=1}^{r} n_i / n_1, \ldots, n_r)$ is the multinomial coefficient, that is the number of ways to allocate $\Sigma_{i=1}^{r} n_i$ items to r cells in which cell i, $i = 1, 2, \ldots, r$, contains n_i items.

The service level for type i items is given by

$$\begin{aligned} \mathrm{SL}_i &= \lambda_i \left(1 - \sum_{n_1=0}^{z_1} \cdots \sum_{n_i=z_i} \cdots \sum_{n_r=0}^{z_r} p(\mathbf{n})\right) \\ &= \lambda_i \frac{G(z_1, z_2, \ldots, z_{i-1}, z_i - 1, z_{i+1}, \ldots, z_r)}{G(\mathbf{Z})}, \quad i = 1, \ldots, r \end{aligned} \tag{13}$$

2.2.5. Produce to Stock with Advance Information

Suppose that it is possible to obtain information about future demands in advance. That is, the arrival of the nth demand, $n = 1, 2, \ldots$, at time t is signaled at time $t - \tau$. If the target or initial stock of finished products is zero, then the item to meet the demand at time t can be released for manufacture at time $t - \tau$. Now if a target stock z of finished products is held, release of an item to manufacture will still take place at time $t - \tau$, but the nth demand, which arrives at time t, will actually be met by an item released for manufacture prior to time $t - \tau$ and triggered by the advice of demand $n - z$. Suppose service times are exponential with parameter μ and advices about future demands arrivals are Poisson with rate λ. Let $\rho = \lambda/\mu$. Then the service level SL, i.e., the fraction of demands that are met from stock, is given by (Buzacott and Shanthikumar 1994).

$$\mathrm{SL} = 1 - \rho^z e^{-\mu\tau(1-\rho)} \tag{14}$$

and the average delay in meeting a demand is given by

$$\overline{w}/\overline{s} = e^{-\mu\tau(1-\rho)} \frac{\rho^z}{1 - \rho} \tag{15}$$

The average inventory is given by

$$\overline{i} = z + \lambda\tau - \frac{\rho}{1 - \rho}(1 - \rho^z e^{-\mu\tau(1-\rho)}) \tag{16}$$

3. FLOW LINES AND SERIES SYSTEMS

3.1. Introduction

Flow lines and series systems can be divided up into two broad classes, based on their influence on the worker: *paced* and *unpaced*. In a *paced* system the time allowed to perform a task is limited and once this time is up the job or customer can no longer be worked on, so it is possible that the task may not be completed. In an *unpaced* system there is no maximum time limit imposed on the time for the worker to perform the task.

Paced systems lose throughput because of the incomplete processing of jobs or customers, while unpaced systems lose throughput because of the variability of task times.

3.2. Models of Paced Systems

In a paced system, the tolerance time, τ, the maximum time available to perform the tasks at any workstation, is set. Thus if T_i, the time required by the worker at station i to perform their required tasks, exceeds τ, the tasks will be incomplete and defective products will result. Hence the probability $Q(\tau)$ that a product will not contain any defects is

$$Q(\tau) = P\{T_1 \le \tau, T_2 \le \tau, \ldots, T_m \le \tau\} = \prod_{i=1}^{m} P\{T_i \le \tau\} \tag{17}$$

To meet a given quality target Q, the probability that the product is nondefective should be at least Q. So the minimum tolerance time is the solution to

$$F_i(\tau) = Q^{1/m} \tag{18}$$

One of the managerial controls in a paced system is the tolerance time τ, which is related to the line

speed. Too short a tolerance time means that the quality is low, while too high a tolerance time reduces the utilization of the workers and so lowers productivity. For example, if $m = 10$ and the quality target $Q = 0.98$ then $F_i(\tau)$ must be at least 0.998. This means that if task times were exponential, and with the same distribution at all stations, the tolerance time τ has to be set at $6.2 \times$ the mean processing time of a job at a station. If task times have a normal distribution with mean θ and standard deviation σ, then the tolerance time will have to be set as $\tau = \theta + 2.9\sigma$. It is clear that this would lead to a substantial loss in labor productivity unless the variability of task times can be kept small.

Note that for Q close to 1, $F_i(\tau)$ is approximately given by

$$1 - F_i(\tau) \approx \frac{1 - Q}{m} \tag{19}$$

The gross output rate will be $1/\tau$ but the system throughput of nondefectives will be

$$TH = \frac{Q(\tau)}{\tau} \tag{20}$$

3.3. Models of Unpaced Lines

The unpaced line consists of m stages. Once a job is completed at stage i, it moves on to stage $i + 1$, although it may pass through an intermediate buffer of capacity b_i, $i = 1, 2, \ldots, m - 1$, where capacity is made up of the space for jobs in process at the stage plus the space in the buffer.

3.3.1. Infinite Buffer Systems

Suppose that there is one machine or server at each stage. The processing time of a job at stage i is exponential with mean $1/\mu_i$. Jobs or customers arrive at the system according to a Poisson process with rate λ. In an infinite buffer system, we have

$$\text{TH} = \min_i \left(\frac{1}{\mu_i}\right) \tag{21}$$

and if $\lambda < \text{TH}$ and $\rho_i = \lambda/\mu_i$, then the distribution of N_i, the inventory at station i, is given by

$$P(N_i = k_i) = \rho_i^{k_i}(1 - \rho_i)$$

It follows that the average inventory at stage i is given by

$$E[N_i] = \frac{\rho_i}{1 - \rho_i}$$

and hence the total inventory $\overline{n}$ is given by

$$\overline{n} = \sum_{i=1}^{m} \frac{\rho_i}{1 - \rho_i} \tag{22}$$

3.4. Two-Stage Flow Lines

Consider a flow line that consists of two stages separated by a buffer of capacity z. Each stage consists of a single server, so $b = z + 1$.

3.5. Exponential Service Times

Assume that the service time at each stage is exponentially distributed with mean $1/\mu_i$, $i = 1,2$. Let $\rho = \mu_1/\mu_2$. Then, using a Markov model, it is possible to show that the throughput of this system is given by

$$\begin{aligned} \text{TH} &= \mu_1 \frac{(1 - \rho^{b+1})}{1 - \rho^{b+2})} \qquad \rho \neq 1 \\ &= \mu_2 \frac{(b + 1)}{b + 2} \qquad \rho = 1 \end{aligned} \tag{23}$$

Alternatively, write

$$\text{TH} = \mu_1(1 - B(\mu_1, \mu_2, b + 1)) = \mu_2(1 - I(\mu_1, \mu_2, b + 1))$$

where

$$B(\mu_1, \mu_2, b + 1) = \frac{\rho^{b+1}(1 - \rho)}{1 - \rho^{b+2}} \quad \rho \neq 1$$
$$= \frac{1}{b + 2} \quad \rho = 1 \tag{24}$$

$$I(\mu_1, \mu_2, b + 1) = \frac{1 - \rho}{1 - \rho^{b+2}} \quad \rho \neq 1$$
$$= \frac{1}{b + 2} \quad \rho = 1 \tag{25}$$

Note that $1/\text{TH}$ is the mean time between parts arriving at the line and equals the mean time between parts leaving the line. That is, the system behaves as if it were equivalent to a single machine with mean service time $1/\mu^*$ given by the two equivalent expressions

$$\frac{1}{\mu^*} = \frac{1}{\mu_1} + \frac{1}{\mu_2} B(\mu_1, \mu_2, b) \tag{26}$$

$$= \frac{1}{\mu_2} + \frac{1}{\mu_1} I(\mu_1, \mu_2, b) \tag{27}$$

3.6. General Service Times

If the service times at the two stations are general, then a good approximation can be obtained by viewing the system as a stopped arrival queue, where stage 1 corresponds to the arrival process and stage 2 corresponds to the service process (Buzacott et al. 1995). Suppose the service time at stage i is a random variable S_i, $i = 1, 2$. Define $\rho = E[S_2]/E[S_1]$, $\rho_R = 1/\rho$, and $\lambda = 1/E[S_1]$. If the buffer capacity is z, then the maximum number of jobs in the system is $z + 2$ and $b = z + 1$. Then the approximation is

$$\text{TH} = \lambda \frac{(1 - \rho\sigma^b)}{1 - \rho^2\sigma^b} \quad \rho \neq 1$$
$$= \lambda \frac{C_{S1}^2 + C_{S2}^2 + 2b}{2(C_{S1}^2 + C_{S2}^2 + b)} \quad \rho = 1 \tag{28}$$

with

$$\sigma = (\hat{n} - \rho)/\hat{n} \quad \rho < 1$$
$$= \hat{n}_R/(\hat{n}_R - \rho_R) \quad \rho > 1$$

where $\hat{n}$ is the approximate average number of jobs in a $G/G/1$ queueing system with arrival distribution S_1 and service distribution S_2, while $\hat{n}_R$ is the approximate average number of jobs in a $G/G/1$ queueing system with arrival distribution S_2 and service distribution S_1. The two moment approximations given above can be used to find $\hat{n}$ or $\hat{n}_R$ as appropriate. Table 1 shows the accuracy of the approximation for the case where S_1 and S_2 have Erlang-3 distributions.

3.7. Three-Stage Flow Lines

Consider a three-stage flow line with finite buffer storage space. The number of spaces in the buffer i between stages $i - 1$ and i is z_1. Set $b_i = z_i + 1$. Assume that the service time at each stage is exponentially distributed with mean $1/\mu_i$, $i = 1, \ldots, m$.

Such a system is best analyzed using a Markov process model. Then the state of the system is defined by $\{n_2, n_3\}$, where n_i is the number of jobs in the system that have been processed by station $i - 1$ but have not yet completed processing by station i, $i = 2, 3$. The state space $\mathcal{S} = \{(n_2,n_3)$: $0 \leq n_2 \leq b_2 + 1$, $0 \leq n_3 \leq b_3 + 1$, $n_2 + n_3 \leq b_2 + b_3 + 1\}$. Note that when $n_i = b_i + 1$, stage $i - 1$ is blocked by stage i. Let $\mathbf{p} = (p(n_2, n_3), (n_2, n_3) \in \mathcal{S})$ be the stationary probability vector of this Markov process. The steady-state balance equations for $\mathbf{p}$, obtained by equating the rate of leaving a state with the rate of entering the state, are given by

TABLE 1 Adequacy of Throughput Approximation $C_{S1}^2 = 1/3$, $C_{S2}^2 = 1/3$

		b				
ρ	TH/λ	1	2	3	4	5
0.5	Sim.	0.9432	0.9928	0.9989	0.9999	1.0000
	approx.	0.9462	0.9895	0.9978	0.9996	0.9999
0.8	Sim.	0.8390	0.9348	0.9704	0.9864	0.9938
	approx.	0.8658	0.9411	0.9710	0.9850	0.9920
1.0	Sim.	0.7595	0.8583	0.9002	0.9250	0.9395
	approx.	0.8000	0.8750	0.9091	0.9286	0.9412
1.25	Sim.	0.6729	0.7475	0.7744	0.7882	0.7954
	approx.	0.6926	0.7529	0.7768	0.7880	0.7937
2.0	Sim.	0.4723	0.4976	0.5013	0.4991	0.5002
	approx.	0.4731	0.4947	0.4989	0.4998	0.5000

From Buzacott and Shanthikuma, © 1993. Reprinted by permission of Prentice-Hall Inc., Upper Saddle River, NJ.

$$
\begin{aligned}
\mu_1 p(0,0) &= \mu_3 p(0,1) \\
(\mu_1+\mu_3)p(0,n_3) &= \mu_2 p(1,n_3-1) + \mu_3 p(0,n_3+1), \quad 1 \le n_3 \le b_3 \\
(\mu_1+\mu_3)p(0,b_3+1) &= \mu_2 p(1,b_3) \\
(\mu_1+\mu_2)p(n_2,0) &= \mu_1 p(n_2-1,0) + \mu_3 p(n_2,1), \quad 1 \le n_2 \le b_2 \\
(\mu_1+\mu_2+\mu_3)p(n_2,n_3 &= \mu_1 p(n_2-1,n_3) + \mu_2 p(n_2+1,n_3-1) + \mu_3 p(n_2,n_3+1), \\
&\qquad 1 \le n_2 \le b_2,\ 1 \le n_3 \le b_3 \\
(\mu_1+\mu_3)p(n_2,b_3+1) &= \mu_1 p(n_2-1,b_3+1) + \mu_2 p(n_2+1,b_3), \quad 1 \le n_2 \le b_2-1 \\
\mu_3 p(b_2,b_3+1) &= \mu_1 p(b_2-1,b_3+1) + \mu_2 p(b_2+1,b_3) \\
\mu_2 p(b_2+1,0) &= \mu_1 p(b_2,0) + \mu_3 p(b_2+1,1) \\
(\mu_2+\mu_3)p(b_2+1,n_3) &= \mu_1 p(b_2,n_3) + \mu_3 p(b_2+1,n_3+1), \quad 1 \le n_3 \le b_3-1 \\
(\mu_2+\mu_3)p(b_2+1,b_3) &= \mu_1 p(b_2,b_3)
\end{aligned}
$$

These $|\mathcal{S}| = (b_2+1)(b_3+1) - 1$ equations along with the normalizing equation

$$\sum_{(n_2,n_3)\in\mathcal{S}} p(n_2,n_3) = 1$$

can be solved for **p**. Unfortunately, they do not possess a formula type solution.

With present computing facilities, direct solution of the equations using a standard procedure is probably as easy a method to use as any other. However, as b_2 and b_3 increase, it would be desirable to make use of the fact that the coefficient matrix has a large number of zero entries and thus a sparse matrix solution procedure would be appropriate. Once **p** has been computed, the throughput can be obtained by

$$\mathrm{TH} = \mu_3 \left(1 - \sum_{n_2=0}^{b_2+1} p(n_2,0)\right)$$

For the special case of no (extra) buffer capacity (i.e., $b_2 = 1$, $b_3 = 1$), an explicit formula for the throughput can be obtained. It is

$$\mathrm{TH}(\mu_1,\mu_2,\mu_3) = 1 \Bigg/ \left\{ \frac{1}{\mu_2} + \frac{\mu_2(\mu_1+\mu_3)}{\mu_1\mu_3(\mu_1+\mu_2)(\mu_2+\mu_3)} \left[\frac{(\mu_1^2+\mu_3^2)(\mu_1+\mu_2+\mu_3)^2 - \mu_1^2\mu_3^2}{(\mu_1+\mu_3)^3 + \mu_2(\mu_1^2+\mu_1\mu_3+\mu_3^2)} \right] \right\} \tag{29}$$

This approach of setting up and solving a Markov model can be used for other similar systems with exponential service times. Typically, the number of equations to be solved increases rapidly

with the number of stages and the size of the buffers, so while straightforward it becomes increasingly difficult to use.

3.8. Multiple-Stage Flow Lines with Exponential Processing Times

The flow line is modeled by a tandem queueing system with m stages and with finite buffer capacities $b_2, \ldots, b_m$ with $b_i = z_i + 1$, where z_i is the number of spaces in buffer i, and exponentially distributed processing times with mean $1/\mu_i$, $i = 1,2, \ldots, m$. The jobs arrive at the flow line according to a Poisson process with rate λ.

We use this situation to illustrate a particular approach to developing approximate models of manufacturing and service system performance that is of wide applicability.

