Manufacturing Flow Line Systems: A Review of Models and Analytical Results

Yves Dallery
Laboratoire MASI (UA 818, CNRS)
Université Pierre et Marie Curie
4, Place Jussieu, 75252 Paris Cedex 05

Stanley B. Gershwin
Laboratory for Manufacturing and Productivity
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Abstract

The most important models and results of the manufacturing flow line literature are described. These include the major classes of models (asynchronous, synchronous, and continuous); the major features (blocking, processing times, failures and repairs); the major properties (conservation of flow, flow rate-idle time, reversibility, and others); and the relationships among different models. Exact and approximate methods for obtaining quantitative measures of performance are also reviewed. The exact methods are appropriate for small systems. The approximate methods, which are the only means available for large systems, are generally based on decomposition, and make use of the exact methods for small systems. Extensions are briefly discussed. Directions for future research are suggested.

Keywords: Manufacturing Flow Line Systems, Blocking, Failures, Modelling, Performance Evaluation, Analytical Methods, Exact Analysis, Approximate Analysis.

1 Introduction

1.1 Goals of the Paper

Manufacturing flow line systems consist of material, work areas, and storage areas. Material flows from work area to storage area to work area; it visits each work and storage area exactly once in a fixed sequence; there is a first work area through which material enters and a last work area through which it leaves the system. The times that parts spend in work areas are random and this is the only source of randomness. This randomness may be due to random processing times, random failure and repair events, or both. Storage areas can hold only a finite amount of material. Machines are never allowed to be idle while they have parts to work on and space in which to put parts they have worked on. Manufacturing flow lines are also called **transfer lines** and **production lines.** In this paper, we mainly use the term 'manufacturing flow lines' or simply **flow lines**. The work areas are usually called **machines**. Storage areas are often called **buffers.** The material in most cases consists of **discrete parts.** There is only a single kind of material in the system. Each piece of material travels the same sequence of machines and buffers, but each may experience different delays at each point in the system. Figure 1 depicts a five-machine flow line. A major example of the use of transfer lines is in the high volume production of metal parts of automobiles, but flow lines can be found throughout manufacturing industry. In the language of queuing theory, a flow line can be represented as a **finite buffer, tandem queueing system**. In that case, machines are called **servers,** storage areas are called **buffers,** and discrete parts are called **customers,** or **jobs.**

Our purposes are to survey the most widely known methods and publications in this area; to summarize the most important results and conjectures; to organize the great deal of work that has been done; to show relationships among models; and to offer some opinions. We will try to emphasize those papers that are most influential, those papers that are most well-known, or those papers that (in our opinion) should be. Like all fields, this one has fuzzy boundaries. We will try to focus on papers that are clearly in the flow line/transfer line/production line literature, and avoid those that belong in the much larger general queuing theory domain.

There are many different kinds of flow lines, and many different kinds of models in the literature. The great variety of models in part reflects the variety of different kinds of systems; in part it reflects the fact that different models lend themselves to analysis more or less easily for different purposes. In what follows, we present or quote many mathematical results on the behavior of these systems. The simple structure of a flow line permits very strong statements in some cases. Some of these statements (like conservation of flow) are quite general, and can be thought of as applying to actual systems. Others (such as methods for calculating production rates) are specific to individual models.

Notation There have been many authors in this field, and almost as many different sets of notation. The present authors have therefore given up the notion of satisfying everyone, and have chosen their own earlier notation, with some modifications and compromises. Consequently, the squares in Figure 1 represent machines and the circles represent buffers. Machines are numbered from 1 to K, where K is the number of machines in the system. There are $K-1$ buffers, and the buffer between Machines M_i and M_{i+1} is $B_{i,i+1}$. Parts flow from outside the system to Machine

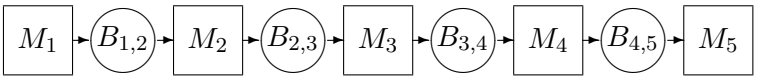

Figure 1: Five-Machine Flow Line

M_1, then to Buffer $B_{1,2}$, then to Machine M_2, then to Buffer $B_{2,3}$, and so on up to Machine M_K after which they leave the system. All the parts have to be processed on all the machines. A great deal of additional notation is defined throughout the paper.

Historically, this work has been aimed mostly at manufacturing systems. For that reason, there is a great emphasis on machine failures as the source of randomness. The goal has been primarily to calculate the maximum rate of flow of material through a production line. The maximum flow rate is often called **production rate, efficiency** (in some models), or **throughput**. Thus, it is assumed that whenever a machine can do an operation, it does. Deliberate idleness is not considered in these models. Other performance measures are also important, especially the average amount of material that is found in the buffers.

Whenever Machine M_i processes material, it reduces the level of Buffer $B_{i-1,i}$ and it increases the level of Buffer $B_{i,i+1}$. On the other hand, when Machine M_i fails or takes an especially long time to process a part, and its neighbors work normally, the level of Buffer $B_{i-1,i}$ tends to increase and the level of Buffer $B_{i,i+1}$ tends to decrease. If that persists, Buffer $B_{i-1,i}$ might become full or Buffer $B_{i,i+1}$ might become empty. In that case, one of the neighbors of M_i is not able to operate; either M_{i-1} is **starved** or M_{i+1} is **blocked** (and is thus **idle**). Production is then reduced because time is wasted. The **isolated production rate** of a machine is the rate that it would operate at if it were not in a system with other machines and buffers.

The production rate of a line is limited in two ways. (1) The throughput can be no greater than that of the machine with the smallest isolated production rate. When the machines are very different in their isolated production rates, the speeds of all but the slowest are largely wasted. (2) The unsynchronized disruptions that cause buffers to be empty or full also waste machine capability. Buffers become empty or full because machines fail or take long times to process material *at different times*. If all machines could be perfectly synchronized, not only in performing operations, but also in failing and getting repaired, buffers would not affect flow. It is the lack of synchronization that causes machines to be starved or blocked, and thus to lose the opportunity to work.

Models, Reality, Mathematics, and Engineering Like all mathematical models, the models in the flow line literature are compromises between fidelity to reality and tractability. All of engineering requires the creative use of results that are based on simplifications of reality, and the design of production systems is not an exception. It is not possible to prove a theorem on the bounds of errors between a model and reality, since it is not possible to fully describe reality. Thus, in spite of the apparent restrictions on the class of systems we are considering, these results may be widely applicable. For example, although we have assumed that there is only

a single part type, these methods may be usable for systems with many part types. If several parts are produced in a line, and they require different lengths of time at each machine, the material may be treated as a single part type with random operation times. The distribution of the processing time then represents the differences of the processing times of the different parts and the randomness of the mixture of parts.

In addition, while we follow the literature and distinguish between reliable and unreliable systems, it is sometimes useful to think of a failure as a long processing time, which occurs at random. Consequently, unreliable systems may be analyzed by methods designed for reliable systems with random operation times, and vice versa.

1.2 Major Features and Properties of Real Manufacturing Flow Lines

Transfer and production lines are of great economic importance. Thus, much of the literature that we survey has practical value, as well as academic interest. In this section, we informally discuss some features of real systems. Mathematical models of these phenomena are defined more precisely in Section 2. Much of the literature is aimed at developing ways of treating the more intractable features.

1.2.1 Synchronous/Asynchronous

Most real systems are unsynchronized. That is, the machines are not constrained to start or stop their operations at the same instant. Even when machines have fixed, equal cycle times (the times required for operations), the presence of buffers between them allows them to start and stop independently, as long as the intermediate buffers are neither empty nor full. In some applications, the machines are not machines at all, but people. In others, the operation times cannot be fixed (for example when the parts are different, but treated as a single type). Finally, uncertain failure and repair times can lead to unsynchronized operation times.

Consequently, **asynchronous** systems form an important class of mathematical models in the literature. However, it is very difficult to treat asynchronous systems with deterministic operation times. This difficulty is generally met in one of three ways: (1) Asynchronous systems are usually modeled with random operation times that have exponential, phase-type, or other tractable probability distribution. (2) **Synchronous** systems are defined, in which operation times are assumed to be deterministic and equal, and when machines are not under repair, they start and stop at the same instant. Alternatively, one may view these models as having time discretized, and it is not important when events occur during the time intervals; by convention, they are treated as though they occur at the beginnings or at the ends of the intervals. (3) **Continuous** material systems are defined; see Section 1.2.5.

1.2.2 Saturated/Non-saturated

Material arrives at and leaves from a factory in a variety of different ways. It is always possible for raw material to be absent, or for the means of removal of finished goods to fail. However, in the literature, it is almost always assumed that the first machine is never starved and the last is never blocked. Such models are called **saturated** models. Saturated models are of interest because a saturated model is appropriate for addressing the most important performance issue of

a flow line, the maximal (average) number of parts that can be produced per unit of time. This quantity is referred to as the **production rate** of the flow line. However, the behavior of a flow line under a given input process and/or output process is also of interest. Thus, some authors, to represent the uncertain arrival and departure processes, add a buffer upstream of the first machine, with random arrivals to it, or a buffer downstream of the last machine, with random departures from it. Often these buffers are infinite, while the others are finite. Such models are called **unsaturated** models. Another approach is simply to declare that the first machine of the model represents the arrival process, and the second machine of the model corresponds to the first machine of the real system. That is, an unsaturated system can equivalently be represented by a saturated model. Because of the predominance of saturated models in the literature, and because of the relationship between saturated and unsaturated models, we concentrate on saturated systems in this paper.

1.2.3 Blocking and Starvation and Decoupling

The function of a buffer is to decouple machines. If a machine is subject to a disruption (a failure or a long operation time), the machine upstream can still operate until the upstream buffer fills up, and the machine downstream can still operate until the downstream buffer becomes empty. The larger the buffers, the longer before the filling or emptying occur, and the larger the production rate. Zero buffers, or pairs of machines that have no storage space between them, have the greatest coupling; and infinite buffers, or storage areas that are never filled, have the least. (Infinite buffers allow coupling when they become empty.)

1.2.4 Failures

Some models of flow lines have machines that can fail. When a failure occurs, a machine may not process any material, so the buffer upstream cannot lose material and the buffer downstream cannot gain material. A variety of assumptions about the conditions under which failure may occur, the time until a failure starts, the time that a failure lasts, and so forth, are considered in the literature. In this paper, we call systems in which machines can fail **Flow Lines with Unreliable Machines (FLUMs)**. Systems in which machines cannot fail are called **Flow Lines with Reliable Machines (FLRMs)**.

In FLRMs, all the randomness is due to the variability of the processing times. In FLUMs, some randomness is due to the failures of the machines and, in some models, some randomness may be due to variability of the processing times.

An important focus of the literature, and of this paper, are the up- and down-time distributions, the probability distributions of the time between a repair and the next failure, and of the time between a failure and the following repair. The most common assumption, and the most mathematically tractable, is exponential. Reality is not always so convenient, so the literature describes a variety of ways of treating non-exponential distributions for some classes of systems.

1.2.5 Discrete/Continuous

The literature described here is most often directed at manufacturing systems with discrete parts. That is, individual parts are treated, and each requires a non-zero, finite amount of

time at each machine. On the other hand, systems that treat continuous material share some characteristics with these systems: in both, machines can fail, and finite buffers can become empty or full and thereby propagate disturbances and reduce production rates. Continuous models, in addition to describing real systems with continuous material, can also approximate discrete systems.

1.2.6 Realistic Up-, Down-, and Operation Times

Transfer lines are artificial systems that are built for economic purposes. As a consequence, they have certain characteristics that are important for researchers to consider when developing models and approximation techniques. For example, in most real systems, the machines do not differ greatly from one another in their production rates. This is because the production rate of the system is limited by the slowest machine, and any investment in machines that are much better than the slowest is wasted. This is made more precise, and other realistic characteristics are described, in Section 3.10.

1.2.7 Operating Policy

In all models surveyed, machines are not allowed to be idle if they can be operated. That is, whenever a machine is neither blocked nor starved, it is used for an operation. Buzacott (1982) demonstrates that this is the optimal operating policy for a two-machine line when the system production rate is the performance measure. He points out that other policies, such as keeping the buffer level as close as possible to some intermediate value (to avoid blocking and starvation), have been used in practice.

There are good reasons for using other policies, however. Maximizing production rate does not take inventory costs into account. When inventory is expensive, it may be optimal to keep the buffer level close to an intermediate value, and to use the buffer size to limit the deviation from that ideal level. Even still, it is useful to study systems operated in this way to determine their maximum possible production rates.

1.2.8 Non-Perishability

In the literature we survey, the material in buffers is assumed to be non-perishable. That is, it does not decay or lose value, no matter how long it waits.

1.3 Other Features

There are manufacturing systems that differ from those described in Section 1.2, but which are close enough so that the methods and characteristics surveyed in this paper should, to some extend, be extendible to them. They include systems with:

machines in parallel Systems are built with machines in parallel for two reasons: either to achieve a greater production rate or to achieve a greater reliability. The first case is often observed when some operation is inherently much slower than the others. The second case is encountered when some machine is much less reliable than the others.

assembly operations In a flow line, each machine feeds a single buffer, and each buffer feeds a single machine. In assembly systems, however, two or more buffers can feed a single machine. The machine takes one part from each upstream buffer, and assembles a part from them. (This can be generalized in a variety of ways, including disassembly.)

pallets Some systems require parts to be fitted onto pallets or fixtures before they are allowed to enter. In some cases, the fixtures allow for very precise location of holes. Because the number of pallets is limited, parts must sometimes wait before they can be processed, even when the first machine is operational and not blocked. From the point of view of the pallets, such systems are closed loops. In fact, one can calculate performance measures by ignoring the parts and modeling only the movement of pallets.

1.4 Review of Reviews

Because of the economic and academic interest in this area, it has generated a great deal of literature, starting in the early 1950's. That literature, in turn, has generated a large number of reviews, which we review here. Because this survey emphasizes the most recent approaches and results, we do not cover all papers that have been devoted to flow lines. Many other references are listed or described in these reviews. In addition, excellent surveys can be found in theses, including those by Ammar (1980), Anderson (1968), Boxma (1977), Buzacott (1967a), Dattatreya (1978), De Koster (1988a), Dudick (1979), Jafari (1982), Liu (1990), Schick (Schick and Gershwin, 1978), Sheskin (1974), Wiley (1981), as well as the monograph of Newell (1979).

Buzacott (1967a) describes the earliest Russian work. Because this work is difficult to obtain, and to translate, we quote from Buzacott (1967a):

> It is not known when buffer stocks were first used to improve the efficiency of an automatic transfer line. It seems to have been about 1946 in Russia. The earliest theoretical papers were published there (Erpsher, 1952; Vladzievskii, 1952 and 1953).
>
> Vladzievskii's work is important as he was the first author to use probability theory to explain the behaviour of automatic transfer lines. In the 1953 paper he used a Markov process approach to solve the case of two identical stages with identical exponential repair time distributions separated by a fixed capacity buffer... Vladzievskii has subsequently written a book on automatic transfer lines (1958 — referred to in Yu Retsker and Bunin, 1964)....
>
> Yu Retsker and Bunin (1964) gives curves based on Vladzievskii's work which enable the economic optimum number of sections into which a line should be divided to be found.

Koenigsberg (1959) begins his review by saying that the production line "has been all but neglected in the annals of operations research and management science." This problem has been substantially remedied since then, as evidenced by the size of this paper and its reference list. He says that "Three major problems in the design and operation of production lines are concerned with (a) the number of stages in the line, (b) the location of bunkers or pulsating stores [buffers], (c) the size of these pulsating stores." Tools for the solution of these problems did not appear

until the 1980's. They are discussed in Section 5. Koenigsberg describes a number of different approaches and some systems that were in use in industry.

Buxey, Slack, and Wild (1973) survey a wider variety of phenomena than are found in most of the papers described here. Papers they surveyed covered line balancing, flexibility (in those days called *mixed-model* production), human factors, parallel stations, allocation of part types to production lines, and the "launching" of work into lines. They consider conveyor belt systems, and they survey studies of the effect of belt speed. They also survey the literature to that date on the effects of buffer stocks.

Buzacott and Hanifin (1978a) introduce the concepts of single station and total line failures, and operation dependent and time dependent failures. (See Section 2.1.3.) They provide simple formulas to calculate the production rate of a line without buffers and with either time dependent and operation dependent failures. They describe the work of Vladzievskii (1953) (which is available only in Russian) and Sevast'yanov (1962) and other, more recent papers. They point out that it would be easy to include the effects of total line failures on any of these models. Finally, they compared the performance prediction of one of Buzacott's models with a simulation based on real data, and they conclude that there are significant differences, which they attribute to the non-memoryless behavior of the repair and failure times. Buzacott and Hanifin (1978b) discuss the state of the art in transfer line design and modeling. They describe such physical and mechanical issues as the transfer mechanism, shunt versus series banks (which determine whether the material in buffers is moved according to FIFO or LIFO), and the design of the line to reduce cycle time, failure frequencies, and downtime duration. They discuss and critique both the practice of simulation as a tool for the design of lines, and the existing analytic models.

Perros (1984) is simply a list of 75 relevant papers on queueing networks with blocking.

Smunt and Perkins (1985) focus on "unpaced assembly lines with stochastic task times" — roughly, what we call asynchronous flow lines with reliable machines. They are particularly interested in line design, the problem of locating and sizing buffers, and allocating tasks to stations. They review many simulation papers and they perform their own simulations to test Hillier and Boling's "bowl phenomenon." They conclude that it is "highly situation specific."

Awate and Sastry (1987) survey much of the transfer and flow line literature. They review the solution methods of most of the important papers. Gun (1987) is a systematic description of 23 of the major papers in this field. For each paper, the model, the performance measure, and the method are briefly sketched. Perros (1988) describes the literature of two-node queuing networks with blocking. Because this review is restricted to small systems, it is able to report on many analytic solutions. It is restricted to asynchronous models. Perros (1989) surveys the literature on queueing networks with blocking. It includes models with reliable or unreliable machines, having tandem or more general topologies. He considers models useful for different applications: computer systems, communication networks, and production systems. Most of the paper deals with approximate techniques. Onvural (1990) surveys closed queuing networks with finite buffers. This paper is more concerned with computer systems than production systems, and, like Perros (1988), it emphasizes asynchronous models. A variety of blocking mechanisms and equivalences among network types are described. (See Section 2.1.1.)

Perros and Altiok (1989) is the proceedings of a conference on queuing networks with blocking. It contains many papers on a variety of topics in this area.

1.5 Outline

In Section 2 we present the major classes of mathematical models of flow lines. We examine the most important properties of flow lines, and the relationships among the models, in Section 3. Methods that analyze systems exactly are explained in Section 4. Because of the finiteness of the buffers, only special systems have exact solutions. Approximate methods are shown in Section 5. The most important methods are decompositions, in which large systems are broken into a set of small systems. Extensions are considered in Section 6. Conclusions and directions for further research are presented in Section 7. Coxian and phase-type distributions, which are used extensively throughout the paper, are described in the appendix. Also, some mathematical claims are proved in the appendix.

2 Flow Line Models

In this section, we introduce three major classes of models that have been considered for the analysis of tandem production lines. We describe assumptions that are made in most of the literature. Some exceptions are summarized in Section 6.