The basis of the approximate approach is to consider stage i as if it were isolated from the rest of the system, and so we model it as an $M/M/1/b_i + 1$ queue. Jobs are assumed to arrive as a Poisson process with rate $\hat{\lambda}_i$ and service is exponentially distributed with rate μ_{id}, so the probability that an arriving job is blocked is given by $B(\hat{\lambda}_i, \mu_{id}, b_i + 1)$, where $B(\hat{\lambda}_i, \mu_{id}, b_i+1)$ is given by Eq. (24). Observe that if $\lambda < \mathrm{TH}$, the job departure rate for stage i is λ. Thus,

$$\hat{\lambda}_i(1 - B(\hat{\lambda}_i, \mu_{id}, b_i + 1)) = \lambda \tag{30}$$

Also, considering stage $i + 1$ as a two-stage flow line, using Eq. (26) we approximate μ_{id} by

$$\frac{1}{\mu_{id}} = \frac{1}{\mu_i} + \frac{1}{\mu_{i+1d}} B(\hat{\lambda}_{i+1}, \mu_{i+1d}, b_{i+1}), \qquad i = 1, \ldots, m - 1 \tag{31}$$

and $\mu_{md} = \mu_m$. That is, we have the recursive relations (30) and (31) to solve for $\hat{\lambda}_i$, $i = 1, \ldots, m - 1$ with the initial condition $\mu_{md} = \mu_m$. Note that for (30) to have a solution, we require $\mu_{id} > \lambda$, $i = 1, \ldots, m - 1$. The mean number of jobs in the system and at the different stages can be approximated by $\Sigma_i \overline{N}_i$ and $(\hat{N}_i, i = 1, \ldots, m)$ delivered by the following algorithm. Let $\hat{N}_{m/M/1/b}(\lambda, \mu, b)$ be the mean number of customers in an $M/M/1/b$ queueing system with arrival rate λ, service rate μ, and buffer capacity b. If $\rho = \lambda/\mu$,

$$\hat{N}_{M/M/1/b}(\lambda, \mu, b) = \frac{\rho}{1 - \rho} - \frac{(b + 1)\rho^{b+1}}{1 - \rho^{b+1}}$$

3.8.1. Algorithm 1: Work-in-Process (Finite Buffer Flow Lines: Single Servers, Exponential Processing Times)

Step 1: Set $\mu_{md} = \mu_m$.
Step 2: For $i = m, \ldots, 2$,
  if $\mu_{id} < \lambda$, then the system is unstable, go to step 4;
  else, solve $\hat{\lambda}_i(1 - B(\hat{\lambda}_i, \mu_{id}, b_i + 1)0 = \lambda$ for $\hat{\lambda}_i$
  Set $1/\mu_{i-1d} = 1/\mu_{i-1} + 1/\mu_{id}\, B(\hat{\lambda}_i, \mu_{id}, b_i)$.
  Compute $N_1 = \hat{N}_{M/M/1/b}(\hat{\lambda}_i, \mu_{id}, b_i + 1)$.
Step 3: Compute $N_1 = \hat{N}_{M/M/1}(\lambda/\mu_{1d})$.
Step 4: Stop.

In step 2 we need to solve the nonlinear equation $\hat{\lambda}(1 - B(\hat{\lambda}, \mu, b + 1)) = \lambda$ for the unknown $\hat{\lambda}$ when $\mu > \lambda$. Since $\hat{\lambda}(1 - B(\hat{\lambda}, \mu, b+1))$ is increasing and concave in $\hat{\lambda}$, the following iterative scheme

$$\hat{\lambda}^{(k)} = \hat{\lambda}^{(k-2)} + \frac{(\hat{\lambda}^{(k-1)} - \hat{\lambda}^{(k-2)})(\lambda - \hat{\lambda}^{(k-2)}(1 - B(\hat{\lambda}^{(k-2)}, \mu, b + 1)))}{\hat{\lambda}^{(k-1)}(1 - B(\hat{\lambda}^{(k-1)}, \mu, b + 1)) - \hat{\lambda}^{(k-2)}(1 - B(\hat{\lambda})^{k-2}, \mu, b + 1))}, \quad k = 2, \ldots,$$

starting with $\hat{\lambda}^{(0)} = 0$; $\hat{\lambda}^{(1)} = \lambda$, will monotonically converge to the solution.

In algorithm 1 we see that if $\lambda > \mu_{id}$ for any i, $i = 1, \ldots, m$, the system is unstable. Let $\lambda^* = \max\{\lambda : \mu_{id} \geq \lambda, i = 1, \ldots, m\}$. We will use λ^* as our approximate throughput of the flow line. It can be verified that the following iterative scheme will converge and provide λ^*. Let

$$I(\lambda, \mu, b) = \frac{1 - \rho}{1 - \rho^{b+1}}$$

be the steady-state probability that the server is idle in an $M/M/1/b$ queueing system with arrival rate λ, service rate μ, and buffer capacity b.

3.8.2. Algorithm 2: Throughput (Finite Buffer Flow Lines: Single Servers, Exponential Processing Times)

Step 1: Set $\mu_{1u} = \mu_1$; $\mu_{id} = \mu_i$, $i = 2, \ldots, m$.
Step 2: For $i = 2, \ldots, m$, compute
$1/\mu_{iu} = 1/\mu_i + 1/\mu_{i-1u}\, I(\mu_{i-1u}, \mu_{id}, b_i)$
Step 3: For $i = m, \ldots, 2$, compute
$1/\mu_{i-1d} = 1/\mu_{i-1} + 1/\mu_{id}\, B(\mu_{i-1u}, \mu_{id}, b_i)$
Step 4: If $|\mu_1(1 - B(\mu_1, \mu_{2d}, b_2 + 1)) - \mu_m(1 - I(\mu_{m-1u}, \mu_m, b_m + 1))| < \epsilon$, set $\lambda^* = \mu_1(1 - B(\mu_1, \mu_{2d}, b_2 + 1))$ and stop. Otherwise go to step 2.

3.9. General Service Time Approximation

Suppose that the service times at the stages are random variables S_j, $j = 1, \ldots, m$. Let $1/\mu_j$ and C_{Sj}^2 be the mean and squared coefficient of variation of the service time at stage j. Suppose the buffer capacity between stage $j - 1$ and j is z_j, $j = 2, \ldots, m$.

The basis of the approximation is to analyze each stage j, $j = 2, \ldots, m$ as a $GI/GI/I/z_j + 2$ stopped arrival queue. The effective arrival process and service times at stage j, $j = 2, \ldots, m$ will be chosen so that they effectively reflect the upstream and downstream portions of the flow line, where upstream means the overall effect of stages $1, \ldots, j - 1$ and downstream means the overall effect of stages $j, \ldots, m$. Let $(1/\mu_{j-1u}, C_{S_{j-1u}}^2)$ denote the mean and scv of the effective input or arrival process to stage j when turned on (i.e., when stage $j - 1$ is not blocked because the queue is full), and let $(1/\mu_{jd}, C_{S_{jd}}^2)$ be the mean and scv of the effective service or output process at stage j (i.e., when stage j is not starved because the stage is empty). Also let TH_j be the throughput of the stage j stopped arrival queue, obtained using a $GI/GI/1/b_j + 1$ stopped arrival queue approximation with input parameters $(1/\mu_{j-1u}, C_{S_{j-1u}}^2)$ and service parameters $(1/\mu_{jd}, C_{S_{jd}}^2)$.

There are useful equations relating μ_{ju} and μ_{jd}. Consider stage j. The time between the $k - 1$th and kth job departures from stage j can be written as

$$T_k^{(j)} = I_k^{(j)} + S_k^{(j)} + H_k^{(j)} \tag{32}$$

where $I_k^{(j)}$ is the idle time of stage j waiting for the next job to arrive and $H_k^{(j)}$ is the holding or blocking time of the job while it waits for queue space in the buffer at stage $j + 1$. Next observe that $S_k^{(j)} + H_k^{(j)}$ is the effective service time experienced by a job in the stopped arrival queue corresponding to stage j, that is $E[S_k^{(j)} + H_k^{(j)}] = 1/\mu_{jd}$. Also note that $I_k^{(j)} + S_k^{(j)}$ is the effective interarrival time in the stopped arrival queue corresponding to stage $j + 1$, that is, $E[I_k^{(j)} + S_k^{(j)}] = 1/\mu_{ju}$. Also, $E[T_k^{(j)}]$ is the mean interdeparture time from the original system and $1/\mathrm{TH}_{j+1}$ is from the decomposed system. In order for $1/\mathrm{TH}_{j+1}$ to be correct, we must have $E[T_k^{(j)}] = 1/\mathrm{TH}_{j+1}$. Thus, rewriting (32),

$$S_k^{(j)} + H_k^{(j)} = S_k^{(j)} + T_k^{(j)} - (I_k^{(j)} + S_k^{(j)})]$$

and taking expectations, we obtain

$$\frac{1}{\mu_{jd}} = \frac{1}{\mu_j} + \frac{1}{TH_{j+1}} - \frac{1}{\mu_{ju}}$$

Note that the average blocking time of stage j is $1/\mathrm{TH}_{j+1} - 1/\mu_{ju}$.

Similarly, by considering the time between successive job inputs to stage $j + 1$ it can be shown that

$$\frac{1}{\mu_{j+1u}} = \frac{1}{\mu_{j+1}} + \frac{1}{\mathrm{TH}_{j+1}} - \frac{1}{\mu_{j+1d}}$$

That is, we get the following set of $2(m - 1)$ equations

$$\frac{1}{\mu_{jd}} = \frac{1}{\mu_j} + \frac{1}{\mathrm{TH}_{j+1}} - \frac{1}{\mu_{ju}}, \quad j = 2, \ldots, m-1 \tag{33}$$

$$\frac{1}{\mu_{md}} = \frac{1}{\mu_m}$$

$$\frac{1}{\mu_{1u}} = \frac{1}{\mu_1}$$

$$\frac{1}{\mu_{j+1u}} = \frac{1}{\mu_{j+1}} + \frac{1}{\mathrm{TH}_{j+1}} - \frac{1}{\mu_{j+1d}}, \quad j = 1, \ldots, m-2 \tag{34}$$

Given the $C^2_{S_{ju}}$ and the $C^2_{S_{jd}}$ (see below as to alternative approaches for determining these quantities), the above $2(m-1)$ equations plus the $m-1$ equations to determine TH_j, $j = 2, \ldots, m$, the throughput of each stopped arrival queue from the mean and scv of the (equivalent) arrival and service processes, give $3(m-1)$ equations in the $3(m-1)$ variables TH_j, μ_{j-1u} and μ_{jd}, $j = 2, \ldots, m$. These equations can be solved recursively.

3.9.1. Algorithm 3: Throughput (Finite Buffer Flow Lines: Single Servers, General Processing Times)

Step 0: Set $\mu^{(0)}_{jd} = \mu_j$, $j = 2, \ldots, m$.

Step 1: For $k = 1, 2, \ldots$ and for $j = 1, \ldots, m-1$, calculate (i) $\mathrm{TH}^{(k)}_{j-1}$ using the $GI/GI/1/b_{j-1}+1$ stopped arrival queue approximation with input parameters $(1/\mu^{(k)}_{ju}, C^2_{S_{ju}})$ and service parameters $(1/\mu^{(k-1)}_{j+1d}, C^2_{S_{j+1d}})$, (ii) $\mu^{(k)}_{j+1u}$ using

$$\frac{1}{\mu^{(k)}_{j+1u}} = \frac{1}{\mu_{j+1}} + \frac{1}{\mathrm{TH}^{(k)}_{j+1}} - \frac{1}{\mu^{(k-1)}_{j+1d}}, \quad j = 1, \ldots, m-2,$$

and (iii) $\mu^{(k)}_{jd}$ using

$$\frac{1}{\mu^{(k)}_{jd}} = \frac{1}{\mu_j} + \frac{1}{\mathrm{TH}^{(k)}_{j+1}} - \frac{1}{\mu^{(k)}_{ju}}, \quad j = 2, \ldots, m-1$$

Step 2: Stop once

$$\left|\mathrm{TH}^{(k)}_m - \mathrm{TH}^{(k-1)}_m\right| < \epsilon$$

It can easily be verified that at convergence $\mathrm{TH}^{(k)}_i = \mathrm{TH}^{(k)}_j$ for all $i \neq j$.

3.9.2. Squared Coefficient of Variation Recursions

So far it has not been specified how to calculate the scv $C^2_{S_{ju}}$ and $C^2_{S_{jd}}$. Two approaches are possible. The simplest, approximation (a), is to set

$$C^2_{S_{ju}} = C^2_{S_j}, \quad j = 1, \ldots, m-1$$

$$C^2_{S_{jd}} = C^2_{S_j}, \quad j = 2, \ldots, m$$

This approximation considers only the immediate upstream and downstream stations and ignores the impact of possible blocking or starving on the variance of the equivalent service times or arrival times.

The other approach [see Buzacott et al. 1995, approximation (b)] tries to take account of the impact of possible blocking and starving on the variance of the service times. It uses the following set of $2(m-2)$ equations by which the scv's can be determined

$$E[S^2_{jd}] = E[S^2_j] + \delta_{j+1d}(E[S^2_{j+1d}] + 2E[S_j]E[S_{j+1d}]), \quad j = 2, \ldots, m-1$$

with $E[S^2_{md}] = E[S^2_m]$ and δ_{j+1d} defined by

$$\delta_{j+1d} = \mu_{j-1d}\left(\frac{1}{\mathrm{TH}_{j+1}} - \frac{1}{\mu_{ju}}\right), \quad j = 2, \ldots, m-1.$$

and

$$E[S_{ju}^2] = E[S_j^2] + \delta_{ju}(E[S_{j-1u}^2] + 2E[S_j]E[S_{j-1u}]) \quad j = 2, \ldots, m-1$$

with $E[S_{1u}^2] = E[S_1^2]$ and

$$\delta_{ju} = \mu_{j-1u}\left(\frac{1}{\mathrm{TH}_j} - \frac{1}{\mu_{jd}}\right) \quad j = 2, \ldots, m-1$$

Table 2 illustrates the approximation for a four-stage system where the service times are exponentially distributed. Note that the case (a) approximation is now identical to algorithm 2 given above for the throughput of an exponential flow line.

4. TRANSFER LINES

In a transfer line, movement of jobs at all stations in a section of the line is synchronized. Thus, transfer can only begin when the slowest station has completed its operation. If no station fails, then the mean time between successive transfers is the *cycle time*, τ. The cycle time will consist of the maximum time to perform operations at a station in the section, plus the time required for transfer. The *gross production rate* is $1/\tau$. The *net production rate* is TH and is less than the gross production rate because of line stoppages due to station or transfer mechanism failure. Note that if any station in the section fails and thus its operation is not completed, then no transfer will occur and all stations in the section will be forced down. The extent to which this section stoppage will affect other sections of the line depends on the degree of integrated linkage and control between sections. If no inventory can be kept between sections, then the rest of the line will be forced down almost immediately. In lines with many stations, there is much equipment that can break down, so the net production rate can be much less than the gross production rate. The efficiency of the transfer line is defined by

$$\eta = \frac{\text{net production rate}}{\text{gross production rate}} = \mathrm{TH}\tau$$

One means of increasing the net production rate of a transfer line and reducing the impact of stoppages is to insert in-process storage or a bank between the two sections of the line. A bank has the effect of decoupling the two sections of the line, allowing each section to operate independently.

4.1. Models

4.1.1. Transfer Lines with No Inventory Banks

Consider a transfer line with m stages or stations. The station failures can be either time dependent (failure is possible when the station is idling) or operation dependent (failure only occurs when the station is working on a part).