2.1 Asynchronous Models

2.1.1 Blocking Issues

All real buffers have finite capacity. It is convenient to define the **intermediate storage capacity** between Machines M_i and M_{i+1}, $C_{i,i+1}$, to be the total number of parts that can be stored *between* the two machines. We define the **capacity** or **size** $N_{i,i+1}$ of Buffer $B_{i,i+1}$ to include the space on Machine M_{i+1}. It satisfies $N_{i,i+1} = C_{i,i+1} + 1$. Let $\boldsymbol{N} = (N_{1,2}, ..., N_{K-1,K})$ denote the **buffer capacity vector**. These quantities are constant system parameters. We also define the **buffer level** $n_{i,i+1}$ to be the random variable that indicates the number of parts in Buffer $B_{i,i+1}$ at any time, including the part on Machine M_{i+1}, if any. It satisfies

$$0 \leq n_{i,i+1} \leq N_{i,i+1} \tag{1}$$

Since the buffers have finite capacity, blocking may occur. Different types of blocking mechanisms are of interest: **blocking-after-service** and **blocking-before-service** (Perros, 1989). Blocking-after-service (**BAS**) is also referred to as **type-1 blocking** (Onvural and Perros, 1986), **manufacturing blocking**, **production blocking**, **transfer blocking**, and **non-immediate blocking** (Gun and Makowski, 1989). BAS blocking occurs if, at the instant of completion of a part on Machine M_i, the downstream buffer, $B_{i,i+1}$, is full. In that case, the part stays on the machine until a space is available in Buffer $B_{i,i+1}$. During this time the machine is prevented from working and is said to be **blocked**. When a space becomes available in the downstream buffer, the part is immediately transferred and the machine can start processing another part, if any.

Blocking-before-service (**BBS**), is also referred to as **type-2 blocking** (Onvural and Perros, 1986), **communication blocking**, **service blocking**, and **immediate blocking** (Gun and Makowski, 1989). A machine can start processing a part only if there is a space available in

the downstream buffer. Otherwise, it has to wait until a space becomes available. Machine M_i is said to be **blocked** when Buffer $B_{i,i+1}$ is full. BBS is further classified according to whether the position (space) on the machine may be occupied while the machine is blocked or not. These two cases are referred to (Perros, 1989) as **BBS-PO** (**blocking before service with position occupied** while the machine is blocked) and **BBS-PNO** (**blocking before service with position non-occupied** while the machine is blocked). Most often, production lines operate under the BAS mechanism, and therefore most authors assume BAS.

Thus, when a machine is blocked, it is prevented from working. A machine may also be prevented from working because it has no material to work on. This phenomenon is **starvation**. In the case of BBS, starvation corresponds to the situation where the upstream buffer is empty. That is, Machine M_i is starved if $n_{i-1,i} = 0$. In the case of BAS, starvation corresponds to the situation where *either* the upstream buffer is empty, *or* it contains a single part whose processing has already been completed. The second condition corresponds to the case where the part cannot be transferred because the machine is blocked. In this case, although the buffer is not empty, the machine has no part to work on. (We note, however, that some authors define starvation simply as the situation where the upstream buffer is empty.)

A machine is said to be **idle** if it is either starved or blocked. A machine may be simultaneously starved and blocked.

Following the majority of papers in the literature, we assume that the first machine, M_1, is never starved. That is, there are always parts at the input of the system. Also, we assume that the last machine, M_K, is never blocked. There are always spaces for Machine M_K to deliver its parts. In other words, we only consider saturated models. See Section 1.2.2.

The issue of blocking definition was first raised by Altiok and Stidham (1982) who criticized a two-machine model of Gershwin and Berman (1981) that assumed BBS-PO. They pointed out that in a manufacturing system, there is no reason for the first machine to stop until it has completed an operation and there is no room for the completed part. The more recent literature on flow line models with blocking has paid a great deal of attention to blocking mechanisms. Actually, although BAS is likely to be encountered more often, there also exist flow lines operating under BBS assumptions.

2.1.2 Processing Times

The time required for a machine to perform an operation on a part is called the **processing time** or **operation time** or sometimes **cycle time**. This processing time may be either a constant or a variable. In the later case, the processing times at a machine are usually assumed to be random variables having a common distribution. Moreover, it is usually assumed that successive processing times are independent of one another. In other words, processing times are i.i.d. random variables.

It is also usually assumed that processing times at different machines are independent of one another. Deterministic (constant) processing time is just a special case of this. Other typical distributions commonly used are **exponential distributions, geometric distributions, Coxian distributions, and phase-type distributions** (Kleinrock, 1975; Neuts, 1981). Since these distributions are of great importance when analyzing flow lines, a brief review of Coxian and phase-type distributions is given in the Appendix.

2.1.3 Failures and Repairs

In some systems, machines are prone to failures. When a failure occurs, the machine must be repaired and is then unavailable for processing parts. A machine is said to be **operational** if it is up and is said to be **working** if it is operational and not idle (neither starved nor blocked). Two major types of failures have been considered in the literature: **operation dependent failures** (ODF) and **time dependent failures** (TDF) (Buzacott and Hanifin, 1978a).

ODFs are failures that are related to the processing of parts and thus can only occur when the machine is working. On the other hand, TDFs are not related to the processing of parts and thus can occur at any time, including when a machine is idle. In transfer lines that perform high-volume metal-cutting operations, such as for the automobile industry, ODFs are mainly due to mechanical causes (like tool breakage or motor burnout) while TDFs are mainly due to failures of electronic systems, such as controllers. In most production lines, most failures are ODFs (Buzacott and Hanifin, 1978a). As a result, most authors assume ODFs, and unless we explicitly state otherwise, we assume that failures are operation dependent.

It is generally assumed that uptimes and downtimes are i.i.d. random variables. (In reality, failures among different machines may not be independent, for example, when a poorly cast metal part — with hard spots — causes excess wear on all the tools in a transfer line.) A **downtime** (or repair time) of a machine corresponds to the *time* from the instant of a failure of the machine to the instant of the next repair.

On the other hand, there are two different ways of measuring the **uptime** (or time to failure). In the first, the uptime corresponds to the total working *time* of the machine between the instant of the last repair to the instant of the next failure. The working time corresponds to the time where the machine is busy processing parts and does not include idle times (when the machine is either starved or blocked). The cause of failure is related to the time that machine has been busy processing parts. A typical example is the wear of tools. In this case, the distribution of uptimes is a continuous distribution.

In the second, the uptime corresponds to the total *number of parts* produced by the machine from the instant of the last repair of the machine to the instant of the next failure. The cause of failure is related to the number of operations that the machine has performed. A typical example is a failure of the loading/unloading mechanism of parts on the machine. In that case, the distribution of uptimes is a discrete distribution.

We refer to these two failure types as **time-ODFs** and **number-ODFs**, respectively. When uptimes are much larger than processing times, there is little difference between them.

In the case of TDFs, the uptime corresponds to the total *time* (including working and idle time) of the machine between the instant of the last repair to the instant of the next failure. In this case, the distribution of uptimes is a continuous distribution. There is no counterpart of the number-ODF concept.

We need to describe more precisely what happens when a failure occurs. First, in terms of storage, two cases can be considered. Either the part stays on the machine during the repair time, or it is moved back in the intermediate storage area. (The second alternative requires a special treatment when the intermediate storage area is full.) Secondly, we need to define what happens when the machine is repaired. Either the part can be reworked or it cannot. If it cannot, it is thrown away (or **scrapped**). If it can, either the work resumes exactly at the point

it stopped, or the total operation has to be performed again. Very few papers have considered scrapping. They are discussed briefly in Section 6.2.

As for processing times, deterministic, exponential, geometric, Coxian, and phase-type are commonly used distributions. Most authors assume that if several machines are down at the same time, the repair process of each machine is not affected by the repair processes of the others. This means that the repair time of a machine is the same whether or not there are other machines currently under repair. The failures we just considered are referred to as **single machine failures** (Buzacott and Hanifin 1978a). Other failures that may occur in a production line are **total line failures** (Buzacott and Hanifin, 1978a). A typical example is a failure of the power system which forces the whole line to stop. Total line failures can easily be handled (Buzacott and Hanifin, 1978a) and, as a result, we only consider single machine failures.

2.1.4 Buffer Behavior

When a part is transferred from Buffer $B_{i-1,i}$ to Buffer $B_{i,i+1}$, $n_{i-1,i}$ goes down by 1 and $n_{i,i+1}$ goes up by 1. In the case of BBS, this happens at the instant at which Machine M_i completes a part. Indeed, in this case, there must be a space available in Buffer $B_{i,i+1}$ for Machine M_i to be working. As soon as the machine completes its operation, the part can be transferred into the downstream buffer.

For BAS, this is also true as long as the downstream buffer is not full at the instant of processing completion. Otherwise, the part cannot be transferred immediately. The transfer will occur as soon as a space is available in Buffer $B_{i,i+1}$. This transfer will therefore either occur at the instant of processing completion of machine M_{i+1}, or it will be delayed if this machine is blocked.

An important feature of BAS is that simultaneous transfers can occur. Indeed, suppose that at some instant, Buffers $B_{i,i+1}$, $B_{i+1,i+2}$, ..., $B_{j-1,j}$ are full and Machines M_i, M_{i+1}, ..., M_{j-1} have completed their operations and are therefore blocked. The unblocking of all these machines will occur at the instant at which Machine M_j completes its operation. At this instant, a simultaneous transfer of parts will take place in all the buffers. The resulting state is such that all Buffers $B_{i,i+1}$, $B_{i+1,i+2}$, ..., $B_{j-1,j}$ are again full. However, now all Machines M_i, M_{i+1}, ..., M_{j-1} are busy working on new parts.

An issue related to buffer behavior is the **transfer time** of parts through the buffers. In real systems, a part that is released to a buffer by the upstream machine is not immediately available for the downstream machine. It takes some time for the part to be moved throughout the buffer. It is the case, for instance, in automated transfer lines where the buffer consists of a conveyor. This may or not have a significant impact on the behavior of transfer lines depending on the relative values of transfer times and processing times. Most authors assume that a part which is transferred into a buffer by its upstream machine is immediately available for the downstream machine. In other words, the transfer time through the buffer is negligible. Non-zero transit time are briefly discussed in Section 6.5.

2.2 Synchronous Models

There is a distinction between **synchronous** and **discrete time** models, but that distinction is not always observed. A synchronous model is one in which events may only *occur* at certain discrete times. A discrete time model is one whose behavior is only *described* at certain discrete times. For example, a model in which operations, failures, and repairs may start and end at times $t_j = j\Delta$ is a synchronous model. On the other hand, a model in which operations, failures, and repairs may start and end at any time, but the changes in system state are only represented at times $t_j = j\Delta$ is a discrete time model. The distinction is important if, in the discrete time model, the transition equations or performance measures are somehow adjusted to account for the differences between the times that events occur and t_j. (We are not aware of any author actually making such an adjustment.) Discrete time models are often used as approximations for asynchronous systems. See Section 3.8.

Having stated this distinction, we will no longer observe it. We use the term 'synchronous' throughout this paper.

In a synchronous model, buffer levels $n_{i,i+1}$ and machine states have their changes observed at times $t_j = j\Delta$. Δ is referred to as the **time unit** and, without loss of generality, it is usually assumed that $\Delta = 1$. The equations for the dynamics of the changes are influenced by whether buffer levels are assumed to change before or after the machine states during the interval $[t_{j-1}, t_j]$. In Buzacott's synchronous models (1967a, b), changes in buffer levels are assumed to take place before machine state changes occur; in Gershwin's (1987a), machine state changes occur first. Which comes first in a time step is only a matter of convenience and convention, and does not affect the behavior of the model. A buffer level does not change if both adjacent machines are inoperable due to being down, blocked, or starved. It also does not change if neither adjacent machines is inoperable due to being down, blocked, or starved. Otherwise, it increases or decreases by 1.

In the models of which we are aware, all the operations and machine state changes in the line are treated as simultaneous, and all the buffer level changes are treated as simultaneous. In models of operations and machine state changes, adjacent buffers must be considered, because if they are empty or full, the machine is starved or blocked and the operation is not allowed to occur. The blocking concepts described above for asynchronous systems — BBS and BAS — apply to synchronous systems as well.

Synchronous models may involve both reliable and unreliable machines. FLRM (reliable) models become non-trivial only with random operation times, for example with discrete phase-type distributions. FLUM (unreliable) models may have deterministic or random operation times as long as they have randomness in their up- or down-times. However, most FLUM models have deterministic operation times. As in an asynchronous model, ODFs and TDFs can be considered. However, we note that in the case of a synchronous model with deterministic operation times, there is no difference between time-ODFs and number-ODFs.

2.3 Continuous Models

The feature that distinguishes continuous models from the others is that the material is treated as continuous rather than discrete. That is, instead of discrete parts moving from buffer to

machine to buffer at specific instants, there is a fluid that is transferred continuously. Machine M_i is **starved** if one of the machines $M_j, j < i$ upstream of it is down and all buffers between them, $B_{j,j+1}, \ldots, B_{i-1,i}$, are empty. Similarly, a machine is **blocked** if one of the machines downstream from it is down and all buffers between them are full. The **speed** μ_i of Machine M_i is the maximum rate at which it can transfer material from its upstream buffer to its downstream buffer, when both machines are up and neither buffer is empty or full.

In a continuous model where machines have different speeds, a machine may be slowed down, that is, forced to work at a rate slower than its speed. A machine M_i is slowed down by the upstream part of the line (**upstream limited**) if one of the machines $M_j, j < i$ upstream of it works at a slower speed $\mu_j < \mu_i$ and all buffers between are empty. Similarly, a machine is slowed down by the downstream part of the line (**downstream limited**) if one of the machines downstream of it works at a slower speed and all buffers between are full.

Buffer levels change in a continuous way. If Machine M_i and Machine M_{i+1} are both working at their own speed, then the level of Buffer $B_{i,i+1}$ changes by $(\mu_i - \mu_{i+1})\delta t$ during a time interval of length δt. If either machine is down, then the appropriate term in this expression is deleted; if both are down, then the buffer level does not change. If either machine is slowed down, then the adjusted speed is used in this expression.

In a continuous model, some of the issues discussed for the asynchronous model are not meaningful, for instance, concepts like BAS and BBS. The continuous model has mainly been used when machines are unreliable. Again, the uptimes and downtimes are usually assumed to be i.i.d. random variables with continuous distributions, and both ODFs and TDFs can be considered. In some continuous models with ODFs, the failure rate of a machine is reduced when this machine is slowed down. (See Section 3.8.1.)

The continuous model is naturally suited to production systems in which the material that is to be processed is a fluid rather than discrete entities like parts — such as chemical processing. In this paper, however, we mainly devote our attention to production systems with discrete parts. The continuous model is nevertheless of interest since it has often been used as an approximation of discrete models, especially the asynchronous model. (See Section 3.8.)

2.4 Typical Assumptions of All Models

Most of the results that are discussed in Sections 3 to 6.7 are based on the following assumptions. The first machine is never starved and the last machine is never blocked. All the random variables (processing times, uptimes, downtimes) are independent random variables. The transfer time through the buffers takes zero time. The failures are single-machine failures and most often time-ODFs. When a failure occurs, the part stays on the machine; it can be reworked when the machine is up again (i.e., there is no scrapping of parts); the work resumes exactly at the point it stops. Times to failure and times to repair of machines are usually assumed to be exponentially distributed when time is continuous, and geometrically distributed when time is discrete.

3 General Properties

The purpose of this section is to survey several results pertaining to flow line models that are of general interest. We first introduce and discuss the major performance measures of flow lines.

3.1 Performance Measures of Flow Lines

We first define some basic quantities of the flow line models.

T_i: average processing time of Machine M_i.

μ_i: average processing speed of Machine M_i.

$MTTF_i$: average time to failure of Machine M_i.

p_i: average failure rate of Machine M_i.

$MTTR_i$: average time to repair of Machine M_i.

r_i: average repair rate of Machine M_i.

e_i: efficiency of Machine M_i in isolation.

ρ_i: production rate of Machine M_i in isolation.

The **isolated efficiency** e_i is the average fraction of the time that Machine M_i would be operational if it were operated in isolation, that is, never starved or blocked. This quantity is also referred to as the **availability** of Machine M_i. Note that $e_i = 1$ for a reliable machine and $e_i < 1$ for an unreliable machine. For synchronous systems, all μ_i are the same, and the time unit is usually chosen so that $\mu_i = 1$.

The following relate these quantities:

$$\mu_i = \frac{1}{T_i} \tag{2}$$

$$p_i = \frac{1}{MTTF_i} \tag{3}$$

$$r_i = \frac{1}{MTTR_i} \tag{4}$$

$$e_i = \frac{MTTF_i}{MTTF_i + MTTR_i} = \frac{r_i}{p_i + r_i} \tag{5}$$

$$\rho_i = \mu_i e_i = \frac{\mu_i r_i}{p_i + r_i} \tag{6}$$

All the results that we review in this paper are concerned with steady-state (average long term) behavior of the production line. Therefore, all the quantities we calculate are steady-state performance parameters. Several measures of performance are of interest when analyzing flow line models. The most important is the **production rate**, P, the average number of parts that leave the system per unit of time. Also of importance is the inventory level of the system. Define $\bar{n}_{i,i+1}$ to be the **average number of parts** (or **average buffer level** in Buffer $B_{i,i+1}$. The **average work-in-process** (or **WIP**) in the system, $\bar{n}$, is given by

$$\bar{n} = \bar{n}_{1,2} + ... + \bar{n}_{K-1,K}. \tag{7}$$

Using Little's law (Little, 1961), the average flow time of a part, W, can then be obtained as: $W = \bar{n}/P$.

Other performance parameters of interest are:

E_i: probability of Machine M_i being working.

P_i: production rate of Machine M_i.

S_i: probability of Machine M_i being starved.

B_i: probability of Machine M_i being blocked.

I_i: probability of Machine M_i being idle.

D_i: probability of Machine M_i being down.

In FLRMs, the quantity E_i corresponds to the proportion of time Machine M_i is not idle (neither starved nor blocked) and is referred to as the **utilization rate** of Machine M_i. In FLUMs, it corresponds to the proportion of time Machine M_i is neither idle (starved or blocked) nor down and is referred to as the **efficiency** of Machine M_i. In the following, we use efficiency to refer to E_i for an unreliable machine as well as for a reliable machine. Finally, we note that the production rate of a flow line as defined above is given by $P = P_K$.

3.2 Some Basic Relationships

Several relationships hold under very general conditions, especially for general distributions of processing times, uptimes and downtimes. We first discuss relationships pertaining to a single machine. The first relates the production rate of a machine to its efficiency and average processing rate:

$$P_i = \mu_i E_i \tag{8}$$

This relationship holds for FLRMs and also for FLUMs provided that the work resumes exactly at the point it stopped in case of failure. (See Section 2.1.3.) In synchronous systems where all μ_i are 1, $P_i = E_i$. Consequently, E_i is often called production rate in synchronous systems.

Consider FLUMs with ODFs. Because idle times do not influence the failure/repair behavior of the machine, we have:

$$p_i E_i = r_i D_i \tag{9}$$

A proof of this result is given in the Appendix. Using equations (5) and (9) and the fact that the probabilities sum up to 1, i.e., $E_i + D_i + I_i = 1$, we obtain:

$$E_i = e_i(1 - I_i) \tag{10}$$

This relationship is referred to as the **flow rate-idle time** relationship. It holds for any machine with ODFs.