4.1.2. Time-Dependent Failures

Suppose the stations can fail even when they are idling. Let the mean time to failure, and repair times of stage i, be $\overline{U}_i$ and $\overline{D}_i$ respectively, for $i = 1, 2, \ldots, m$. With no inventory bank, all stations in the line stop as soon as any station fails. Then it follows that

TABLE 2 Four Stage Throughput with Exponential Service Times

Parameters							TH		
μ_i				z_i				App.	
1	2	3	4	2	3	4	Exact	(a)	(b)
1	1.1	1.2	1.3	1	1	1	0.710	0.689	0.700
1	1.2	1.4	1.6	1	1	1	0.765	0.746	0.756
1	1.5	2	2.5	1	1	1	0.861	0.850	0.855
1	2	3	4	1	1	1	0.929	0.925	0.927

From Buzacott et al. 1995. Reproduced with permission of Kluwer Academic Publishers.

$$\eta = \prod_{i=1}^{m} A_i^T \tag{35}$$

where

$$A_i^T = \frac{\bar{U}_i}{\bar{U}_i + \bar{D}_i} \qquad i = 1, \ldots, m$$

4.1.3. Operation-Dependent Failures

Suppose that stations can fail only when they are working on a part. Let $\bar{T}_i$ be the mean number of cycles station i operates between failures, $\bar{D}_i$ be the mean downtime of station i, and τ be the average cycle time when the line is running and $x_i^o = \bar{D}_i/(\bar{T}_i\tau)$. Then the line consisting just of station i would have efficiency

$$A_i^o = \frac{\bar{T}_i}{\bar{T}_i + \bar{D}_i/\tau} = \frac{1}{1 + x_i^o} \qquad i = 1, \ldots, m$$

Then it can be shown that

$$\eta = \frac{1}{1 + \Sigma_{i=1}^{m} x_i^o} \tag{36}$$

If, however, the part in process at the instant of failure must be scrapped, then

$$\eta = \frac{\Pi_{i=1}^{m}(1 - 1/\bar{T}_i)}{1 + \Sigma_{i=1} m x_i^o \Pi_{j=1}^{i-1}(1 - 1/\bar{T}_j)} \tag{37}$$

If only a fraction q_i of the failures result in scrapping of the part, then

$$\tau = \frac{\Pi_{i=1}^{m}(1 - q_i/\bar{T}_i)}{1 + \Sigma_{i=1}^{m} x_i^o \Pi_{j=1}^{i-1}(1 - q_j/\bar{T}_j)} \tag{38}$$

4.1.4. Systems Separated by Infinite Inventory Banks

Suppose the stages in an m-stage transfer line are separated by inventory banks of infinite capacity. Let the mean time to failure and repair times of stage i be $\bar{U}_i$ and $\bar{D}_i$ respectively, for $i = 1, \ldots, m$. Define

$$x_i = \frac{\bar{D}_i}{\bar{U}_i} \qquad i = 1, \ldots, m$$

4.1.4.1. No Parts Scrapped The efficiency of the line is

$$\eta = \min\left\{\frac{1}{1 + x_i} \qquad i = 1, \ldots, m\right\} \tag{39}$$

4.1.4.2. Scrapping of Parts Suppose when station i fails, the part being processed is scrapped, with probability q_i, $i = 1, \ldots, m$. Then

$$\eta = \min_{1 \le j \le m}\left\{\prod_{k=j}^{m}\left(\frac{1 - q_k/\bar{T}_k}{1 + x_j}\right)\right\} \tag{40}$$

4.1.5. Two-Stage Synchronized Line with Finite Capacity Inventory Banks

Consider a series transfer line with two stations, one finite intermediate inventory bank with capacity z and synchronized part transfer. Suppose

$a_i = P$(station i is failed at time $n + 1$|station i is working at time n)
$\hat{a}_i = P$(station i is failed at time $n + 1$|station i is either blocked or idle at time n)
$b_i = P$(station i is up at time $n + 1$|station i is down at time n) $\quad i = 1, 2$

$\hat{a}_i$ is zero if failures are operation dependent, and equal to a_i if failures are time dependent.

Then a discrete time Markov chain model with the system observed just prior to the beginning of transfer can be used to show that the line efficiency is given by

$$\eta(z) = \frac{1 - r^*\rho^z}{1 + x_1 - (1 + x_2)r^*\rho^z} \qquad a_1b_2 \neq b_1a_2 \tag{41}$$

where $x_i = a_i/b_i$, that is, $1/(1 + x_i)$ is the efficiency of station i if it operates on its own,

$$r^* = \frac{(\hat{a}_1 + b_1)(\hat{a}_2\alpha_1 + a_2\beta_1)}{(\hat{a}_2 + b_2)(\hat{a}_1\alpha_2 + a_1\beta_2)} \tag{42}$$

$$\rho = \frac{\beta_1\alpha_2}{\alpha_1\beta_2} \tag{43}$$

and

$$\begin{aligned}
\alpha_1 &= a_1 + a_2 - a_1a_2 - b_1a_2 \\
\alpha_2 &= a_1 + a_2 - a_1a_2 - a_1b_2 \\
\beta_1 &= b_1 + b_2 - b_1b_2 - a_1b_2 \\
\beta_2 &= b_1 + b_2 - b_1b_2 - b_1a_2.
\end{aligned} \tag{44}$$

4.1.5.1. Balanced Stages (Case Where $a_1b_2 = b_1a_2$) This means that $x_1 = x_2 = \hat{x}$, $\alpha_1 = \alpha_2 = \alpha$, $\beta_2 = \beta_2 = \beta$, $\rho = 1$ and $\hat{x} = \alpha/\beta$. Then it can be shown that

$$\eta(z) = \frac{\hat{a}_1 + b_1 + \hat{a}_2 + b_2 - (\hat{a}_1 + b_1)(\hat{a}_2 + b_2) + (\hat{a}_1 + b_1)(\hat{a}_2 + b_2)z(1 + \hat{x})}{(\hat{a}_1 + b_1 + \hat{a}_2 + b_2)(1 + \hat{x}) + (\hat{a}_1 + b_1)(\hat{a}_2 + b_2)(\hat{x}(b_1 + b_2)/b_1b_2 + (z - 1)(1 + \hat{x})^2)} \tag{45}$$

Note that if $\hat{a}_1 = \hat{a}_2 = \hat{a}$, $b_1 = b_2 = b$, that is, the stations are identical,

$$\eta(z) = \frac{2 - b - \hat{a} + (b + \hat{a})z(1 + \hat{x})}{2(1 + 2\hat{x} + \hat{x}\hat{a}/b) + (b + \hat{a})(z - 1)(1 + \hat{x})^2} \tag{46}$$

4.1.6. Operation-Dependent and Time-Dependent Failures

Two limiting cases are of interest: (1) operation-dependent failures: $\hat{a}_1 = \hat{a}_2 = 0$, and (2) time-dependent failures: $\hat{a}_1 = a_1$, $\hat{a}_2 = a_2$. For operation-dependent failures and identical stations,

$$\eta(z) = \frac{2 - b + bz(1 + \hat{x})}{2(1 + 2\hat{x}) + b(z - 1)(1 + \hat{x})^2} \tag{47}$$

and for time-dependent failures

$$\eta(z) = \frac{2 - b(1 + \hat{x}) + bz(1 + \hat{x})^2}{2(1 + \hat{x})^2 + b(z - 1)(1 + \hat{x})^3} \tag{48}$$

4.1.6.1. Failure-Transfer Coefficients The quantities $\delta(z)$ and $\delta'(z)$ defined next are the failure transfer coefficients.

$$\delta(z) = Pr\{\text{bank is empty}|\text{station 1 is failed}\} \tag{49}$$

$$\delta'(z) = Pr\{\text{bank is full}|\text{station 2 is failed}\} \tag{50}$$

These quantities will be used later to describe an approximation for the efficiency of m-stage transfer lines. Later we will use $\delta(z, a_1, b_1, a_2, b_2)$ and $\delta'(z, a_1, b_1, a_2, b_2)$ for $\delta(z)$ and $\delta'(z)$ respectively to show explicitly the failure and repair probabilities of the two stages.

It can be shown that

$$\delta(z) = \frac{1 - r^*}{1 - r^*\sigma^z}, \qquad x_1 \neq x_2,$$
$$= \frac{b_1 + b_2 - b_1b_2}{b_1 + b_2 - b_1b_2 + zb_1b_2(1 + \hat{x})}, \qquad x_1 = x_2 = \hat{x} \tag{51}$$

and

$$\delta'(z) = \frac{\sigma^z(1 - r^*)}{1 - r^*\sigma^z}, \qquad x_1 \neq x_2,$$
$$= \frac{b_1 + b_2 - b_1b_2}{b_1 + b_2 - b_1b_2 + zb_2b_2(1 + \hat{x})}, \qquad x_1 = x_2 = \hat{x} \tag{52}$$

They can be expressed in an alternative way that provides further insight into their meaning.

For other two-stage transfer line models, see Buzacott and Shanthikumar (1993).

4.2. Multiple-Stage Transfer Lines

Consider a transfer line with m stages and $m - 1$ banks of capacities $z_1, z_2, \ldots, z_{m-1}$. The number of cycles of operation before failure and the repair times all have geometric distributions with mean $1/a_i$ and $1/b_i$ respectively for stage i, $I = 1, \ldots, m$. The parameters of stage i are then (a_i, b_i).

4.2.1. Approximation

The basic concept of the approximation approach is the recognition that viewed from any inventory bank the line appears to have two stages, with the arrivals at the bank determined by the upstream stages and the departures from the bank determined by the downstream stages. If the upstream stages are replaced by a single equivalent station and the downstream stages are replaced by a single equivalent station, then, viewed from the bank, the system appears to consist of the bank and two stations. If the equivalent stations have time to failure and repair time distributions for which the two station system can be solved, then the throughput of the system can be determined. The approximation arises because, even when the individual stages have distributions that in a two-station system would be solvable, grouping the upstream or downstream stages results in the grouped stages no longer having the same form of distributions of time to failure and time to repair as their constituent stages. However, in order to use the known two-stage results, it is necessary to approximate the actual distributions by distributions for which the two-stage system has a solution.

The specifics of the approximation procedure will be described for a system where each stage j, $j = 1, \ldots, m$ has geometric operation-dependent time to failure with failure probability a_j and geometric repair time with repair probability b_j. However, the procedure can easily be adapted to the case where the stages have geometric time to failure and identical deterministic repair time $\overline{D}$ with all banks having a capacity such that $z_j = l_j\overline{D}$ where l_j is an integer. Viewed from bank j, the upstream stages will be assumed to have a geometric time to failure with failure probability a_j^U and geometric repair time with repair probability b_j^U, while the downstream stages will be assumed to have geometric time to failure with failure probability a_{j+1}^D and geometric repair time with repair probability b_{j+1}^D. These parameters are given by

$$a_j^U = a_j + (1 - a_j^D)a_{j-1}^U\delta_{j-1,j}, \qquad 2 \leq j \leq m - 1$$
$$b_j^U = a_j^U \Bigg/ \left(\frac{a_j}{b_j} + \frac{a_{j-1}^U}{b_{j-1}^U}\,\delta_{j-1,j}\left(1 - a_j^D\,\frac{b_{j-1}^U}{b_j^D + b_{j-1}^U - b_j^Db_{j-1}^U}\right)\right), \qquad 2 \leq j \leq m$$

$$a_1^U = a_1$$
$$b_1^U = b_1$$

$$a_{j+1}^D = a_{j+1} + (1 - a_{j+1}^U)a_{j+2}^D\delta_{j+2}\delta_{j+2,j+1}, \qquad 0 \leq j \leq m - 2,$$
$$b_{j+1}^D = a_{j+1}^D \Bigg/ \left(\frac{a_{j+1}}{b_{j+1}} + \frac{a_{j+2}^D}{b_{j+2}^D}\,\delta_{j+2,j+1}\left(1 - a_{j+1}^U\,\frac{b_{j+2}^D}{b_{j+1}^U + b_{j+2}^D - b_{j+1}^Ub_{j+2}^D}\right)\right),$$
$$1 \leq j \leq m - 2$$
$$a_m^D = a_m$$
$$b_m^D = b_m$$

where

$$\delta_{j-1,j} = \delta(z_{j-1}, a^U_{j-1}, b^U_{j-1}, a^D_j, b^D_j)$$

and

$$\delta_{j+1,j} = \delta'(z_j, a^U_j, b^U_j, a^D_{j+1}, b^D_{j+1})$$

There are thus $4m - 2$ equations in $4m - 2$ unknowns. Let $(a^U_j, b^U_j, a^D_j, b^D_j, j = 1, \ldots, m)$ be a solution to these equations. Then it can be shown that the efficiency $\eta(z_j,a^U_j,b^U_j,a^D_{j+1},b^D_{j+1})$ of the two stage line is the same for all $j = 1, \ldots, m - 1$. Specifically, we have

$$\eta^O := \frac{1}{1 + a^U_m/b^U_m} = \eta(z_j, a^U_j, b^U_j, a^D_{j+1}, b^D_{j+1})$$
$$= \eta^I := \frac{1}{1 + a^D_1/b^D_1}, \quad j = 1, \ldots, m - 1 \tag{53}$$

Since the equations are nonlinear, they are best solved using an iterative technique, where convergence is checked by determining whether $\eta^I = \eta^O$.

4.2.2. Algorithm 4: Multistage Transfer Line

Step 0: Set $k = 1$, $a^D_j(0) = a_j$, $b^D_j(0) = b_j$, $j = 2, \ldots, m$.

Step 1: Set $a^U_1(k) = a_1$, $b^U_1(k) = b_1$. For $j = 1, 2, \ldots, m - 1$, analyze the two stage transfer line with parameters $(a^U_j(k),b^U_j(k))$ and $(a^D_{j+1}(k - 1), b^D_{j+1}(k - 1))$ and determine $a^U_{j+1}(k)$ and $b^U_{j+1}(k)$.

Step 2: Set $a^D_m(k) = a_m$, $b^D_m(k) = b_m$. For $j = m - 1, m - 2, \ldots, 1$ analyze the two-stage transfer line with parameters $(a^D_{j+1}(k), b^D_{j+1}(k))$ and $(a^U_j(k),b^U_j(k))$ and determine $a^D_j(k)$ and $b^D_j(k)$.

Step 3: Calculate $\eta^O = 1/(1 + a^U_m(k)/b^U_m(k))$ and $\eta^I = 1/(1 + a^D_1(k)/b^D_1(k))$. If $|\eta^O - \eta^I| < \epsilon$ stop, otherwise set $k = k + 1$ and go to step 1.

The convergence of this algorithm is shown in Buzacott and Shanthikumar (1993). Convergence to a solution is usually very rapid, and the solution is usually very close to the exact line efficiency. Table 3 shows exact and approximate results for a number of three-stage lines. There are some situations where convergence of the algorithm is very slow, such as, in some three station systems where the failure probability of the middle stage is much less than that of the outer stages (case 6 in Table 3). If all the b_j are identical and equal to b and the a_j's are sufficiently small that it is unlikely more than one stage is failed at any time, then the above $4(m - 1)$ equations can be reduced to the following $2(m - 1)$ equations by seting $b^U_j = b^D_{j+1} = b$ for $1 \le j \le m - 1$:

$$\begin{aligned} a^U_j &= a_j + a^U_{j-1}\hat{\delta}_{j-1,j}, \qquad 2 \le j - m \\ a^D_{j+1} &= a_{j+1} + a^D_{j+2}\hat{\delta}_{j+2,j+1}, \qquad 0 \le j \le m - 2 \\ a^U_1 &= a_1 \\ a^D_m &= a_m \end{aligned} \tag{54}$$

Here

$$\begin{aligned} \hat{\delta}_{j-1,j} &= \hat{\delta}(z_{j-1}, a^U_{j-1}, b, a^D_j, b) \\ \hat{\delta}_{j+1,j} &= \hat{\delta}'(z_j, a^U_j, b, a^D_{j+1}, b), \\ \hat{\delta}'(z, a_1, b, a_2, b) &= \hat{\delta}(z, a_2, b, a_1, b) \end{aligned} \tag{55}$$

and

$$\hat{\delta}(z, a_1, b, a_2, b) = \frac{1 - a_2/a_1}{1 - \dfrac{a_2}{a_1}\left(\dfrac{a_1 + a_2 - a_1 b}{a_1 + a_2 - a_2 b}\right)^z}$$

TABLE 3 Throughput of Three-Station Systems

Case	a_1	a_2	a_3	b_1	b_2	b_3	z_1	z_2	η exact	η approx
1	0.03	0.05	0.02	0.20	0.20	0.20	15	15	0.777846	0.777759
2	0.01	0.02	0.005	0.20	0.10	0.15	15	3	0.814949	0.814970
3	0.7	0.9	0.6	0.3	0.4	0.9	7	5	0.285463	0.285463
4	0.9	0.05	0.6	0.4	0.4	0.4	10	10	0.307692	0.307692
5	0.001	0.0003	0.005	0.7	0.02	0.3	8	4	0.970335	0.970337
6	0.6	0.04	0.6	0.8	0.4	0.8	9	9	0.565423	0.571365

When $a_1 = a_2$,

$$\hat{\delta}(z, a, b, a, b) = \frac{1}{1 + \dfrac{zb}{2 - b}}$$

The efficiency of the line can be found from $\eta^O = 1/(1 + a_m^U/b)$ or $\eta^I = 1/(1 + a_1^D/b)$, or by determining the efficiency of any of the two stage lines with parameters (a_j^U, b) and (a_{j+1}^D, b) and inventory bank capacity z_j by the efficiency formula

$$\hat{\eta}(z_j, a_j^U, b, a_{j+1}^D, b) = \frac{1}{1 + (a_j^U \hat{\delta}_{j,j+1} + a_{j+1}^D)/b} \tag{56}$$

It is not difficult to show that the same efficiency is obtained for all j. That is,

$$\hat{\eta}(z_j, a_j^U, b, a_{j+1}^D, b) = \hat{\eta}(z_{j+1}, a_{j+1}^U, b, a_{j+2}^D, b), \qquad j = 1, \ldots, m - 2 \tag{57}$$

Therefore, the preceding $m - 2$ equations may be used to replace $(m - 2)$ of those specified above for a_j^U's and a_{j+1}^D's and still solve for $\{(a_j^U, a_{j+1}^D), j = 1, \ldots, m - 1\}$.

5. DYNAMIC JOB SHOPS

The models in this section were originally developed to describe job shops in manufacturing, but they are also applicable to service systems where customers take different routes through the system, depending on their service requirements or diagnosis. So while we will use the language of manufacturing job shops, the reader can easily replace this with comparable language from service applications.

5.1. Open Jackson Queueing Network Model

A single class of jobs arrive at the job shop according to a Poisson process with arrival rate λ. The fraction of jobs that will join machine center i on their arrival is γ_i, $i = 1, \ldots, m$. $(\Sigma_{i=1}^m \gamma_i = 1)$. The fraction of jobs that complete service at machine center i that will directly go to machine center j is p_{ij}. Then $1 - \Sigma_{j=1}^m p_{ij}$ is the fraction of jobs among those completing service at machine center i, that will directly leave the system. Of course, at least for one or more $i = 1, \ldots, m$, $1 - \Sigma_{j=1}^m p_{ij} > 0$ so that all jobs entering the system will eventually leave the system. The service times of jobs at machine center i are i.i.d. exponential random variables with mean $1/\mu_i$, $i = 1, \ldots, m$. All the service times and the arrival times are mutually independent.

The rate at which a job is processed at machine center i when there are n jobs is assumed to be $\mu_i r_i(n) = 0, 1, \ldots$. This allows us to represent, as special cases, single or multiple machines in parallel at the machine center i. Specifically if there are c_i machines in parallel at machine center i, we set $r_i(n) = \min\{n, c_i\}$, $n = 0, 1, \ldots$; $i = 1, \ldots, m$. Leaving this representation as a general function r_i allows one to model the effect of the number of jobs in a machine center on the worker efficiency.

Assume that the job shop uses service protocols such as first come first served or last come first served that are independent of the job service time requirements.

We model this system by a Markovian open queueing network (i.e., Jackson open queueing network) where jobs are routed from one service center to another according to a transfer probability matrix $\mathbf{P} = (p_{ij})_{i,j=1,\ldots,m}$. Let $N_i(t)$ be the number of jobs at machine center i at time t; $i = 1, \ldots, m$ and $\mathbf{N}(t) = \{N_1(t), \ldots, N_m(t)\}$. Then $\mathbf{N}(t)$ is a continuous time Markov process on $\mathfrak{N}_+^m$. Define the stationary distribution of $\mathbf{N}$ by $p(\mathbf{n}) = \lim_{t\to\infty} P\{\mathbf{N}(t) = \mathbf{n}\}$, $\mathbf{n} \in \mathfrak{N}_+^m$. Then it can be shown (see Jackson (1963))

$$p(\mathbf{n}) = \prod_{i=1}^{m} p_i(n_i), \qquad \mathbf{n} \in \mathcal{N}_+^m \tag{58}$$

where if the λ_i, $i = 1, \ldots, m$ are the solution to the set of linear equations

$$\lambda_i = \lambda\gamma_i + \sum_{j=1}^{m} \lambda_j p_{ji} \qquad j = 1, \ldots, m \tag{59}$$

then

$$p_i(n_i) = p_i(0) f_i(n_i) \qquad n_i = 0,1, \ldots, i = 1, \ldots, m \tag{60}$$

$$f_i(0) = 1; \; f_i(n_i) = \frac{f_i(n_i - 1)\lambda_i}{\mu_i r_i(n_i)}, \qquad n_i = 1, 2, \ldots; \; i = 1, \ldots, m \tag{61}$$

$$p_i(0) = 1 \Big/ \sum_{n_i=0}^{\infty} f_i(n_i), \qquad i = 1, \ldots, m \tag{62}$$

Note that λ_j is the rate of job arrivals (including both internal and external arrivals) to the machine center j, $j = 1, \ldots, m$. Furthermore, note that $p_i(.)$ is the same as the stationary distribution of the number of jobs in an $M/M(n)/1$ queueing system, with arrival rate λ_i and state-dependent service rate $\mu_i r_i(n_i)$, when there n_i jobs in it, $n_i = 0,1, \ldots$ Therefore, the results given in Chapter 83 for the $M/M/1$ and $M/M/c$ queueing systems can be directly applied to this queueing network model. Particularly when there is only a single machine at each machine center (i.e. $c_i = 1$, $i = 1, \ldots, m$), we have

$$p(\mathbf{n}) = \prod_{i=1}^{m} (1 - \rho_i)\rho_i^{n_i}, \qquad \mathbf{n} \in \mathcal{N}_+^m \tag{63}$$

where $\rho_i = \lambda_i/\mu_i < 1$, $i = 1, \ldots, m$. In this case, the average number of jobs in machine center i is

$$E[N_i] = \frac{\rho_i}{1 - \rho_i}, \qquad i = 1, \ldots, m \tag{64}$$

and the total number of jobs in the job shop is

$$E[N] = \sum_{i=1}^{m} \frac{\rho_i}{1 - \rho_i} \tag{65}$$

Applying Little's formula, one then obtains the average flow time of an arbitrary job as

$$E[T] = \frac{1}{\lambda} \sum_{i=1}^{m} \frac{\rho_i}{1 - \rho_i} \tag{66}$$

Since $\lambda_i = \lambda v_i$, where v_i is the expected number of visits made to machine center i by an arbitrary job before it leaves the system, (66) can be
rewritten as

$$E[T] = \sum_{i=1}^{m} v_i E[T_i] \tag{67}$$

where

$$E[T_i] = \frac{1}{\lambda_i} \frac{\rho_i}{1 - \rho_i} \tag{68}$$

is the average flow time of an arbitrary job in machine center i each time it visits machine center i, $i = 1, \ldots, m$.

The independence of $N_1, \ldots, N_m$ allows one to compute the variance of the total number of jobs in the job shop. Particularly since $\text{var}(N_i) = \rho_i/(1 - \rho_i)^2$, $i = 1, \ldots, m$, we have

$$\text{var}(N) = \sum_{i=1}^{m} \frac{\rho_i}{(1 - \rho_i)^2} \tag{69}$$

Even though the stationary number of jobs in the machine centers is statistically independent, which we used in obtaining (69), the stationary flow time of arbitrary jobs through the different machine centers is, in general, not independent. Consequently, we cannot obtain a simple expression for the variance of the flow time of an arbitrary job. In Buzacott and Shanthikumar (1993), a good approximation for this variance is given.

5.2. Multiple-Job-Class Open Jackson Queueing Network Model

In some situations, aggregation of all job types into a single job class is unacceptable in a queueing network model. We concentrate on situations where the operation/machine sequence is different for different classes of jobs, but the service requirements of the different classes at any machine center are probabilistically almost the same. In addition, we will assume that jobs are selected for service according to a First-Come-First-Served service protocol.

Class l jobs arrive at the job shop according to Poisson process with rate $\lambda^{(l)}$, $l = 1, \ldots, r$. All these r arrival processes are mutually independent. The fraction of class l jobs that join machine i is $\gamma_i^{(l)}$ ($\Sigma_{i=1}^m \gamma_i^{(l)} = 1$) and the fraction of class l jobs among those class l jobs that complete service at machine center i, that proceed directly to machine center j as a class k job, is $p_{ij}^{(l)(k)}$, $i, j = 1, 2, \ldots; l,k = 1,2, \ldots, r$. For each l ($l = 1, \ldots, r$) we assume that there exist at least one or more i such that $\Sigma_{k=1}^r \Sigma_{i=1}^m p_{ij}^{(l,k)} < 1$ so that class l jobs that enter the system will eventually leave the system. The service time of a class l job at machine center i is exponentially distributed with mean $1/\mu_i$, $i = 1, \ldots, m$. Jobs are served at a rate $r_i(n_i)$ at machine center i when there are n_i jobs, $r_i(0) = 0$, $r_i(n_i) > 0$, $n_i = 1, 2, \ldots$ Note that this service rate is independent of the number of individual classes of jobs, but depends only on the total number. The arrival, service, and job transfers from one machine center to another are mutually independent. Assuming that each job is transferred from one machine center to another and from one class to another, according to a transfer probability matrix $(p_{ij}^{(l)(k)})_{i,j=1,\ldots,m}^{l,k=1,\ldots,r}$ we model this job shop by a multiple-job-class open Jackson queueing network.

Let $N_i(t)$ be the number of jobs at machine center i at time t and $X_{i;j}(t)$ be the class index of the job in the jth position of the queue at machine center i,$j = 1, \ldots, N_i(t)$; $i = 1, \ldots, m$. We assume that the job in the first position is in service, being served at a rate $r_i(n_i)$ when $N_i(t) = n_i$, $i = 1, \ldots, m$. Then $\{\mathbf{X}(t) = (X_{i;j}(t), j = 1, \ldots, N_i(t), i = 1, \ldots, m), t \geq 0\}$ is a continuous-time Markov process. Let $q(\mathbf{x}) = \lim_{t\to\infty} P\{\mathbf{X}(t) = \mathbf{x}\}$ be the stationary probability distribution of $\mathbf{X}$. Then it can be shown that

$$q(\mathbf{x}) = \prod_{i=1}^{m} q_i(\mathbf{x}_i) \tag{70}$$

and

$$q_i(\mathbf{x}_i) = \frac{1}{\Sigma_{n=0}^{\infty} f_i(n)} \prod_{s=1}^{n_i} \frac{\lambda^{(x_{i;s})}}{\mu_i r_i(s)}, \qquad i = 1, \ldots, m \tag{71}$$

with $\lambda_j^{(l)}$, $l = 1, \ldots, r$; $j = 1, \ldots, m$ the solution to

$$\lambda_i^{(l)} = \lambda^{(l)}\gamma_i^{(l)} + \sum_{j=1}^{m} \sum_{k=1}^{r} \lambda_j^{(k)} p_{ji}^{(k)(l)}, \qquad l = 1, \ldots, r; i = 1, \ldots, m \tag{72}$$

and $\lambda_j = \Sigma_{l=1}^r \lambda_j^{(l)}$, $f_i(0) = 1$, and $f_i(n_i) = f_i(n_i - 1)\lambda_i/\mu_i r_i(n)$; $n = 1,2, \ldots, m$.

We can now use (70) to obtain the joint probability distribution of $\mathbf{N}$, the number of jobs of each class at each machine center. Let $p(\mathbf{n}) = P\{\mathbf{N} = \mathbf{n}\}$ and $p_i(\mathbf{n}_i) = P\{\mathbf{N}_i = \mathbf{n}_i\}$. Here $\mathbf{n}_i = (n_i^{(1)}, n_i^{(2)}, \ldots, n_i^{(r)})$ and $n_i^{(l)}$ is the number of class l jobs at machine center i. Then one has

$$p_i(\mathbf{n}_i) = \binom{n_i}{n_i^{(1)}, \ldots, n_i^{(r)}} \prod_{l=1}^{r} \left(\frac{\lambda_i^{(l)}}{\lambda_i}\right)^{n_i^{(l)}} \frac{f_i(n_i)}{\Sigma_{n=0}^{\infty} f_i(n)}, \qquad \mathbf{n}_i \in \Im\!\mathfrak{l}_+^r \tag{73}$$

where $n_i = \Sigma_{l=1}^r n_i^{(l)}$.

Therefore the joint distribution of $\mathbf{N}$ is

$$p(\mathbf{n}) = \prod_{i=1}^{m} \binom{n_i}{n_i^{(1)}, \ldots, n_i^{(r)}} \prod_{l=1}^{r} \left(\frac{\lambda_i^{(l)}}{\lambda_i}\right)^{n_i^{(l)}} \left\{\frac{f_i(n_i)}{\Sigma_{n=0}^{\infty} f_i(n)}\right\}, \qquad \mathbf{n} \in \mathfrak{N}_+^{r\times m} \tag{74}$$

Observe that $f_i(n_i)/\Sigma_{n=0}^{\infty} f_i(n)$, $n_i = 0,1,....$ is the probability distribution of the number of customers in an $M/M(n)/1$ queueing system with arrival rate λ_i and state-dependent service rate $\mu_i r_i(\cdot)$. On the other hand

$$\binom{n_i}{n_i^{(1)},\cdots,n_i^{(r)}} \prod_{l=1}^{r} \left(\frac{\lambda_i^{(l)}}{\lambda_i}\right)^{n_i^{(l)}}$$

is the multinomial probability of choosing $n_i^{(l)}$ of type l items according to a probability $\lambda_i^{(l)}/\lambda_i$, $l = 1, \ldots, r$. Therefore, the joint distribution of the number of jobs at the different machine centers is the same as that in a single-job-class queueing network with arrival rate λ_i to machine center $i = 1, \ldots, m$. We may therefore solve the open queueing network model with single class and obtain the joint distribution of the number of jobs of different classes at different machines using a multinomial sampling with probability $\lambda_i^{(l)}/\lambda_i$ for class l jobs ($l = 1, \ldots, r$) at machine center i ($i = 1, \ldots, m$).

For a machine center that has only one machine, we have

$$p_i(\mathbf{n}_i) = \binom{n_i}{n_i^{(1)},\cdots,n_i^{(r)}} \prod_{l=1}^{r} \left(\frac{\lambda_i^{(l)}}{\lambda_i}\right)^{n_i^{(l)}} (1 - \rho_i)\rho_i^{n_i}, \qquad \mathbf{n}_i \in \mathfrak{N}_+^{r} \tag{75}$$

where $\rho_i = \lambda_i/\mu_i < 1$. For $c_i = +\infty$ we have

$$p_i(\mathbf{n}_i) = \prod_{l=1}^{r} \frac{(\rho_i^{(l)})^{n_i^{(l)}} e^{-\rho_i^{(l)}}}{n_i^{(l)}!}, \qquad \mathbf{n}_i \in \mathfrak{N}_+^{r} \tag{76}$$

where $\rho_i^{(l)} = \lambda_i^{(l)}/\mu_i$, $l = 1, \ldots, r$. The number of jobs of different classes at a machine center with infinitely many machines are independent. In the other cases this need not be true. As we will see next, the marginal distribution of any class of job, say l, at any machine center with a single machine, say i, is exactly the same as that in an $M/M/1$ queue with arrival rate $\lambda_i^{(l)}$ and service rate $\mu_i - \Sigma_{s=1;s\neq l}^{r} \lambda_i^{(s)}$. Then it can be shown that

$$P\{N_i^{(l)} = n_i^{(l)}\} = (1 - \hat{\rho}_i^{(l)})(\hat{\rho}_i^{(l)})^{n_i^{(l)}}, \qquad n_i^{(l)} = 0,1, \ldots \tag{77}$$

where

$$\hat{\rho}_i^{(l)} = \frac{\rho_i^{(l)}}{1 - \rho_i + \rho_i^{(l)}} = \frac{\lambda_i^{(l)}}{\mu_i - \Sigma_{k=1,k\neq l}^{r} \lambda_i^{(k)}}$$

and $\rho_i^{(l)} = \lambda_i^{(l)}/\mu_i$. Therefore, if we are interested only in studying the flow performance of a particular class of job, say l, in a job shop with single or infinite machine centers, all we have to do is redefine the service rates of the single-machine machine center (say i) by $\mu_i - \Sigma_{k=1,k\neq l}^{r} \lambda_i^{(k)}$ and analyze a single open queueing network with only class l jobs and the modified service rates. We cannot, however, do this *job class decomposition* when there are machine centers with a limited number of parallel machines.