In the case of TDFs, equations (9) and (10) are replaced by

$$p_i(E_i + I_i) = r_i D_i \tag{11}$$

$$E_i = e_i - I_i \tag{12}$$

Equation (11) is also proved in the Appendix.

A fundamental relationship of flow lines is the **conservation of flow**. It states that all the machines have the same average production rate, that is,

$$P_1 = P_2 = ... = P_K = P \tag{13}$$

Conservation of flow holds for FLRMs. It also holds for FLUMs provided that there is no scrapping of parts. (If there is scrapping, it can be adjusted accordingly.)

Actually, this relationship, as well as some of the results presented below can be established using a sample path approach. It is described in Section 3.3.

3.3 Evolution Equations

The **sample path** behavior of any flow line can be described by means of recursive equations. These equations will be referred to as the **evolution equations** of the flow line. They have proved to be very useful in flow line analysis, mainly for establishing qualitative properties such as monotonicity and reversibility. These equations have been used in many papers under various forms, e.g., Hildebrand (1967, 1968), Yamazaki and Sakasegawa (1975), Muth (1979), Shanthikumar and Yao (1989), Dallery, Liu, and Towsley, (1990).

Consider a FLRM with the following initial condition. At time $t = 0$, all buffers are empty and the first machine initiates the processing of a new part. Let $\sigma_{i,n}$ denote the n'th processing time of machine M_i, and let $D_{i,n}$ denote the n'th departure time of a part from Machine M_i. In the case of BBS-PO, we have the following evolution equations:

$$D_{i,n} = \max\left(D_{i,n-1}, D_{i-1,n}, D_{i+1,n-N_{i,i+1}}\right) + \sigma_{i,n} \ , \quad \forall i, n \geq 1 \tag{14}$$

where, by convention, $D_{i,n} = 0 \ \forall i$ if $n \leq 0$.

Equation (14) results from the following observations. Since we consider BBS, the departure time from Machine M_i occurs exactly $\sigma_{i,n}$ units of time after the start of the process. The start time of the n'th process is expressed by the first term of equation (14). There are three conditions that must be satisfied before the n'th process of Machine M_i can begin: 1) the machine must be available; 2) the upstream Buffer, $B_{i-1,i}$, must be non-empty; and 3) the downstream Buffer, $B_{i,i+1}$, must be non-full. The first condition is satisfied when Machine M_i has completed its $(n-1)$'th process, which occurs at time $D_{i,n-1}$. The second condition is satisfied when Machine M_{i-1} has completed its n'th process, which occurs at time $D_{i-1,n}$. The third condition is satisfied when Machine M_{i+1} has completed its $(n - N_{i,i+1})$'th process, which

occurs at time $D_{i+1,n-N_{i,i+1}}$. Since these three conditions must all be satisfied, the time that processing begins is thus the maximum of these three times.

In the case of BAS, the evolution equations are modified as follows:

$$D_{i,n} = \max\left(\max\left(D_{i,n-1}, D_{i-1,n}\right) + \sigma_{i,n},\ D_{i+1,n-N_{i,i+1}}\right),\ \ \forall i, n \geq 1 \tag{15}$$

The difference is that in the case of BAS, the process can start even though the downstream buffer is full, but the transfer can take place only when a space is available. Finally, we note that these evolution equations can appropriately be modified in the case of a non-empty initial condition. (See e.g., Dallery, Liu, and Towsley, (1990).) These equations do not require any assumption on the sequences of processing times $\sigma_{i,n}$, $n \geq 1$. Similar evolution equations can be derived in the case of BBS-PNO (Dallery, Liu, Towsley, 1991).

In both cases (BBS and BAS), the production rate of Machine M_i can then be expressed as:

$$P_i = \lim_{n\to\infty} \frac{n}{E[D_{i,n}]} \tag{16}$$

General conditions under which this limit exists are given in (Dallery, Liu, and Towsley, 1990). In particular, it exists if the processing times $\sigma_{i,n}$, $n \geq 1$, are i.i.d. random variables.

It follows from equation (14) in the case of BBS-PO or equation (15) in the case of BAS that:

$$D_{i+1,n} \geq D_{i,n} \geq D_{i+1,n-N_{i,i+1}} \tag{17}$$

which implies:

$$\frac{E[D_{i+1,n}]}{n} \geq \frac{E[D_{i,n}]}{n} \geq \frac{E[D_{i+1,n-N_{i,i+1}}]}{n - N_{i,i+1}}\ \frac{n - N_{i,i+1}}{n} \tag{18}$$

This equation implies $P_i = P_{i+1}$ when n goes to infinity (provided that the limits exist), thus proving conservation of flow (equation (13)).

These evolution equations have sometimes been used in a simpler form corresponding to the case of flow lines with no intermediate storage. In the case of BAS, equation (15) then reduces to:

$$D_{i,n} = \max\left(D_{i-1,n} + \sigma_{i,n},\ D_{i+1,n-1}\right),\ \ \forall i, n \geq 1 \tag{19}$$

Note that one term has been dropped. The reason is that in the case of BAS and no intermediate storage, we have $D_{i-1,n} \geq D_{i,n-1}$.

Using evolution equations corresponding to the case of flow lines with no intermediate storage is however not restrictive since any flow line can be transformed into a flow line with no intermediate storage by adding fictitious machines with zero processing times that represent the transfer of parts from a buffer space to the next one. This trick is due to Avi-Itzhak (1965). Evolution equations of this simpler form were used, among others, by Hildebrand (1967, 1968), Yamazaki and Sakasegawa (1975), and Muth (1979).

Remark. Some of the properties presented in Sections 3.4, 3.5, and 3.6, have been obtained using a sample path approach based on the above evolution equations. These properties hold under fairly general assumptions pertaining to the processing times (see e.g. Dallery, Liu,

and Towsley, 1990). These assumptions include independent and identically distributed (i.i.d.) sequences of processing times (i.e., GI distributions) as a special case. Although these evolution equations have mainly been used in the context of FLRMs, they can be adapted to FLUMs with ODFs by interpreting $\sigma_{i,n}$ as the completion time instead of just the processing time of Machine M_i. (See Section 3.7.) Therefore, we conjecture that all properties that have been obtained for FLRMs based on these evolution equations are also valid for FLUMs with ODFs. See Dallery, Liu and Towsley (1990) for a discussion pertaining to this issue.

3.4 Blocking Issues

The purpose of this section is to discuss the relationships between the different types of blocking. We first compare BAS (blocking-after-service) and BBS-PNO (blocking before service with position at the machine non-occupied while the machine is blocked). It is convenient to define the vector $\mathbf{1} = (1, \ldots, 1)$.

Consider two flow lines that differ only from one another by the buffer capacities and the type of blocking. One line has buffer capacity vector $\boldsymbol{N}$ and is operated according to the BAS blocking mechanism. The other has buffer capacity vector $\boldsymbol{N} + \mathbf{1}$ and is operated according to the BBS-PNO blocking mechanism. Then these two flow lines have exactly the same behavior. In particular, they have the same production rate. In other words, BAS with buffer capacity $\boldsymbol{N}$ is equivalent to BBS-PNO with buffer capacity $\boldsymbol{N} + \mathbf{1}$. This result was first established by Onvural and Perros (1986) in the case of exponential processing time distributions. This equivalence actually holds for general distributions as shown by Dallery, Liu and Towsley (1991). The proof is based on comparing the evolution equations of the two lines. The intuitive idea of this equivalence is that, for each machine, a single buffer space (namely the first space of the upstream buffer) for the line operating with BAS plays exactly the same role as two buffer spaces (namely the first space of the upstream buffer and the last space of the downstream buffer) for the line operating with BBS-PNO.

Next, we compare BAS and BBS-PO. Consider first the case of two-machine flow lines ($K = 2$). In this case, there is again an exact equivalence between BBS-PO with buffer capacity $N_{1,2}$ and BAS with buffer capacity $N_{1,2} - 1$, as pointed out by Altiok and Stidham (1982), among others. This can be viewed as a simple consequence of the above equivalence between BBS-PNO and BAS since, in the case of two-machine flow lines, there is no difference at all between BBS-PO and BBS-PNO. Unfortunately, in this case there is no equivalence for flow lines consisting of more than two machines. However, it is possible to obtain bounds on the production rate of one model from the production rate of the other model. The following results are proved in Dallery, Liu, and Towsley (1991):

$$P(BBS - PO, \boldsymbol{N}) \leq P(BAS, \boldsymbol{N}) \leq P(BBS - PO, \boldsymbol{N} + \mathbf{1}) \tag{20}$$

$$P(BAS, \boldsymbol{N} - \mathbf{1}) \leq P(BBS - PO, \boldsymbol{N}) \leq P(BAS, \boldsymbol{N}) \tag{21}$$

This result has the following consequence. Consider, for instance, equation (20). For flow lines with large buffer sizes, the difference in the production rates with buffer capacity $\boldsymbol{N}$, $P(BBS - PO, \boldsymbol{N})$, and with buffer capacity $\boldsymbol{N} + \mathbf{1}$, $P(BBS - PO, \boldsymbol{N} + \mathbf{1})$, is very small. As a

result, the production rates with BAS, $P(BAS, \boldsymbol{N})$, and BBS-PO, $P(BBS-PO, \boldsymbol{N})$ are very close to one another. As a consequence, the distinction between BAS and BBS-PO becomes less important for flow lines with large buffers. Large buffers are often encountered in FLUMs or in FLRMs with processing times having large variances.

Remark. Because of the equivalence of BBS-PNO with BAS, we only consider BAS and BBS-PO in the rest of the paper, and we refer to the latter as BBS, or blocking-before-service.

3.5 Buffer Issues

The effects of intermediate storage between machines in a flow line is of great interest. Some qualitative observations are discussed in Section 3.9. In this section, we provide some theoretical results pertaining to these observations.