We have seen how the number of jobs of different classes in single- and infinite-machine machine centers decomposes. Next we will make an observation about the effect of aggregating the number of jobs at different-infinite machine machine centers. Let $I \subset \{1, \ldots, m\}$ be the set of machine centers that have infinitely many machines. Then one finds that

$$P\{N_i^{(l)} = n_i^{(l)}, l = 1, \ldots, r; i \in I\} = \prod_{i\in I} \prod_{l=1}^{r} \frac{e^{-\rho_i^{(l)}} (\rho_i^{(l)})^{n_i^{(l)}}}{n_i^{(l)}!}$$

Then it is easily verified that

$$P\left\{\sum_{i\in I} N_i^{(l)} = n^{(l)}, l = 1, \ldots, r\right\} = \prod_{l=1}^{r} \frac{e^{-\rho_I^{(l)}} (\rho_I^{(l)})^{n^{(l)}}}{n^{(l)}!} \tag{78}$$

where $\rho_I^{(l)} = \Sigma_{i\in I} \rho_i^{(l)}$, $l = 1, \ldots, r$. Therefore, if we are interested only in the total number of jobs

of different classes in the set I of machine centers, we may aggregate all these $|I|$ machine centers into one infinite-machine machine center.

It should be noted that the results discussed here for the infinite-machine machine centers hold true even if the distributions of the service times are general.

5.2.1. Incorporating Transport and Material Handling in the Jackson-Type Job Shop Model

Consider the material-handling configuration where each link (i,j) that connects machine centers i and $j(i,j = 1, \ldots, m; i \neq j)$ has a sufficient number of transporters or a conveyor system such that when a job completes processing at machine center i, it can immediately begin movement from machine center i to its destination. The average time taken to transfer a job from machine center i to machine center j is $1/\mu_{(i,j)}$, $i,j = 1, \ldots, m$; $i \neq j$. Suppose the job transfers from one machine center to another are represented by a job transfer probability matrix $\mathbf{P} = (p_{ij})_{i,j=1,\ldots,m}$. If we now incorporate the transporters on each link (i,j) as a service center, the job transfer probabilities are $p'_{ii} = p_{ii}$, $p'_{i,(i,j)} = p_{ij}$ and $p'_{(i,j),j} = 1$, $i,j = 1, \ldots, m$; $i \neq j$. All other transfer probabilities are zero. Here (i,j) represents the service center corresponding to the transporters on link (i,j),$i,j = 1, \ldots, m$; $i \neq j$. If we model this system by an open Jackson queueing network, as described earlier, we find that the stationary distribution of the number of jobs in the machine centers (**n**), and the number of jobs in the transporters (**l**), is given by

$$p(\mathbf{n},\mathbf{l}) = \left\{\prod_{i=1}^{m} p_i(n_i)\right\} \prod_{i=1}^{m} \prod_{j=1;j\neq 1}^{m} p_{(i,j)}(l_{(i,j)}), \qquad \mathbf{n} \in \mathfrak{N}_+^m, \mathbf{l} \in \mathfrak{N}_+^{m^2-m} \tag{79}$$

where $p_i(n_i) = f_i(n_i p_i(0)$, $p_i(0) = 1/\Sigma_{n_i=0}^{\infty} f_i(n_i)$, and $f_i(n_i) = \lambda_i f_i(n_i - 1)/\mu_i r_i(n_i)$, $f_i(0) = 1$; $i = 1, \ldots, m$,

$$p_{(i,j)}(l_{(i,j)}) = \frac{e^{-\rho(i,j)} \rho_{(i,j)}^{l_{(i,j)}}}{l_{(i,j)}!}, \quad i,j = 0, 1, \ldots; i \neq j \tag{80}$$

and $\rho_{(i,j)} = \lambda_{(i,j)}/\mu_{(i,j)} = \lambda_i p_{ij}/\mu_{(i,j)}$, $i \neq j, i,j = 1, \ldots, m$. Here n_i is the number of jobs in machine center i and $l_{(i,j)}$ is the number of jobs in transit from machine center i to j, $i \neq j$, $i,j = 1, \ldots, m$. Observe that if $p_{ij} = 0$ then $\rho_{(i,j)} = 0$ and the number of jobs in transit on link (i,j) is also zero, as it should be. If we are only interested in the total number of jobs in transit, we find that

$$P\{l \text{ jobs are in transit}\} = e^{-\hat{\rho}} \hat{\rho}^l / l!, \qquad l = 0,1,... \tag{81}$$

where

$$\hat{\rho} = \sum_{i=1}^{m} \lambda_i \sum_{j=1;j\neq 1}^{m} p_{ij}/\mu_{(i,j)}$$

and $\hat{\rho}$ is the mean number of jobs in transit. Applying Little's law, one sees that the additional average time spent by an arbitrary job in the shop due to material handling is

$$E[\hat{T}] = \frac{1}{\lambda} \sum_{i=1}^{m} \lambda_i \sum_{j=1;j\neq i}^{m} p_{ij}/\mu_{(i,j)} = \sum_{i=1}^{m} \sum_{j=1;j\neq 1}^{m} \frac{v_{(i,j)}}{\mu_{(i,j)}} \tag{82}$$

where $v_{(i,j)}$ is the expected number of times an arbitrary job is moved along the link (i,j), $i \neq j$; $j = 1, \ldots, m$. Now observe that the distribution of the number of jobs at different machine centers given in (79) is independent of the transportation times. Hence we see that the number of jobs at different machine centers is unaffected by the actual transportation time. All it does is to add an additional amount $E[\hat{T}]$ to the average flow time of an arbitrary job.

5.3. General Service Times

The modeling of open queueing networks with general service times, is discussed in Chapter 83, Section 7.2. The basis of the approximation is to assume that the queue length distributions at the different service centers are independent, that is,

$$P\{N_i = n_i, i = 1, \ldots, m\} = \prod_{i=1}^{m} P\{\hat{N}_i = n_i\}, \qquad \mathbf{n} \in \Im_+^M \tag{83}$$

$$E[N] = \sum_{i=1}^{m} E[\hat{N}_i] \tag{84}$$

and

$$E[T] = \sum_{i=1}^{m} v_i E[\hat{T}_i] \tag{85}$$

where N_i is the number of jobs at machine center i, $N = \Sigma_{i=1}^{m} N_i$ is the total number of jobs in the system, and T is the flow time of an arbitrary job. v_i is the average number of visits made to a service center by an arbitrary job.

Each service center i, $i = 1, \ldots, m$ is modeled by a $GI/GI/c_i$ queue. The parameters of this queue will be $(\lambda_i, C_{a}^2, \mu_i, C_{si}^2, c_i)$, where λ_i and $C_{a_i}^2$ are the mean arrival rate and squared coefficient of variation of the time between arrivals at service center i, μ_i and $C_{S_i}^2$ are the mean service rate and squared coefficient of variation of the service time of a job at service center i. Then, using some appropriate approximation, the mean number of jobs at service center i, $\hat{n}_i(\lambda_i, C_{a_i}^2, \mu_i, C_{S_i}^2, c_i)$ and the mean flow time of a job at service center i, $\hat{T}(\lambda_i, C_{a_i}^2, \mu_i, C_{S_i}^2, c_i)$ are estimated.

Given the job transfer probability matrix, $\mathbf{P} = (p_{ij})$, then the λ_i are the solution to the equations

$$\lambda_i = \lambda\gamma_i + \sum_{j=1}^{m} \lambda_j p_{ji}, \qquad i = 1, \ldots, m$$

and $v_i = \lambda_i/\lambda$, $i = 1, \ldots, m$.

The $C_{a_i}^2$ are determined by observing that if $C_{d_j}^2$ is the squared coefficient of variation of departures from service center j, then with probabilistic routing, the stream of jobs from service center j will arrive at service center i with a squared coefficient of variation C_{aj}^2, given by

$$C_{aji}^2 = 1 - p_{ji} + p_{ji} C_{dj}^2$$

Jobs coming from outside the system will arrive in a stream with squared coefficient of variation $C_{a0j}^2 = 1 - \gamma_i + \gamma_i C_a^2$. This job stream will merge with job streams coming from other service centers and from outside the system. Approximately, the squared coefficient of variation of arrivals at service center i, $C_{a_i}^2$, is given by

$$C_{a_i}^2 = \frac{1}{\lambda_i}\left\{\sum_{j=1}^{m} \lambda_j C_{aji}^2 + \lambda\gamma_i C_{a0i}^2\right\} \tag{86}$$

5.3.1. Approximations for $\hat{n}_i$ and $C_{d_i}^2$

In implementing the decomposition approach, a variety of ways of approximating $\hat{n}_i$ and $C_{d_i}^2$ have been used. It can be shown (Buzacott and Shanthikumar 1993) that in a $GI/G/1$ queue, C_d^2 and $E[N]$ are related by

$$C_d^2 = C_a^2 + 2\rho^2 C_S^2 + 2\rho(1 - \rho) - 2(1 - \rho)E[N] \tag{87}$$

so it would seem reasonable to assume that if $c_i = 1$, the approximated $C_{d_i}^2$ and the approximation $\hat{n}_i$ are connected by the same relationship as Eq. (87).

The choice of approximation $\hat{n}_i$ is largely determined by the purpose of the modeling. If the purpose is to arrive at performance predictions that will agree closely with simulation results, then a more complex approximation is used. However, if the purpose of the approximation is to compare a number of different system designs, then simpler approximations are usually quite adequate to get the necessary insight. Whitt (1983) and Shanthikumar and Buzacott (1981) used relatively complex approximations because their purpose was to demonstrate that in many situations the approximate decomposition approach yields remarkably accurate predictions. Harrison and Nguyen (1990) suggest that, in fact, a simple heavy-traffic approximation is often quite adequate. As a general rule, the more there is some randomness in job routings the more accurate the approximations are. The QNA approximations are given in Chapter 83, so we give here are two other approaches:

1. Buzacott and Shanthikumar (1993): For $c = 1$ this sets

$$E[\hat{N}(\lambda, C_a^2, \mu, C_S^2, 1)] = \left\{\frac{\rho^2(1 + C_S^2)}{1 + \rho^2 C_S^2}\right\}\left\{\frac{C_a^2 + \rho^2 C_S^2}{2(1 - \rho)}\right\} + \rho \tag{88}$$

2. Harrison and Nguyen (1990): Assume $c = 1$. Their approximation is

$$E[\hat{N}(\lambda_i, C_{a_i}^2, \mu_i, C_{S_i}^2, 1)] = \frac{\rho_i^2(C_{a_i}^2 + C_{S_i}^2)}{2(1 - \rho_i)} + \rho_i \tag{89}$$

where $\rho_i = \lambda_i/\mu_i$ and they set

$$C_{d_i}^2 = C_{S_i}^2$$

We would, however, recommend setting $C_{d_i}^2$ using Eq. (87), that is,

$$C_{d_i}^2 = (1 - \rho_i^2)C_{a_i}^2 + \rho_i^2 C_{S_i}^2 \tag{90}$$

An even simpler approximation is to set

$$E[\hat{N}(\lambda_i, C_{a_i}^2, \mu_i, C_{S_i}^2, 1)] = \frac{C_{a_i}^2 + C_{S_i}^2}{2(-\rho_i)} + \rho_i \tag{91}$$

and

$$C_{d_i}^2 = C_{S_i}^2 \tag{92}$$

This approximation is particularly useful for gaining insight into alternative ways of allocating tasks to workers in service systems, for example, should workers have a broad range of tasks in a system or should they specialize (Buzacott 1996).

6. FLEXIBLE MACHINING SYSTEMS

The key feature of a flexible machining system that must be represented in queueing models is the limited number of pallets or work holders available for holding jobs in process. This means that the number of jobs in process cannot exceed the number of pallets.

6.1. Single-Class Closed Jackson Queueing Network Model

Consider a flexible machining system consisting of m machine centers $(1, \ldots, m)$ and a load/unload station "0." There are n parts of a single (probably aggregated) class of parts that circulate from one service center to another according to a transfer probability matrix $\mathbf{P} = (p_{ij})_{i,j=0,\ldots,m}$. We will assume that all self transitions have been eliminated so that $p_{ii} = 0$, $i = 1, \ldots, m$. The service requirements of parts at service center i are i.i.d. exponential random variables with mean $1/\mu_i$, $i = 0, \ldots, m$. The sequence of service requirements at the different service centers is mutually independent. The rate at which unit service requirement of a part is processed at a machine center i when there are k parts is assumed to be $r_i(k)$, $k = 1, \ldots, n$; $i = 0, \ldots, m$. This allows us to represent, as special cases, single or multiple servers in parallel at each service center. The parts at each service center are served according to a first-come-first-served service protocol.

Let $N_i(t)$ be the number of parts at service center i at time t, $i = 0, 1, \ldots, m$ and let $\mathbf{N}(t) = \{N_0(t), \ldots, N_m(t)\}$. Then $\{\mathbf{N}(t), t \geq 0\}$ is a continuous-time Markov process on the state space S_n, where

$$\begin{aligned} \mathcal{S}_l &= \{\mathbf{k} : \mathbf{k} \in \mathcal{R}_+^{m+1}, |\mathbf{k}| = l\}, \qquad l = 1, 2, \ldots \\ \mathcal{S}_0 &= \{0, \ldots, 0\} \end{aligned} \tag{93}$$

Define the stationary distribution of $\mathbf{N}$ by $p(\mathbf{k}) = \lim_{t\to\infty} P\{\mathbf{N}(t) = \mathbf{k}\}$, $\mathbf{k} \in S_n$. Then it can be shown that (e.g., see Gordon and Newell (1967))

$$p(\mathbf{k}) = \frac{1}{G(n)} \prod_{i=0}^{m} p_i(k_i), \qquad \mathbf{k} \in \mathcal{S}_n \tag{94}$$

where

$$p_i(k_i) = \frac{v_i^{k_i}}{\prod_{j=1}^{k_i} \mu_i r_i(j)}, \qquad k_i = 0, \ldots, n; \qquad i = 0, \ldots, m \tag{95}$$

v_i, $i = 1, \ldots, m$ is the solution to

$$v_i = \sum_{j=0}^{m} v_j p_{ji}, \; i = 1, \ldots, m \tag{96}$$

with $v_0 = 1$ and

$$G(n) = \sum_{\mathbf{k} \in \mathcal{S}_n} \prod_{i=0}^{m} \frac{v_i^{k_i}}{\prod_{j=1}^{k_i} \mu_i r_i(j)} \tag{97}$$

Let Y_i be a generic random variable representing the stationary distribution of the number of parts in an $M/M(n)/1$ queueing system with arrival rate $\lambda_i = \lambda v_i$ and state-dependent service rate $\mu_i r_i(k_i)$ when there are k_i parts in the system ($k_i = 1, 2, \ldots$). λ can be any value that guarantees the existence of a stationary distribution for the $M/M(n)/1$ queue. For example, if $r_i(k_i) = \min\{k_i, c_i\}$, that is, we have c_i parallel servers at service center i, then we require that $\lambda_i < \mu_i c_i$ or equivalently $\lambda < \mu_i c_i / v_i$. It is easily verified that

$$P\{\mathbf{N} = \mathbf{k}\} = P\{\mathbf{Y} = \mathbf{k} | \, |\mathbf{Y}| = n\}, \qquad \mathbf{k} \in \mathcal{S}_n \tag{98}$$

where $Y_0, \ldots, Y_m$ are independent. It is also clear that the distribution of $\mathbf{Y}$ is the stationary distribution of an open Jackson queueing network with a set of service centers $\{0, \ldots, m\}$, and an arbitrary part visiting service center i, on the average v_i number of times before it leaves the system ($i = 1, \ldots, m$) and the load/unload center twice ($= 2v_0$). The service rate at service center i is $\mu_i r_i(k_i)$ when there are k_i parts for $i = 1, \ldots, m$ and $2\mu_0 r_0(k_0)$ when there are k_0 parts at the load/unload service center. Note that the average load or unload times are $1/2\mu_0$ while the combined load/unload operation (incorporated as a single operation) in the closed queueing network model requires an average of $1/\mu_0$ units of time. The external part arrival rate is λ. Note that this open Jackson network could be our model for the FMS if we had free inflow of raw parts into the system at rate λ. Therefore, the distribution of the number of parts in the closed queueing network model is the same as that in an open queueing network model, provided we observe the open queueing network only when there are a total of n parts in it.