The most important property is **monotonicity** of the production rate of a flow line with respect to the buffer capacities. Consider two flow lines, L^1 and L^2, which have identical machines but with different buffer capacity vectors $\boldsymbol{N}^1$ and $\boldsymbol{N}^2$. The capacity of each buffer in L^2 is at least as large as the corresponding buffer in L^1. That is, $\boldsymbol{N}^1 \leq \boldsymbol{N}^2$. Then the production rate of the flow line satisfies:

$$P(\boldsymbol{N}^1) \leq P(\boldsymbol{N}^2) \tag{22}$$

In other words, the production rate in an increasing function of the buffer capacities. This result was proved by Shanthikumar and Yao (1989) using the evolution equations presented in Section 3.3. Although it was established in the context of closed systems, it is readily applicable to the case of flow lines.

An interesting question is: what happens when one or more buffers increase without limit? Consider a flow line L with buffer capacity vector $\boldsymbol{N}$. Suppose first that the capacity of one buffer, say $B_{i,i+1}$, is increased while the capacity of all other buffers remains constant. From the monotonicity property, we know that the production rate of line L will increase and since it is bounded (in particular by the production rate in isolation of the last machine, ρ_K), it will asymptotically reach a limit. Let P^* denote this limit. This quantity is the production rate of a flow line having one infinite buffer.

The production rate of a flow line in which one buffer is infinite can be obtained by decomposing the line into two sublines. Let L^a be the part of line L that consists only of the first i machines and the first $i-1$ buffers. Similarly, let L^b be the part of line L that consists only of the last $(K-i)$ machines and the last $(K-i-1)$ buffers. Let P^a and P^b be the production rates of lines L^a and L^b, respectively. Then, we have

$$P^* = \min\left(P^a, P^b\right) \tag{23}$$

This result was proved by Baccelli (1990) in the context of timed marked graphs, a special case of a Petri net. Since a flow line can be seen as a special case of a timed marked graph, this result applies to our case. By combining this result with the monotonicity property, we obtain the following upper bound for the production rate of the original line:

$$P \leq \min\left(P^a, P^b\right) \tag{24}$$

By applying this decomposition several times, we obtain the following well known result: the production rate of a flow line is bounded by the isolated production rate of the machine that has the smallest isolated production rate. That is:

$$P \leq \min\left(\rho_1, \rho_2, ..., \rho_K\right) \tag{25}$$

This result was reported, among others, by Muth (1973). Note that $\min\left(\rho_1, \rho_2, ..., \rho_K\right)$ is the production rate of the flow line with infinite buffers. The following tighter upper bound on the production rate also follows from the above approach:

$$P \leq \min\left(P^{1,2}, P^{2,3}, ..., P^{K-1,K}\right) \tag{26}$$

where $P^{i,i+1}$ is the production rate of the two-machine flow line consisting of Machine M_i, Buffer $B_{i,i+1}$, and Machine M_{i+1}. This result may be useful since the production rates of two-machine flow lines can in most cases be exactly calculated. (See Section 4.)

The monotonicity property can also be used to obtain the following lower bound on the production rate:

$$P \geq P^0 \tag{27}$$

where P^0 is the production rate of the flow line with no intermediate buffer storage, i.e., $C_{i,i+1} = 0$, for all $i = 1, ...K-1$. A discussion of how this lower bound can be calculated is provided in Section 4.1.

3.6 Reversibility and Duality Properties

Consider a flow line, L^r, which is obtained from flow line L by reversing the flow of parts. The first machine of L^r is the same as the last machine of L. More generally, Machine M_i in L^r is the same as Machine M_{K-i+1} in line L. Also, Buffer $B_{i,i+1}$ is the same as Buffer $B_{K-i,K-i+1}$.

Assume that the blocking mechanisms of L and L^r are both BAS. Then, the following **reversibility** property has been established by Yamazaki and Sakasegawa (1975), Dattatreya (1978), and Muth (1979): the production rate of the reversed line L^r is the same as that of the original line L. The proof is based on the comparison of the sample paths of the two systems again using the evolution equations introduced in Section 3.3. (Note that they actually used equation (19).) With BAS, the production rate is the only performance parameter (among those defined in Section 3.1) which is preserved by the reversibility transformation. Further results on the reversibility of flow lines, especially pertaining to transient results and to the case of parallel machines, can be found in (Yamazaki and Sakasegawa, 1975; Yamazaki, Kawashima, and Sakasegawa, 1985; Melamed, 1986; Dallery, Liu, and Towsley, 1991).

Consider now the case of BBS. In that case, there is a much stronger equivalence between the two systems. This equivalence is based on the concept of job/hole (or part/hole) **duality** introduced by Gordon and Newell (1967) and also noticed earlier by Sevast'yanov (1962) and termed articles/anti-articles. The idea is that in line L, whenever a part moves in one direction, a hole (empty space) moves in the other direction. In the case of BBS, it is easy to check that the behavior of parts in the reversed system is the same as the behavior of holes in the original system. Indeed, starvation in the reversed system corresponds to blocking in the original

system, and vice-versa. As a result, the steady-state distribution of parts in the reversed line is exactly the same as the steady-state distribution of holes in the original line. This equivalence especially implies that the two systems have the same production rate and that the average buffer levels of corresponding buffers sum up to the capacity of the buffer. These results were proved by Ammar (1980) for a Buzacott-type model (Section 4.3.2), Ammar and Gershwin (1989) in the case of exponentially distributed processing times, and under general conditions (general processing time distributions) by Dallery, Liu, and Towsley (1990) and Liu (1990) using sample path arguments. Finally, we note that a similar duality property was obtained by De Koster (1988a) in the case of continuous models of flow lines.

A question that naturally arises is whether or not these arguments can also be used in the case of BAS. Unfortunately, the answer is no. The concept of job/hole duality still makes sense. However, in that case, the behavior of parts in the reversed system is no longer the same as the behavior of holes in the original system. In particular, the blocking mechanism of holes in line L is BBS whereas that of parts in line L^r is BAS. Thus, it appears that starvation and blocking-before-service are dual of each other with respect to the job/hole concept, but starvation and blocking-after-service are not.

In the special case of two-machine lines with BAS, it is possible to obtain an equivalence between the original and the reverse lines, which is similar to the duality property. This is simply done by using the equivalence between BAS and BBS discussed in Section 3.4. Indeed, $L(BAS, N)$ is equivalent to $L(BBS, N+1)$ and $L^r(BAS, N)$ is equivalent to $L^r(BBS, N+1)$. Consequently, the duality property that relates lines $L(BBS, N+1)$ and $L^r(BBS, N+1)$ can be reinterpreted in terms of the performance measures of $L(BAS, N)$ and $L^r(BAS, N)$.

Besides being of theoretical interest, duality and reversibility properties have some practical value. They may be used as criteria for testing the validity of approximation methods of long flow lines. (See Section 5.) Indeed, one may check whether a given approximate method is consistent with these equivalence properties. In the case of BAS, an approximation is consistent with these properties if it provides the same estimate of the production rate for any given flow line and its reversed line.

3.7 Reliable Versus Unreliable Machines

In this section, we discuss some equivalences between reliable and unreliable machines. We show that in some cases, an unreliable machine can exactly or approximately be modeled by a reliable machine, and vice-versa. Before presenting these results, we discuss the usefulness of such equivalences. Suppose one has a flow line where some machines are reliable while others are unreliable. Most analytical techniques that are presented in the next sections, especially approximate techniques, are devoted either to FLRMs or to FLUMs. Consequently, in order to be able to use these techniques, one must be able to transform the original flow line model having both types of machines into a model with either all the machines reliable (FLRMs) or all the machines unreliable (FLUMs).

Moreover, one may transform a FLRM into a FLUM in order to be able to use an approximate technique devoted to FLUMs. This may be needed if no approximate technique is available to handle the FLRM under study, or if the parameters of the FLRM are such that one expects that the approximate technique for FLUMs will provide more accurate results. A similar statement

can be made regarding the transformation of FLUMs into FLRMs.

Remark. Equivalences between a reliable and an unreliable machine only exists in the case where the unreliable machine has ODFs. Consequently, in this section, we restrict our attention to ODFs.

3.7.1 Modeling an Unreliable Machine by a Reliable Machine

The transformation of an unreliable machine into a reliable machine is based on the notion of **completion time** introduced by Gaver (1962). The completion time of an unreliable machine is defined as the time between the instants of beginning and completion of the processing of a part. This time includes the time corresponding to the actual processing of the part, plus the repair times corresponding to all the failures that have occurred during the processing of this part. Note that in the case of number-ODF, the number of failures is either 0 or 1, while in the case of time-ODF, it can be any non-negative integer.

Consider first time-ODFs. If uptimes are exponentially distributed, successive completion times are independent. Thus, in that case, the completion time distribution is a GI (general independent) distribution. The idea is then to replace the unreliable machine by a reliable machine that has processing time distribution identical (or approximately identical) to the completion time distribution of the unreliable machine. The question that arises is how to characterize the processing time distribution of the reliable machine.

Altiok and Stidham (1983) consider the simplest case where processing time, uptime, and downtime distributions of the unreliable machines are exponential. They calculate the Laplace transform of the completion time and show that it can be exactly identified as a Coxian-2 distribution. For more general cases, it is still possible to obtain the Laplace transform of the completion time (Nicola, 1986). Unfortunately, no results for obtaining a Coxian or a phase-type distribution that exactly fits this distribution are available. However, it is possible to determine a Coxian or a phase-type distribution that has the same first two (or three) moments as the completion time distribution. (See Appendix.)