Therefore, the probability distribution of the number of parts in an open queueing network model can be used to compute the probability distribution of the number of parts in the closed queueing network. To do this, we need to compute the convolution of the probability distributions $P\{Y_i = k_i\} = P\{Y_i = 0\} \; (\lambda v_i / \mu_i)^{k_i} / \Pi_{j=1}^{k_i} \; r_i(j)$, $i = 0, \ldots, m$. The following algorithm will compute this convolution and the stationary distribution of the number of parts in the closed queueing network.

6.1.1. Algorithm 5: Convolution Algorithm

Step 1: Set $p_i(0) = 1$, $i = 0, \ldots, m$.

Step 2: For $i = 0, \ldots, m$ and $k = 0, \ldots, n - 1$, set

$$p_i(k + 1) = p_i(k) \frac{v_i}{\mu_i r_i(k + 1)}$$

Step 3: Set $\hat{p}(k) = p_0(k)$, $k = 0, \ldots, n$.

For $i = 1, \ldots, m$ and for $k = 0, \ldots, n$, set

$$\hat{q}(k) = \sum_{l=0}^{k} \hat{p}(l) p_i(k - l)$$

For $k = 0, \ldots, n$, set

$$\hat{p}(k) = \hat{q}(k)$$

Step 4: $p(\mathbf{k}) = 1/\hat{q}(n) \; \Pi_{i=0}^{m} \; p_i(k_i)$, $\mathbf{k} \in \mathcal{S}_n$.

Step 5: Stop.

Let TH(n) be the throughput rate of this closed queueing network. This is the rate at which parts are loaded/unloaded at the load/unload service center, that is, TH(n) = $E[\mu_0 r_0(N_0)]$. Using (98), we see that

$$\begin{aligned} \mathrm{TH}(n) &= E[\mu_0 r_0(Y_0) \mid |\mathbf{Y}| = n] \\ &= \sum_{\mathbf{k} \in \mathcal{S}_n} \frac{\mu_0 r_0(k_0) P\{\mathbf{Y} = \mathbf{k}\}}{P\{|\mathbf{Y}| = \boldsymbol{n}\}}. \end{aligned} \tag{99}$$

Since $\mu_0 r_0(k_0) P\{Y_0 = k_0\} = \lambda P\{Y_0 = k_0 - 1\}$, $k_0 \geq 1$ and $r_0(0) = 0$, we get from (99)

$$\begin{aligned} \mathrm{TH}(n) &= \lambda \sum_{\mathbf{k}' \in \mathcal{S}_{n-1}} \frac{P\{\mathbf{Y} = \mathbf{k}'\}}{P\{|\mathbf{Y}| = n\}} \\ &= \lambda \frac{P\{|\mathbf{Y}| = n - 1\}}{P\{|\mathbf{Y}| = n\}}. \end{aligned} \tag{100}$$

Equivalently,

$$\lambda P\{|\mathbf{Y}| = n - 1\} = \mathrm{TH}(n) P\{|\mathbf{Y}| = n\}, \qquad n = 1, 2, \ldots \tag{101}$$

Hence we see that the total number of parts in the open queueing network is the same in distribution as that in an $M/M(n)/1$ queueing system with arrival rates λ and state dependent service rates TH(n), $n = 1, 2, \ldots$ For the purpose of analyzing the aggregate behavior of the total number of parts in the system, we can aggregate the whole FMS and replace it by an equivalent single-stage server with state dependent service rates equal to the production rates.

Next we will look at another property of the closed queueing network process $\{\mathbf{N}(t), t \geq 0\}$ that will help us to efficiently compute the production capacity of the flexible machining system. Denote by $\mathbf{e}_i$ a vector that is all zeroes except for a one in position i. Let $\tau_n^{(i)}$ be the nth time epoch at which an arrival occurs at service center i. We are interested in

$$p^{(i)}(\mathbf{k}) = \lim_{n \to \infty} P\{\mathbf{N}(\tau_n^{(i)}) = \mathbf{k} + \mathbf{e}_i\}, \qquad \mathbf{k} \in \mathcal{S}_{n-1}$$

the probability distribution of number of parts at different service centers seen by an arbitrary arrival at its arrival epoch to service center i. $\mathcal{S}_{n-1}$ contains all possible states that could be seen by any part excluding itself.

The rate at which an arrival to service center i sees the system state $\mathbf{k}$ at its arrival epoch is $\Sigma_{j=0}^{m} p(\mathbf{k} + \mathbf{e}_j)\mu_j r_j(k_j + 1)p_{ji}$. The rate at which parts arrive at service center i is $\Sigma_{\mathbf{k} \in S_{n-1}} \Sigma_{j=0}^{m} p(\mathbf{k} + \mathbf{e}_j)\mu_j r_j(k_j + 1)p_{ji}$. Then it can be shown that

$$p^{(i)}(\mathbf{k}) = \frac{\prod_{l=0}^{m} \dfrac{v_l^{k_l}}{\prod_{j=1}^{k_l} \mu_l r_l(j)}}{\sum_{\mathbf{k}' \in \mathcal{S}_{n-1}} \prod_{l=0}^{m} \dfrac{v_l^{k_l}}{\prod_{j=1}^{k_l} \mu_l r_l(j)}}, \qquad \mathbf{k} \in \mathcal{S}_{n-1} \tag{102}$$

Observe that $p^{(i)}(\mathbf{k})$ is exactly the same as the stationary distribution of the number of parts in the closed queueing network with a total of $n - 1$ parts. Furthermore, this probability distribution is independent of the service center i. Since we will be using this relationship between the closed queueing network with n and $n - 1$ parts in it to compute the performance measures, we will use an index n to associate the total number of parts to its performance measures. Earlier, we saw how to obtain the joint probability distribution $p(\mathbf{k})$, $\mathbf{k} \in \mathcal{S}_n$. However, in most applications it may be sufficient to compute the marginal probability distribution $p_i(k_i;n)$, $k_i = 0, \ldots, n$; its average (i.e., the average number of parts at service center i), $E[N_i(n)]$, the average sojourn time of an arbitrary part in service center i, $E[T_i(n)]$, $i = 0, \ldots, m$, and the throughput TH(n). Then TH(n) = $\mu_0 E[r_0(N_0(n))]$ is the rate at which parts are being loaded/unloaded at the load/unload service center. Using a simple recursive algorithm over the values of n, we can compute these performance measures with considerably less computational effort than required by the convolution algorithm described earlier.

The rate at which parts arrive at service center i is v_iTH(n). Since the probability that an arrival at service center i sees $k_i - 1$ parts in service center i is $p_i(k_i - 1; n - 1)$, the rate of upcrossings from the compound state $\{\mathbf{k}' : \mathbf{k}' \in \mathcal{S}_n, k_i' = k_i - 1\}$ is $v_i \mathrm{TH}(n) p_i(k_i - 1; n - 1)$. Equating this to the rate of downcrossings ($\mu_i r_i(k_i) p_i(k_i;n)$) into this compound state, we get

$$p_i(k_i; n) = \frac{v_i}{\mu_i r_i(k_i)} \mathrm{TH}(n) p_i(k_i - 1; n - 1), \qquad k_i = 1, \ldots, n$$

$$p_i(0; n) = 1 - \sum_{k_i=1}^{n} p_i(k_i; n), \qquad n = 1, 2, \ldots \tag{103}$$

From (103) and Little's results we find that

$$E[N_i(n)] = \mathrm{TH}(n) \sum_{k_i=0}^{n-1} \frac{(k_i + 1)v_i}{\mu_i r_i(k_i + 1)} p_i(k_i; n - 1) \tag{104}$$

and

$$E[T_i(n)] = \sum_{k_i=0}^{n-1} \frac{k_i + 1}{\mu_i r_i(k_i + 1)} p_i(k_i; n - 1) \tag{105}$$

Since $\Sigma_{i=0}^{m} E[N_i(n)] = n$, we obtain

$$\mathrm{TH}(n) = \frac{n}{\sum_{i=0}^{m} v_i E[T_i(n)]} \tag{106}$$

Equations (103)–(106) provide a recursive relationship for $p_i(\cdot; n)$ and $\mathrm{TH}(n)$. As an initial value for this recursion observe that $p_i(0; 0) = 1$, $i = 0, \ldots, m$. Then $\mathrm{TH}(1) = 1/\Sigma_{i=0}^{m} v_i/\mu_i r_i(1)$. An algorithm to compute these performance measures using this recursion is presented next:

6.1.2. Algorithm 6: Marginal Distribution Analysis Algorithm

Step 1: Set $p_i(0; 0) = 1$, $i = 0, \ldots, m$.

Step 2: For $l = 1, \ldots, n$, compute

$$E[T_i(l)] = \sum_{k_i=0}^{l-1} \frac{k_i + 1}{\mu_i r_i(k_i + 1)} p_i(k_i; l - 1); \qquad i = 0, \ldots, m$$

$$\mathrm{TH}(l) = l \Big/ \left\{ \sum_{i=0}^{m} v_i E[T_i(l)] \right\}$$

$$E[N_i(l)] = v_i \mathrm{TH}(l) E[T_i(l)], \qquad i = 0, \ldots, m$$

$$p_i(k_i; l) = \frac{v_i \mathrm{TH}(l)}{\mu_i r_i(k_i)} p_i(k_i - 1; l - 1), \qquad k_i = 1, \ldots, l; \; i = 0, \ldots, m$$

$$p_i(0; l) = 1 - \sum_{k_i=1}^{l} p_i(k_i; l), \qquad i = 0, \ldots, m$$

Step 3: Stop.

When we have only a single server at a service center i (i.e., $c_i = 1$) the computational effort required for the marginal distribution analysis can be reduced by the following observation $E[T_i(n)] = \{E[N_i(n - 1)] + 1\}/\mu_i$. Particularly if each service center has only a single server (i.e. $c_i = 1$, $i = 0, \ldots, m$), the following mean value analysis algorithm provides an efficient way to compute the system performance measures.

6.1.3. Algorithm 7: Mean Value Analysis Algorithm

Step 1: Set $E[N_i(0)] = 0$, $i = 0, \ldots, m$.

Step 2: For $l = 1, \ldots, n$, compute

$$E[T_i(l)] = \{E[N_i(l - 1)] + 1\}/\mu_i, \qquad i = 0, \ldots, m$$

$$\mathrm{TH}(l) = l \Big/ \left\{ \sum_{i=0}^{m} v_i E[T_i(l)] \right\},$$

$$E[N_i(l)] = v_i \mathrm{TH}(l) E[T_i(l)], \qquad i = 0, \ldots, m$$

Step 3: Stop.

6.1.4. *Properties of the Throughput*

The production rate TH(n) has some useful properties in terms of the influence of the number of servers c_i at service center i, and the workload $\hat{\rho}_i = v_i/\mu_i$ assigned to service center i. To indicate this dependence, the production rate is represented by TH($\hat{\rho}$, c, n)

Property 1: TH($\hat{\rho}$, c, n) is increasing and concave in n.

Property 2: TH($\hat{\rho}$, c, n) is increasing and concave in c_i, $i = 0, \ldots, m$.

Property 3: TH($\hat{\rho}$, c, n) is decreasing and convex in each $\hat{\rho}_i$, $i = 0, \ldots, m$.

Property 4: If $c_0 = c_1 = \cdots = c_m$ then, if work load can be reallocated between service centers, the production rate is maximized by having the same work load at all service centers.

6.2. Modeling the Effects of Dedicated Material-Handling Systems

Suppose we have material-handling systems available for each link connecting any two service centers, such that no queueing delays take place during part movement. The only time expended on moving parts is the travel time. Let $1/\mu_{(i,j)}$ be the average travel time of a part on the link (i,j), $i,j = 0, \ldots, m$. Now, treating such a link (i,j) as a service center with infinitely many servers, one sees that $v_{ij} = v_i p_{ij}$ and $E[T_{ij}(n)] = 1/\mu_{(i,j)}$, $i,j = 0, \ldots, m$. Incorporating this in the MVA algorithm, we get the following algorithm to compute the system performance with material-handling effects.

6.2.1. *Algorithm 8: MVA with Material Handling*

Step 1: Let $E[N_i(0)] = 0$, $i = 0, \ldots, m$.

Step 2: For $l = 1, \ldots, n$, compute

$$E[T_i(l)] = \{E[N_i(l-1)] + 1\}/\mu_i, \qquad i = 0, \ldots, m$$

$$\mathrm{TH}(l) = l \Big/ \left\{ \sum_{i=0}^{m} v_i E[T_i(l)] + \sum_{i=0}^{m} \sum_{j=0}^{m} v_i p_{ij}/\mu_{(ij)} \right\}$$

$$E[N_i(l)] = v_i \mathrm{TH}(l) E[T_i(l)], \qquad i = 0, \ldots, m$$

Step 3: Stop.

6.3. General Single-Class Closed Queueing Network Model

Usually service times will not be exponential. We now present an approximation by which performance measures of closed queueing networks can be found. Because the approximation extends the mean value analysis algorithm, it can only be used if there is just one server at each machine center. For an approximate algorithm that can be used when there are multiple servers at a machine center, see Buzacott and Shanthikumar (1993). The extended MVA algorithm is based on the following: each arrival to service center i will see, at the instant of its arrival, the part in service, if any, requiring an average of $E[S_i^2]/2E[S_i]$ additional processing time to complete its service. This mimics the property of an $M/G/1$ queueing system. With this, and the assumption that an arrival with n parts in the system sees the time average behavior of a system with $n-1$ parts, we have

$$\begin{aligned} E[T_i(n)] = {} & \{E[N_i(n-1)] - v_i \mathrm{TH}(n-1) E[S_i] + 1\} E[S_i] \\ & + v_i \mathrm{TH}(n-1) E[S_i^2]/2, \qquad i = 0, \ldots, m \end{aligned} \tag{107}$$

The above relationship is then applied in the following algorithm.

6.3.1. *Algorithm 9: Extended Mean Value Analysis* (EMVA)

Step 1: Set $E[N_i(0)] = 0$, $i = 0, \ldots, m$; $\mathrm{TH}(0) = 0$.

Step 2: For $l = 1, \ldots, n$, compute

$$\begin{aligned} E[T_i(l)] = {} & \{E[N_i(l-1)] - v_i \mathrm{TH}(l-1) E[S_i] + 1\} E[S_i] \\ & + v_i \mathrm{TH}(l-1) E[S_i^2]/2, \qquad i = 0, \ldots, m \end{aligned}$$

$$\mathrm{TH}(l) = l \Big/ \left\{ \sum_{i=0}^{m} v_i E[T_i(l)] \right\}$$

$$E[N_i(l)] = v_i \mathrm{TH}(l) E[T_i(l)], \qquad i = 0, \ldots, m$$

Step 3: Stop.

6.4. Multiple-Class Model

An FMS is often required to process multiple part types where each part type requires a unique pallet type. Let n_s be the number of pallets for type s parts, $s = 1, \ldots, p$. Suppose the service time distribution for all part types at service center i, $i = 1, \ldots, m$ is the same for all part types and is exponential with mean $1/\mu_i$. Assume that parts are served at a service center in first-come-first-served sequence. Suppose the routing of class s parts between service centers of the FMS is described by the probability matrix $p_{ij}^{(s)}$. Let $v_i^{(s)}$ be the solution to

$$v_i^{(s)} = \sum_{j=0}^{m} v_j^{(s)} p_{ji}^{(s)} \tag{108}$$

with $v_0^{(s)} = 1$, for each $s = 1, \ldots, p$. Then $v_i^{(s)}$ is the expected number of times a class s part visits service center i during its sojourn in the system.

Then again it is possible to show that the queue length distributions are product form and is thus possible to determine the performance measures. If there is just one machine at each service center, then the mean value analysis algorithm can be adapted to find the performance measures. Define $\mathbf{n} = (n_1, n_2, \ldots, n_p)$ and $\mathbf{n} - \mathbf{e}_s = (n_1, n_2, \ldots, n_s - 1, \ldots, n_p)$. Then the expected time that a class s job spends at station i, $E[T_{i,s}(\mathbf{n})]$ and the expected number of jobs of class s at station i, $E[N_{i,s}(\mathbf{n})]$ are related by $E[T_{i,s}(\mathbf{n})] = (E[N_{i,s}(\mathbf{n} - \mathbf{e}_s)] + 1)/\mu_i$. The algorithm is as follows.

6.4.1. Algorithm 10: Multiclass MVA

Step 1: Set $E[N_{i,s}(\mathbf{l})] = 0$, $l_s \leq 0$; $s = 1, \ldots, p$; $i = 0, \ldots, m$.

Step 2: For $l_1 = 0, \ldots, n_1; \ldots; l_p = 0, \ldots, n_p$ repeat step 3.

Step 3: For $i = 0, \ldots, m$ and $s = 1, \ldots, p$, compute

$$E[T_{i,s}(\mathbf{l})] = \{E[N_{i,s}(\mathbf{l} - \mathbf{e}_s)] + 1\}/\mu_i$$

$$\mathrm{TH}_s(\mathbf{l}) = l_s \Big/ \left\{ \sum_{i=0}^{m} v_i^{(s)} E[T_{i,s}(\mathbf{l})] \right\}$$

$$E[N_{i,s}(\mathbf{l})] = v_i^{(s)} \mathrm{TH}_s(\mathbf{l}) E[T_{i,s}(\mathbf{l})]$$

Step 4: Stop.

6.4.2. Properties of the Throughput Rate

The throughput function of the multiple-class closed Jackson queueing network possesses some properties that are, at first, somewhat surprising.

Property 1: $\mathrm{TH}_s(\mathbf{n})$ is increasing in n_s for each s ($s = 1, \ldots, p$).

Property 2: $\mathrm{TH}_l(\mathbf{n})$ *need not* increase (and may *decrease*) in n_s for $s \neq l$, $s, l = 1, \ldots, p$.

Property 3: $\mathrm{TH}(\mathbf{n}) = \sum_{s=1}^{p} \mathrm{TH}_s(\mathbf{n})$ *need not* increase (and may *decrease*) in n_s, $s = 1, \ldots, p$.

Properties 2 and 3 mean that the selection of parts to be processed in an FMS requires extreme care. Note that these properties are in part due to the assumption of first-come-first-served sequence at service centers, so their effect can be reduced by more sophisticated scheduling.

6.4.3. General Service Time Distributions

It is possible to heuristically adapt the multiclass MVA algorithm for situations where the service times are not exponential. Let $E[N_{i,s}(\mathbf{l})]$ be the average number of class s parts at service center i when the total number of class s parts in the network is l_s, $s = 1, \ldots, p$. Let $\mathbf{l} + \mathbf{e}_s = (l_1, \ldots, l_{s-1}, l_s + 1, l_{s+1}, \ldots, l_p)$. Also let $E[T_{i,s}(\mathbf{l})]$ be the average flow time of a class s part through service center i and $\mathrm{TH}_s(\mathbf{l})$ be the throughput rate of class s parts. Then the rate at which parts arrive at service center i is

$$\lambda_i(\mathbf{l}) = \sum_{s=1}^{p} v_i^{(s)} \mathrm{TH}_s(\mathbf{l}), \qquad i = 0, \ldots, m$$

The basic idea is to adapt the equation describing the expected time a class s part spends at station i as follows:

$$E[T_{i,s}(\mathbf{l} + \mathbf{e}_s)] = E[S_i^{(s)}] + \sum_{\hat{s}=1}^{p} (E[N_{i,\hat{s}}(\mathbf{l})] - \lambda_i^{(\hat{s})}(\mathbf{l})E[S_i^{(\hat{s})}(\mathbf{l})])E[S_i^{(\hat{s})}] + \lambda_i(\mathbf{l})\frac{E[S_i^2(\mathbf{l})]}{2}, \qquad i = 0, \ldots, m \tag{109}$$

where

$$E[S_i(\mathbf{l})] = \frac{1}{\lambda_i(\mathbf{l})}\sum_{\hat{s}=1}^{p} \lambda_i^{(\hat{s})}(\mathbf{l})E[S_i^{(\hat{s})}], \qquad i = 0, \ldots, m$$

and

$$E[S_i^2(\mathbf{l})] = \frac{1}{\lambda_i(\mathbf{l})}\sum_{\hat{s})=1}^{p} \lambda_i^{(\hat{s})}(\mathbf{l})E[S_i^{(\hat{s})})^2], \qquad i = 0, \ldots, m$$

are the first and second moments of the service time of an arbitrary part at service center i. Since $\Sigma_{i=0}^{m} E[N_{i,s}(\mathbf{l} + \mathbf{e}_s) = \mathrm{TH}_s(\mathbf{l} + \mathbf{e}_s)\ \Sigma_{i=0}^{m} v_i^{(s)}\ E[T_{i,s}(\mathbf{l} + \mathbf{e}_s)] = l_s + 1$, one has

$$\mathrm{TH}_s(\mathbf{l} + \mathbf{e}_s) = \frac{l_s + 1}{\sum_{i=0}^{m} v_i^{(s)}E[T_{i,s}(\mathbf{l} + \mathbf{e}_s)]}. \tag{110}$$

The above equations now provide a recursive scheme to compute $\mathrm{TH}_s(\mathbf{n})$, $s = 1, \ldots, p$ starting with $\mathrm{TH}_s(\mathbf{0}) = 0$, $s = 1, \ldots, p$. The following algorithm computes the approximate performance measures.

6.4.4. Algorithm 11: Extended Multiclass MVA

Step 1: Set $E[N_{i,s}(\mathbf{l})] = 0$, $\mathrm{TH}_s(\mathbf{l}) = 0$, $-1 \le l_{\hat{s}} \le 0$, $s, \hat{s} = 1, \ldots, p$; $i = 0, \ldots, m$.
Step 2: For $l_1 = 0, \ldots, n_1$; $l_2 = 0, \ldots, n_2$; $\ldots$; $l_p = 0, \ldots, n_p$ repeat step 3.
Step 3: For $i = 0, \ldots, m$; $s = 1, \ldots, p$ and $l_s \le n_s - 1$, compute

$$\lambda_i^{(s)}(\mathbf{l}) = v_i^{(s)}\mathrm{TH}_s(\mathbf{l})$$

$$\lambda_i(\mathbf{l}) = \sum_{s=1}^{p} \lambda_i^{(s)}(\mathbf{l})$$

and $E[T_{i,s}(\mathbf{l} + \mathbf{e}_s)]$ by (109). Then compute $\mathrm{TH}_s(\mathbf{l} + \mathbf{e}_s)$ by (110) and

$$E[N_{i,s}(\mathbf{l} + \mathbf{e}_s] = v_i^{(s)}\mathrm{TH}_s(\mathbf{l} + \mathbf{e}_s)/E[T_{i,s}(\mathbf{l} + \mathbf{e}_s)]$$

Step 4: Stop.

7. PRODUCTION COORDINATION

We now consider queueing models that can be used to model the control of workflow in manufacturing and logistic systems. What is different about these models is that it is necessary to model both the flow of information and the flow of material and how information is transformed into material or parts. So although the underlying model is usually that of a multiclass open or closed queue, the models have some unusual features. We consider manufacturing and logistic systems consisting of a number m of cells or stages. We restrict our discussion to the situation where each cell produces just one type of part and where the cells are arranged in series 1, 2, . . . , m and thus part flow is $1 \rightarrow 2 \rightarrow \cdots \rightarrow m$. When cell j completes processing, it puts completed parts in store j. Store j also serves as the input store to cell $j + 1$. If cell $j + 1$ were separated geographically from cell j, then it would be necessary to add in a transport cell between j and $j + 1$. Final demand is met from store m. When store m is empty then demand can not be met immediately. We consider only the case where unmet demand is backlogged. We are interested in using queueing models to find the service level, that is, the fraction of demand met from stock p_{ND}, or the expected delay in meeting demand $E[\Delta]$. We will ignore any possibility of batching and assume that work flow and work release is controlled on a part-by-part basis.

7.1. Base Stock Control

In base stock control there are target stock levels z_j set for the store j. If a demand occurs, information about the occurrence of the demand is immediately sent to each cell. On receipt of the information, a request for a part is sent to the input store, store $j - 1$. If no parts are available, then requests would queue until a part becomes available. Once the part is removed from store $j - 1$, it is then released to the production cell for processing. On completion of processing, the completed part is put in store j. Denote by $N_j(t)$ the number of parts released to the production cell j at time t and not yet completed, $N_{j-1}^{(r)}(t)$ the number of requests for parts waiting at the input store $j - 1$, and $N_j^{(e)}(t)$ the number of parts that would be required to bring the inventory in store j up to z_j. If store j is empty and there is a backlog of waiting requests of size $N_j^{(r)}(t)$, then

$$N_j^{(e)}(t) = z_j + N_j^{(r)}(t), \qquad N_j^{(r)}(t) = 1, 2, \ldots$$

In a base stock system the immediate signaling of all demands to all cells means that

$$N_j^{(e)}(t) \equiv N_j(t) + N_{j-1}^{(r)}(t)$$

Suppose demands occur as a Poisson stream with parameter λ. Then the rate at which requests for parts will arrive at store $j - 1$, $j = 2, \ldots, m + 1$, is λ. In fact, because the requests at different stores are triggered by the same demand, these requests at different cells will be correlated. Our approximation ignores this.

The basis of the base stock model is to model the behavior of each cell and each store and to take into account the way in which they are linked.

7.1.1. Cell Model

The cell j queueing model represents the arrival of the demand at the cell, its transmittal as a request to the input store $j - 1$, the release of the part from the input store to the cell j, and the delivery of the part from the cell to the output store j. That is, we model how the arriving information on the occurrence of a demand is converted to a finished part in the output store. The queueing model consists of two queues in series, where the first queue is where the requests queue at store $j - 1$ and the second queue is where parts queue for processing in cell j. We are also interested in the arrivals of finished parts at the output store j, although parts do not queue to enter the store. Since the input stream to the first queue is Poisson with parameter λ, we will approximate its output stream of releases to the cell by a Poisson stream with parameter λ. So if the cell has single or multiple servers or some network of servers, it can be modeled as an open Jackson queueing network and its stationary queue length distribution $p\{N_j = n\}$, $n = 0, 1, \ldots$, found.

Requests that arrive at store $j - 1$ when it is not empty do not queue and the part can be immediately released to cell j. However, requests that arrive when the store is empty have to wait until a part arrives at the store from cell $j - 1$. It follows that the request queue characteristics will be determined from the inventory shortfall in the store

$$p\{N_{j-1}^{(r)} = n\} = p\{N_{j-1}^{(e)} = n + z_{j-1}\} \qquad n = 1, 2, \ldots$$

$$p\{N_{j-1}^{(r)} = 0\} = \sum_{u=0}^{z_{j-1}} p\{N_{j-1}^{(e)} = u\} \tag{111}$$

Since we assume that the queue lengths at the two queues are product form, we have that

$$p\{N_j^{(e)} = n\} = \sum_{u=0}^{n} p\{N_{j-1}^{(r)} = u\} p\{N_j = n - u\}, \qquad n = 0, 1, \ldots \tag{112}$$

At store 1 we have that

$$p\{N_1^{(r)} = n\} = p\{N_1 = n + z_1\}, \qquad n = 1, 2, \ldots$$

$$p\{N_1^{(r)} = 0\} = \sum_{u=0}^{z1} p\{N_1 = u\}$$

Thus, by considering store 1, store 2, . . . , store m, in sequence using Eqs. (111) and (112) we can determine the $p\{N_j^{(r)} = n\}$, $n = 0, 1, \ldots$; $j = 1, 2, \ldots, m$. Hence,

$$p_{ND} = p\{N_m^{(r)} = 0\}$$

$$E[\Delta] = \sum_{n=0}^{\infty} np\{N_m^{(r)} = n\} \tag{113}$$

7.1.2. Single Server in Each Cell

If each cell consists of a single server with exponential processing time having mean $1/\mu_j$, $j = 1, \ldots, m$ then it is possible to develop a very simple recursive calculation. Let $\rho_j = \lambda/\mu_j$, and we also assume that the ρ_j, $j = 1, \ldots, m$, are all different. We postulate then that

$$p\{N_j^{(r)} = n\} = \sum_{u=1}^{j} c_u^{(j)} \rho_u^n (1 - \rho_u), \qquad n = 1, 2, \ldots; j = 1, \ldots, m$$

$$p\{N_j^{(r)} = 0\} = 1 - \sum_{u=1}^{j} \rho_u c_u^{(j)}, \qquad j = 1, \ldots, m$$

It follows that

$$c_u^{(j)} = c_u^{(j-1)} \rho_u^{z_j+1} \frac{(1 - \rho_j)}{\rho_u - \rho_j}, \qquad u = 1, 2, \ldots, j-1; j = 2, \ldots, m$$

$$c_j^{(j)} = \rho_j^{z_j} \left(1 - \sum_{u=1}^{j-1} c_u^{(j-1)} \rho_u \frac{(1 - \rho_j)}{\rho_u - \rho_j}\right), \qquad j = 2, \ldots, m$$

$$c_1^{(1)} = \rho_1^{z_1} \tag{114}$$

Equations (114) define a recursive scheme by which the $c_u^{(j)}$ can be determined.

The service level measures are then given by

$$p_{ND} = 1 - \sum_{u=1}^{m} \rho_u c_u^{(m)} \tag{115}$$

$$E[\Delta] = \sum_{u=1}^{m} c_u^{(m)} \frac{\rho_u}{1 - \rho_u} \tag{116}$$

Table 4 compares the approximations with simulation results for p_{ND} and $E[\Delta]$, respectively, for a system with $z_1 = 2$ and $z_2 = 2$ for a variety of values of ρ_1 and ρ_2. The approximation tends to underestimate p_{ND} and overestimate $E[\Delta]$ because of the assumption that arrivals at the machine queue are Poisson. In fact, they are less variable than Poisson. The approximation can be extended to systems where demand is notified in advance (Buzacott et al. 1992), that is, to MRP-like systems.

7.2. Kanban Control

In kanban control, there are k_j kanban cards associated with cell j, $j = 1, \ldots, m$. A cell j kanban card waits in store j until it is triggered by the arrival at the store of a cell $j + 1$ kanban card. The cell j kanban card then moves back to store $j - 1$, where it waits until parts are available. As soon as parts are available, then the kanban card and the associated part are released to cell j for processing.

TABLE 4 p_{ND} and $E[\Delta]$ for a Two-Cell Base Stock System with $z_1 = z_2 = 2$

		ρ_1					ρ_1			
ρ_2	p_{ND}	0.3	0.5	0.7	0.9	$E[\Delta]$	0.3	0.5	0.7	0.9
0.3	Sim.	0.903	0.850	0.682	0.314	Sim.	0.045	0.122	0.657	5.63
	approx.	0.899	0.840	0.670	0.298	approx.	0.047	0.131	0.702	6.21
0.5	Sim.	0.747	0.705	0.567	0.259	Sim.	0.258	0.349	0.950	6.07
	approx.	0.739	0.687	0.544	0.240	approx.	0.264	0.375	1.016	6.66
0.7	Sim.	0.512	0.487	0.396	0.181	Sim.	1.15	1.25	1.90	7.22
	approx.	0.502	0.465	0.366	0.160	approx.	1.17	1.31	2.04	7.86
0.9	Sim.	0.195	0.188	0.155	0.071	Sim.	6.94	7.05	7.76	13.3
	approx.	0.188	0.172	0.135	0.059	approx.	7.32	7.51	8.35	14.4

From Buzacott et al. 1992. Reproduced with permission of Springer-Verlag.

On completion of processing by cell j, the finished part and the kanban move to store j, where they will wait until the next cell $j + 1$ kanban arrives. That is, movement of a cell j kanban is from store j to store $j - 1$ to cell j to store j again. Thus the queueing model of a kanban controlled system consists of a multiclass queue with $m + 1$ classes of customers. Class j customers represent the cell j kanbans, $j = 1, \ldots, m$, and class m+1 customers represent customer demands. Class j customers circulate in the closed loop: store $j \rightarrow$ store $j - 1 \rightarrow$ cell $j \rightarrow$ store j, $j = 1, \ldots, m$; while class $m + 1$ customers enter the system, go to store m, and then leave the system. Assume that customer demand is Poisson with parameter λ. It will also be assumed that the cell consists of a single machine, and service times in the cells have exponential distributions with parameter μ_j for cell j, $j = 1, \ldots, m$.