Another approach can be used when both the processing times and the repair times are characterized by Coxian or, more generally, phase-type distributions. In that case, the completion time distribution can be represented exactly by a phase-type distribution which is obtained by construction; see, among others, Altiok (1985b), Bobbio and Trivedi (1988) Gun (1987), Gun and Makowski (1990). To illustrate this, consider again the case where all distributions are exponential. Let μ, p, and r, denote the rates of the processing time, uptime, and downtime distributions, respectively. Then the phase-type form of the completion time distribution is shown in Figure 2. Phases 1 and 2 correspond to the machine being up and down. In phase 1, the machine is working and two events may happen: either the end of the processing, with rate μ; or a failure, with rate p. Thus, the time spent in phase 1 is exponentially distributed with rate $\mu + p$.

Consider now number-ODFs. If uptimes are geometrically distributed, successive completion times are independent and therefore the completion time distribution is again a GI distribution. In that case again, the completion time distribution can be represented exactly by a phase-type distribution provided that processing times and repair times are characterized by phase-type distributions. The phase-type distribution is again simply obtained by construction (Buzacott

Figure 2: Completion Time Distribution

and Kostelski, 1987; Gun,1987).

Finally, we note that by using this approach, both types of ODFs can be combined. Also, this approach can be used to incorporate inspection and rework, and moreover failure and inspection features can be combined (Sastry and Awate, 1988). Thus, this approach is very attractive since it makes it possible to incorporate many features into an equivalent completion time distribution. We note, however, that the number of phases of the resulting distribution may be large, especially if the original processing and repair phase-type distributions have several phases. This is the drawback of this approach compared to the approach based on identification of moments.

3.7.2 Modeling a Reliable Machine by an Unreliable Machine

We now discuss the transformation of a reliable machine into an unreliable machine. Consider first the case where the reliable machine has a Coxian-2 processing time distribution with parameters (μ_1, a_1, μ_2). It can be modeled exactly by an unreliable machine with number-ODFs with processing speed μ_1, repair rate μ_2, and where a_1 is the probability of having a failure at the completion of an operation. See Buzacott (1972). Alternatively, it can also exactly be modeled by an unreliable machine with time-ODFs whose parameters are obtained by reversing the transformation of Altiok and Stidham (1983). It is easy to check that this can be done only if the original distribution of processing times has a coefficient of variation greater than 1.

In more general cases, there is no simple way of doing an exact transformation. However, it is again possible to determine the parameters of an unreliable machine whose associated completion time distribution is close to the original processing time distribution. In that case, one must first choose the characterization of the unreliable machine, i.e., the distributions of processing times, uptimes, and downtimes, as well as the type of ODFs. Then, the parameters

of these distributions are determined in such a way that the resulting distribution of completion time has the same first two (or three) moments as the distribution of processing times of the reliable machine. Such an approximate transformation was used by Liu and Buzacott (1989).

3.8 Relationships Among Models

In Section 2, we introduced three major classes of models: asynchronous, synchronous, and continuous. Each model is of interest as a representation of a physical system. For instance, a production line may operate in such a way that parts can only be transferred every T units of time, in which case the synchronous model is probably the model of choice.

However, each of these models can also be of interest as an approximation of another model. Most importantly, the synchronous and the continuous models can be used as approximations of the asynchronous model. Since several authors have followed this approach, it is worthwhile describing how it works.

3.8.1 Approximation of the Asynchronous Model

Consider an asynchronous FLUM model with the following features: the processing times are deterministic; the uptimes and downtimes are exponentially distributed; failures are operation dependent (ODFs). Consider first the case where all machines have the same processing time, T. No exact solution of this asynchronous model has yet been obtained, even for the simple case of two-machine lines, except in the case of no intermediate storage (Commault and Dallery, 1990). As a result, several authors have proposed to use either the synchronous model or the continuous model as an approximation to the behavior of the asynchronous model.

Approximation by continuous model; identical machine speeds. The idea is to approximate the discrete flow of parts by a continuous flow of material. Each machine of the continuous model has the same distributions for uptimes and downtimes as the corresponding machine of the asynchronous model. All the machines of the continuous model have the same speed, μ, which is the inverse of the processing time: $\mu = 1/T$.

The question is whether or not the continuous model is a good approximation. It was shown experimentally (e.g., Alvarez, Dallery, and David, 1991) that it is actually a good approximation provided that the following assumption is satisfied.

Assumption MTS (Multiple Time Scale). The uptimes and downtimes of the machines of the flow line are much larger (at least one order of magnitude) than the processing times.

A theoretical justification was provided by David, Xie, and Dallery (1990). They showed that, in the case of BAS, the production rate of the asynchronous model with buffer capacity $\boldsymbol{N}$ is bounded by the production rates of the continuous model with appropriate buffer capacities, namely:

$$P(Cont, \boldsymbol{N} - \mathbf{1}) \leq P(Asynch, \boldsymbol{N}) \leq P(Cont, \boldsymbol{N} + \mathbf{1}) \tag{28}$$

FLUMs in which the uptimes and downtimes are large compared to the processing times are likely to have large buffers. In this case, the difference between the first and third terms in

equation (28) is small. This further implies that the production rate of the continuous model is a very good approximation of that of the asynchronous model.

Approximation by synchronous model; identical machine speeds. The asynchronous model may alternatively be approximated by a synchronous model. Each machine of the synchronous model has constant processing T, and has geometric uptimes and downtimes having the same means as the exponential uptimes and downtimes of the asynchronous model. In other words, the synchronous model forces events to occur only a times multiple of T, whereas in the original asynchronous model, events could occur at any time. Again, this model is a good approximation of the asynchronous model under Assumption MTS.

Approximation by continuous model; different machine speeds. Consider now the case where the machines of the asynchronous model have different processing times, T_i. The asynchronous model can no longer be approximated by a synchronous model, since the major feature of the synchronous model is that all machines have the same processing times. On the other hand, it can still be approximated by a continuous model. The speed of machine M_i in the continuous model is $\mu_i = 1/T_i$. The characterization of uptimes requires special attention in the case of ODFs. Let p_i and r_i be the failure and repair rates of Machine M_i in the asynchronous model. If Machine M_i of the continuous model is working at its own speed, μ_i, its failure rate is simply p_i. However, Machine M_i may also be slowed down, as discussed in Section 2.3. Suppose for a period of time, Machine M_i is slowed down by Machine M_j, where $\mu_j < \mu_i$. During this time, machine M_i has a reduced failure rate equal to $p_i(\mu_j/\mu_i)$. The reason is that the behavior of the continuous model must be as close as possible to that of the asynchronous model.

For the sake of simplicity, consider the case where $j = i - 1$. Machine M_i slowed down by Machine M_{i-1} in the continuous model corresponds to the case where buffer $B_{i-1,i}$ is empty in the asynchronous model. In that case, the behavior of the asynchronous model is as follows. Both machines start working on a part at the same instant. Machine M_i completes its operation T_i time units after while Machine M_{i-1} completes its operation T_{i-1} time units after. As a result, Machine M_i is idle (starved) for a length of time equal to $T_{i-1} - T_i$. During the first T_i time units, Machine M_i may fail at rate p_i, whereas during the remaining $T_{i-1} - T_i$ time units, it cannot fail since it is starved and we assume ODFs. Therefore, its average failure rate in this situation is $p_i(T_i/T_{i-1}) = p_i(\mu_{i-1}/\mu_i)$. The continuous model must reflect this and, as a result, Machine M_i has a reduced failure rate. On the other hand the repair times are exponentially distributed with rate r_i. Note that in the case of TDFs, Machine M_i of the continuous model has always a failure rate of p_i since in the asynchronous model Machine M_i may still fail when idle.

As for the case of identical machine speeds, the continuous model is a good approximation of the asynchronous model provided that Assumption MTS is satisfied (Alvarez, Dallery, and David, 1991). Also, the bounding properties expressed in equation (28) still hold (David, Xie, and Dallery, 1990). The fact that the continuous model can approximate asynchronous models with non-identical processing times is one of its major advantages over the synchronous model.

In many FLUMs, failures occur much less often than processing of parts and therefore, Assumption MTS is likely to be often satisfied. Thus, continuous or synchronous models used as

approximation of asynchronous models will often provide very good estimates of the performance parameters.

Most of the work pertaining to the approximation of asynchronous models of FLUMs with deterministic processing times assumes exponential and geometric up- and downtimes in the case of continuous and synchronous models, respectively. However, such an approximation can also be used in the case of general distributions of up and down-times. Again, this approximation will be accurate provided that Assumption MTS is satisfied (David, Xie, and Dallery, 1990).

Other approximations. The major usefulness of the continuous (or synchronous) model as an approximation of the asynchronous model is in the case of FLUMs having deterministic processing times. However, it can also be used in other cases, especially in the case of FLUMs with random processing times satisfying Assumption MTS. Finally, continuous and synchronous models have also been used as approximations of FLRMs (De Koster and Winjgaard; 1989, Liu and Buzacott, 1989). Such approximations should especially work well in the case where the processing times have large variances.

3.8.2 Approximation of the Synchronous Model

The continuous model can also be used as an approximation of the synchronous model of FLUMs. Gershwin and Schick (1980) investigated the relationships between two-machine synchronous FLUM systems and two-machine continuous material FLUM systems by means of the **δ-transformation**. In this transformation, the failure and repair rates of both machines are multiplied by a small number δ and the buffer size is divided by δ. That is, (r_1, p_1, r_2, p_2, N) are the parameters of one synchronous two-machine system, and $(r_1', p_1', r_2', p_2', N')$ are the parameters of the other, with:

$$r_i' = r_i\delta \tag{29}$$

$$p_i' = p_i\delta \tag{30}$$

$$N' = \frac{N}{\delta} \tag{31}$$

The only restriction on δ is that the new buffer size N' must be an integer.

When N' is large, the performance measures of the systems are approximately related. Production rate P' is close to P, and average buffer level $\bar{n}'$ is approximately $\bar{n}/\delta$. In the limit as $\delta \to 0$, the new synchronous system approaches a continuous material flow system whose production rate is also approximately that of the original system. This approximation provides especially good results for systems that satisfy Assumption MTS.