To determine the system performance, it is necessary to model the circulation of class j customers and to model the interrelationship between the service of class j customers at store j and the service of class $j + 1$ customers at store j.

7.2.1. Store Model

Consider store j. Store inventory increases when a part and its associated stage j kanban arrive from cell j. Store inventory decreases when a stage $j + 1$ kanban arrives at the store and triggers the release of a part to cell $j + 1$. The maximum backlog of unmet demands at store j is k_{j+1} so, defining $n_j^{(e)}$ as the inventory shortfall, the inventory position at store j is $k_j - n_j^{(e)}$ with $n_j^{(e)} = 0, 1, \ldots, k_j, \ldots, k_j + k_{j+1}$. For a given $n_j^{(e)}$, the quantities $n_j^{(p)}$ and $n_j^{(r)}$ are determined by

$$n_j^{(p)} = \begin{cases} k_j - n_j^{(e)} & n_j^{(e)} = 0, 1, \ldots, k_j \\ 0 & n_j^{(e)} = k_j + 1, \ldots, k_j + k_{j+1} \end{cases}$$

and

$$n_j^{(r)} = \begin{cases} 0 & n_j^{(e)} = 0, 1, \ldots, k_j \\ n_j^{(e)} - k_j & n_j^{(e)} = k_j + 1, \ldots, k_j + k_{j+1} \end{cases}$$

Note that $n_j^{(r)} \times n_j^{(p)} \equiv 0$. $n_j^{(r)}$ is the number of stage $j + 1$ kanbans waiting at store j, while $n_j^{(p)}$ is the number of stage j kanbans waiting at the store (which is identical to the number of parts in the store).

Let $\lambda_{j:u}(n_j^{(p)})$ be the rate at which stage j kanbans arrive at store j from cell j when there are $n_j^{(p)}$ stage j kanbans at store j. Similarly, let $\lambda_{j:d}(n_j^{(r)})$ be the rate at which stage $j + 1$ kanbans arrive at store j from store $j + 1$ when there are $n_j^{(r)}$ stage $j + 1$ kanbans at store j.

It follows that, for a given $n_j^{(e)}$ such that $0 \le n_j^{(e)} \le k_j - 1$, the rate of increase to $n_j^{(e)} + 1$ is $\lambda_{j:d}(0)$ while, for $n_j^{(e)}$ such that $1 \le n_j^{(e)} \le k_j$, the rate of decrease to $n_j^{(e)} - 1$ is $\lambda_{j:u}(k_j - n_j^{(e)})$. Similarly, for $n_j^{(e)}$ such that $k_j \le n_j^{(e)} \le k_j + k_{j+1} - 1$, the rate of increase to $n_j^{(e)} + 1$ is $\lambda_{j:d}(n_j^{(e)} - k_j)$, and, for $n_j^{(e)}$ such that $k_j + 1 \le n_j^{(e)} \le k_j + k_{j+1}$, the rate of decrease is $\lambda_{j:u}(0)$.

It follows that the probability distribution of $n_j^{(e)}$ is given by

$$p_j(n_j^{(e)}) = \begin{cases} \dfrac{\prod_{v=n_j^{(e)}}^{k_j-1} \lambda_{j:u}(k_j - v - 1)}{\lambda_{j:d}(0)^{k_j - n_j^{(e)}}} p_j(k_j) & n_j^{(e)} = 0, 1, \ldots, k_j - 1 \\ \dfrac{\prod_{v=0}^{n_j^{(e)} - k_j b - 1} \lambda_{j:d}(v)}{\lambda_{j:u}(0)^{n_j^{(e)} - k_j}} p_j(k_j) & n_j^{(e)} = k_j + 1, \ldots, k_j + k_{j+1} \end{cases} \tag{117}$$

and $p_j(k_j)$ is determined by $\sum_{n_j^{(e)}=0}^{k_j+k_{j+1}} p_j(n_j^{(e)}) = 1$.

When a stage $j + 1$ kanban arrives at store j from store $j + 1$ then it will find no other stage $j + 1$ kanbans waiting if, just prior to its arrival, $n_j^{(e)} \le k_j$. However, even though it finds no stage $j + 1$ kanbans waiting, it will have to wait for a part if, just prior to its arrival, $n_j^{(e)} = k_j$. Hence $q_{j:u}$, the probability that an arriving stage $j + 1$ kanban finds no other stage $j + 1$ kanban waiting yet has to wait itself is given by

$$q_{j:u} = p_j^+\{n_j^{(e)} = k_j | 0 \le n_j^{(e)} \le k_j\}$$

where p_j^+ denotes the probabilities at instants of arrival of stage $j + 1$ kanbans. We assume that $p_j^+(n_j^{(e)}) = p_j(n_j^{(e)})$ and set

$$q_{j:u} = \frac{p_j(n_j^{(e)} = k_j)}{\sum_{u=0}^{k_j} p_j(n_j^{(e)} = u)}$$

Once there are waiting stage $j + 1$ kanbans $n_j^{(e)} > k_j$, so the rate at which they will be served will be given by $\mu_{j:u} = \lambda_{j:u}(0)$, irrespective of the number of waiting stage $j + 1$ kanbans.

Similarly, considering the arrivals of stage j kanbans at store j from cell j, the probability $q_{j:d}$ that an arriving stage j kanban finds no other stage j kanban waiting, yet has to wait itself, is given by

$$q_{j:d} = p_j^+\{n_j^{(e)} = k_j | k_j \le n_j^{(e)} \le k_{j+1}\}$$

where p_j^+ denotes the probabilities at the instants of arrival of stage j kanbans. Again we assume that $p_j^+(n_j^{(e)}) = p_j(n_j^{(e)})$ and so

$$q_{j:d} = \frac{p_j(n_j^{(e)} = k_j)}{\sum_{u=k_j}^{k_j+k_{j+1}} p_j(n_j^{(e)} = u)}$$

Since $n_j^{(e)} < k_j$ if there are waiting stage j kanbans, the rate at which waiting stage j kanbans will be served will be given by $\mu_{j:d} = \lambda_{j:d}(0)$, irrespective of the number of waiting stage j kanbans.

7.2.2. Cell Model

Assume that the cell consists of a single server who serves customers at rate μ_j. Alternatively, with minor change in the analysis, the cell can be any open queueing network with a product form solution. Then the closed queueing network representing the circulation of stage j kanbans (or class j customers) will have a product form solution with

$$p(n_j, n_j^{(p)}, n_{j-1}^{(r)}) = K.\left(\frac{1}{\mu_j}\right)^{n_j}\left(\frac{1}{\mu_{j:d}}\right)^{n_j^{(p)}}\left(\frac{1}{\mu_{j-1:u}}\right)^{n_{j-1}^{(r)}} f_p(n_j^{(p)}) f_r(n_{j-1}^{(r)}), \qquad n_j + n_j^{(p)} + n_{j-1}^{(r)} = k_j \tag{118}$$

and

$$f_p(\ell) = \begin{cases} 1, & \ell = 0 \\ q_{j:d} & \ell = 1, \ldots, k_j \end{cases}$$

$$f_r(\ell) = \begin{cases} 1, & \ell = 0 \\ q_{j-1:u} & \ell = 1, \ldots, k_j \end{cases}$$

and K is a normalizing constant

Class $m + 1$ customers only queue at store m, so we have

$$p(n_m^{(r)}) = K\left(\frac{1}{\mu_{m:u}}\right)^{n_m^{(r)}} f_r(n_m^{(r)}), \qquad n_m^{(r)} = 0, 1, \ldots \tag{119}$$

Also note that $\mu_{m:d} = \lambda$.

7.2.3. Connection between Store Model and Cell Model

Lastly, it is necessary to find the $\lambda_{j:u}(n_j^{(p)})$ and the $\lambda_{j:d}(n_j^{(r)})$. To find $\lambda_{j:u}(n_j^{(p)})$ we determine the throughput of the two service center closed queue consisting of store $j - 1$ and cell j. Hence,

$$\lambda_{jw:u}(n_j^{(p)}) = \frac{G_u^*(k_j - n_j^{(p)} - 1)}{G_u^*(k_j - n_j^{(p)})}, \qquad n_j^{(p)} = 0, 1, \ldots, k_j - 1 \tag{120}$$

where

$$G_u^*(\ell) = \sum_{n_{j-1}^{(r)}=0}^{\ell}\left(\frac{1}{\mu_j}\right)^{\ell - n_{j-1}^{(r)}}\left(\frac{1}{\mu_{j-1:d}}\right)^{n_{j-1}^{(r)}} f_r(n_{j-1}^{(r)})$$

Similarly, by determining the throughput in the closed queue consisting of cell $j + 1$ and store $j + 1$, we have

$$\lambda_{j:d}(n_j^{(r)}) = \frac{G_d^*(k_{j+1} - n_j^{(r)} - 1)}{G_d^*(k_{j+1} - n_j^{(r)})}, \qquad n_j^{(r)} = 0, 1, \ldots, k_{j+1} - 1 \tag{121}$$

where

TABLE 5 Comparison of Approximation and Simulation for p_{ND} for a Kanban System with $k_1 = k_2 = 2$

			ρ_1				
			0.1	0.3	0.5	0.7	0.9
ρ_2	0.1	Approx.	0.990	0.982	0.937	0.796	0.447
		sim.	0.990	0.981	0.927	0.750	0.350
	0.3	Approx.	0.910	0.898	0.846	0.707	0.378
		sim.	0.911	0.902	0.847	0.677	0.301
	0.5	Approx.	0.750	0.737	0.678	0.530	0.201
		sim.	0.752	0.744	0.692	0.532	0.182
	0.7	Approx.	0.510	0.496	0.435	0.282	***
		sim.	0.515	0.507	0.454	0.300	***
	0.9	Approx.	0.190	0.177	0.117	***	***
		sim.	0.196	0.187	0.131	***	***

***: unstable.

From Buzacott and Shantikumar © 1993. Reprinted by permission of Prentice-Hall Inc., Upper Saddle River, NJ.

$$G_d^*(\ell) = \sum_{n_{j+1}^{(p)}=0}^{\ell} \left(\frac{1}{\mu_{j+1}}\right)^{\ell - n_{j+1}^{(p)}} \left(\frac{1}{\mu_{j+1:d}}\right)^{\mathbf{n}_{j+1}^{(p)}} f_p(n_{j+1}^{(p)})$$

A recursive scheme can then be developed to solve the equations. Set all $\lambda_{j:d}(u) = \lambda$, then begin by observing that all $\lambda_{1:u}(v) = \mu_1$. Determine $q_{1:u}$, then $\lambda_{2:u}(v)$, $q_{2:u}$ and so on, until stage $m + 1$ is reached. Then, for stage m, determine $q_{m:d\}\text{ and }\lambda m:d}(w)$, then $q_{m-1:d}$, $\lambda_{m-1:d}(w)$, and so on. Repeat this downstream and upstream iterative process until the parameters converge.

7.2.4. Performance Measures

Define $\rho_{mu} = \lambda / \mu_{m:u}$. Then

$$p_{ND} = \frac{(1 - q_{m:u})(1 - \rho_{mu})}{1 - (1 - q_{m:u})\rho_{mu}} \tag{122}$$

and

$$\lambda E[\Delta] = \frac{\rho_{mu}}{1 - \rho_{mu}} (1 - p_{ND}) \tag{123}$$

Tables 5 and 6 compare the performance measures calculated using this approximation for a two-stage system with $k_1 = k_2 = 2$ for a variety of different values of ρ_1 and ρ_2.

TABLE 6 Comparison of Approximation and Simulation for $E[\Delta]$ for a Kanban System with $k_1 = k_2 = 2$

			ρ_1				
			0.1	0.3	0.5	0.7	0.9
ρ_2	0.1	Approx.	0.001	0.003	0.029	0.242	2.20
		sim.	0.001	0.006	0.067	0.549	5.418
	0.3	Approx.	0.039	0.047	0.102	0.376	2.52
		sim.	0.039	0.046	0.127	0.689	6.064
	0.5	Approx.	0.251	0.276	0.419	1.02	6.73
		sim.	0.251	0.266	0.402	1.23	10.5
	0.7	Approx.	1.14	1.22	1.68	3.90	***
		sim.	1.14	1.19	1.55	3.70	***
	0.9	Approx.	7.30	7.97	13.49	***	***
		sim.	6.94	7.45	11.4	***	***

***: unstable

From Buzacott and Shantikumar © 1993. Reprinted by permission of Prentice-Hall Inc., Upper Saddle River, NJ.

CONCLUSIONS

This chapter has focused on using queueing models for performance analysis. It is largely based on the authors' book (Buzacott and Shanthikumar 1993). Other books focusing on performance models of manufacturing systems are Viswanadham and Narahari (1992) and Altiok (1997). Models can also be used for determining the optimal control and scheduling of manufacturing and service systems. See Gershwin (1994) and the papers in Yin and Zhang (1996) for numerous examples.

Acknowledgements

John Buzacott's research was supported by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

Altiok, T. (1997), *Performance Analysis of Manufacturing Systems,* Springer, New York.

Buzacott, J. A. (1996), "Commonalities in Reengineered Business Processes," *Management Science,* Vol. 42, pp. 768–782.

Buzacott, J. A. (2000), "Service System Structure," *International Journal of Production Economics*, Vol. 68, No. 1, pp. 15–27.

Buzacott, J. A., and Shanthikumar, J. G. (1994), "Safety Stock versus Safety Time in MRP Controlled Production Systems," *Management Science,* Vol. 40, pp. 1678–1689.

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

Buzacott, J. A., Liu, X.-G., and Shanthikumar, J. G. (1995), "Multistage Flow Line Analysis with the Stopped Arrival Queue Model," *IIE Transactions,* Vol. 27, pp. 444–455.

Buzacott, J. A., Price, S. M., and Shanthikumar, J. G. (1992), "Service Level in Multistage MRP and Base Stock Controlled Production Systems," in *New Directions for Operations Research in Manufacturing,* G. Fandel, T. Gulledge, and A. Jones, Eds., Springer, Berlin, pp. 445–463.

Daley, D. J., Kreinin, A. Y., and Trengove, C. D. (1992), "Inequalities Concerning the Waiting Time in Single-Server Queues: A Survey," in *Queueing and Related Models,* U. N. Bhat and I. V. Basawa, Eds., Clarendon Press, Oxford, pp. 177–223.

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

Gordon, W. J., and Newell, G. F. (1967), "Closed Queueing Systems with Exponential Servers," *Operations Research,* Vol. 15, pp. 254–265.

Harrison, J. M., and Nguyen, V. (1990), "The QNET Method for Two-Moment Analysis of Open Queueing Networks," *QUESTA,* Vol. 6, pp. 1–32.

Jackson, J. R. (1963), "Job-Shop-Like Queueing Systems," *Management Science,* Vol. 10, pp. 131–142.

Schmenner, R. W. (1986), "How Can Services Businesses Survive and Prosper?," *Sloan Management Review,* Vol. 27, pp. 21–32.

Shanthikumar, J. G., and Buzacott, J. A. (1981), "Open Queueing Network Models of Dynamic Job Shops," *International Journal of Production Research,* Vol. 19, pp. 255–266.

Silvestro, R., Fitzgerald, L., Johnston, R., and Voss, C. (1992), "Towards a Classification of Service Processes," *International Journal of Service Industry Management,* Vol. 3, pp. 62–75.

Viswanadham, N., and Narahari, Y. (1992), *Performance Modeling of Automated Manufacturing Systems,* Prentice Hall, Englewood Cliffs, NJ.

Whitt, W. (1983), "The Queueing Network Analyser," *Bell System Technical Journal,* Vol. 62, pp. 2779–2815.

Yin, G. G., and Zhang, Q., Eds. (1996), *Mathematics of Stochastic Manufacturing Systems,* American Mathematical Society, Providence, RI.