The reason that this is true is that the δ-transformation is essentially a transformation of the cycle time. The new cycle time is δ times the old cycle time, and the amount of material processed during a new cycle time is δ times the amount of material processed during an old cycle time. Consequently, the probability of a failure or a repair during a new cycle is δ times the probability during an old cycle. The size of the buffer has not changed, but if it is measured in units of the amount material processed during a cycle time, it must be divided by δ.

We would suggest that this transformation can be extended to relate all models: synchronous, asynchronous, and continuous, and a variety of failure and repair time distributions, blocking models, etc. It can be useful in reducing numerical effort in analyzing systems.

Finally, we note that numerical results on the approximation of the synchronous model of FLUMs with TDFs by a continuous model are reported by De Koster and Wijngaard (1989). Again, it appears that this approximation is accurate provided that Assumption MTS is satisfied.

3.8.3 Other Relationships Among Models

In this section, we discuss results pertaining to the transformation of an asynchronous FLUM with non-identical deterministic processing times into an asynchronous FLUM with identical deterministic processing times. Such a transformation, although only an approximation, is of interest for several reasons. First, it appears from the above discussion that the synchronous model can be used as an approximation of asynchronous lines with deterministic processing times only in the case where all the processing times are equal. Therefore, if one wants to use synchronous models, it is necessary to be able to transform an asynchronous line with non-identical processing times into an asynchronous line with identical processing times. Secondly, even though continuous models can handle non-identical processing times, such a transformation may still be of interest when using such models. The reason is that some solution techniques may be restricted to (or at least simpler) the case of identical processing times. (See Section 5.2.1.)

Several transformations have been proposed. Gershwin (1987b) suggested replacing each machine of the original line, except the fastest, by a set of two machines with no intermediate buffer. One machine captures the unreliability behavior of the original machine, while the other represents the effective processing time. All the machines of the resulting line have the same processing time which is equal to the processing time of the fastest machine of the original line. This transformation was referred to as **disaggregation** by Dallery, David, and Xie (1989). It is very accurate as long as the failure and repair rates of the machine representing the processing time are much larger than those of the machine capturing the unreliability behavior.

A second transformation was proposed by Dallery, David, and Xie (1989) and referred to as **homogenization**. It consists of replacing each machine of the original line except the fastest by a single equivalent machine. As in the case of disaggregation, all the equivalent machines have the same processing time equal to the processing time of the fastest machine of the original line. The parameters of each equivalent machine are obtained as follows. Its repair rate is the same as that of the original machine. Its failure rate is obtained by prescribing that the equivalent machine has the same production rate in isolation as that of the original machine. Homogenization is a simpler transformation than disaggregation but is not as accurate. However, it provides a good approximation for **nearly homogeneous** lines, i.e., lines where the processing times of all machines are close to one another.

A third transformation was proposed by Liu and Buzacott (1990). It is similar to homogenization except that the failure and repair rates of the equivalent machine are determined by prescribing that the first and second moments of the completion time of the equivalent machine in isolation is the same as those of the original machine. Note that the first moment is nothing but the production rate in isolation. The accuracy of that transformation is similar to that of

Figure 3: Production rate as a function of buffer size for a two-machine synchronous line in which $r_1 = 0.08, p_1 = 0.01, r_2 = 0.09, p_2 = 0.01$.

homogenization.

3.9 Qualitative Behavior

In all the models described in the following sections, several important features were observed. In some cases, they were demonstrated analytically for a limited class of systems. In others, they were observed by numerical experimentation or by simulations. They include

Saturation and Limiting behavior Production rate is a saturating function of buffer size; that is, it is an increasing function with a finite upper bound that it approaches asymptotically as buffer size increases without limit. (See Section 3.5.) Saturation is illustrated in Figure 3. This upper bound is the production rate of the slowest machine, in the sense of isolated production rate ρ_i. If any one machine is significantly slower than the others, in that sense, then it limits the behavior of the system. The system production rate is less than that of that machine; buffers upstream of that machine tend to be full, and buffer downstream tend to be empty. In Figure 3, the limiting production rate is 0.8889, which is the isolated production rate of Machine M_1, the slower of the two machines.

Effectiveness of buffers – buffers and disruptions Buffers increase the production rate of systems *only* by decoupling the effects of disruptions from one part of the system to other parts. If a failure happens upstream of a buffer, and the buffer has enough material in it, the downstream part of the system can keep operating. Similarly, if a failure happens downstream of a buffer, and the buffer has enough space, the upstream part of the system can keep operating. Therefore, for a buffer to be effective, its size must be the same order of magnitude as the

disruptions that it must block, when measured in comparable units. If the mean down time of Machine M_i is $1/r_i$ time units, then the size of buffers $B_{i-1,i}$ and $B_{i,i+1}$ should be on the order of the amount of material that the machines typically produce in $1/r_i$ time units. On the other hand, if the system is reliable (FLRM) and operation times are random, then the buffer size should be roughly the standard deviation of an operation time. If the buffer is much smaller than this estimate, it will have very little effect on performance; if it is much larger, it is in the flat part of the saturation curve, and much of its capacity is wasted. In Figure 3, the maximum curvature of the curve occurs around $N = 40$, which is on the order of $1/r_i = 12.5$. (A more accurate value of the optimal N depends on the performance measure, as well as all four machine reliability parameters.)

Effectiveness of buffers – system balance and imbalance Because buffers increase the production rate of systems *only* by decoupling the effects of disruptions, they do not compensate for systems that are highly unbalanced. They should *not* be placed near the bottleneck (neither the machine with the smallest ρ nor the machine with the smallest μ); rather they should be placed where disruptions are greatest. A large buffer upstream of a bottleneck will be empty most of the time; a large buffer downstream of a bottleneck will be full most of the time. In both cases, it is wasted (along with the in-process inventory it accumulates, when the upstream part of the line is faster than the downstream).

Effect of increasing line length on performance Hillier and Boling (1966), using the method of Hillier and Boling (1967), observed that as a transfer line with identical machines and buffers increases in length, its production rate seems to decrease to a non-zero limit. De Kok (1988) developed an approximate technique for evaluating long transfer lines consisting of identical machines and buffers. His numerical experiments suggested the same conclusion. Ancelin and Semery (1987), David, Semery, Ancelin, and Terracol (1987), Gershwin (1989), and Semery (1988), made similar observations (and drew similar graphs) for different models using different methods. (Based on the decompositions reported in Section 5.2.1, Gershwin (1991b) developed an algorithm for predicting the limiting production rate.)

Bowl phenomenon Sevast'yanov (1962) developed a technique for analyzing long transfer lines. (See Section 5.2.1.) Based on this approximate method, he studied the optimal design of a line that has very many machines, where the designer can group the machines into a given number of groupings, and where a given amount of storage space can be divided among these groupings. He showed that the storage space should be divided equally, and the machines should be grouped together in such a way that the first and last machine groups are equally reliable; and that all the rest of the groups are equally reliable; and that the inner groups should be somewhat more reliable than the outer groups.

Hillier and Boling (1966, 1972, 1977) also observed that the optimal distribution of system resources is not an equal allocation. That is, if there is a pool of machine speed, and the goal is to maximize the system's production rate, the pool should not be divided equally among the machines. Instead, the machines in the middle of the line should get more, and the machines at the ends should get less. (This assumes that the buffer sizes are equal.) The distribution should

be symmetric. Similarly, if the machines are identical and there is a pool of storage space, the buffers in the middle of the line should get more, and the buffers at the ends should get less. They called this the "bowl phenomenon" because of the shape of the distribution of operation time (which is the inverse of machine speed).

While this phenomenon is a real feature of mathematical models of production lines, there is some question as to whether it is of practical importance. As indicated above, Smunt and Perkins (1985) did not feel that it generally had a large impact. Numerical results of Hillier and Boling (1972) indicate that the gain over an equal distribution is usually less than 1%. Not only might this gain be negated by errors in system parameters, but the reason for the bowl shape is the usual assumption that the first machine is never starved and the last machine is never blocked. This implies that the machines at the ends of the line are forced to be idle less than those in the middle. The bowl shape compensates for this. In a real system, however, the first machine is occasionally starved when suppliers fail to deliver raw material, and the last machine is blocked during temporary declines in demand.

3.10 Realistic Values of Parameters

In most real systems,

(1) all machines operate at approximately the same speed while they are operational. Therefore, all μ_i are close to one another. In other words, the line is nearly homogeneous.

(2) there is no machine which is much worse than all the others. Therefore, all $\rho_i = \frac{\mu_i r_i}{r_i + p_i}$ are close to one another.

(3) failures and repairs are much less frequent than operations. Therefore, p_i and r_i are at least an order of magnitude less than μ_i. That is, Assumption MTS is satisfied.

(4) the production rate of the line (P) is not less than half the isolated production rate of a typical machine (ρ_i).

It is always possible to find exceptions to these fuzzy rules of thumb. These considerations are important to those who design approximate methods for analyzing these systems. If a method works well under these conditions, it is likely to be of practical value; if not, it probably will not be. See also the features described in Section 3.9.

4 Exact Analysis

In this section, we describe flow line models that have exact analytic solutions. Such models are important because (1) exact solutions are better than simulations or approximations when the models fit real systems closely, (2) they provide useful qualitative insight into the behavior of systems, and (3) the fact that they can be solved rapidly makes them essential parts of the decomposition and aggregation methods that are described in Section 5.

Most of the results pertaining to the exact analysis of flow line models are based on Markovian analysis. In order to be able to describe the behavior of the flow line by a Markov process, the distributions have to be of special form: exponential or, more generally, continuous phase-type

distributions in the case of continuous time models; geometric or, more generally, discrete phase-type distributions in the case of discrete time models. (See the Appendix.) However, there are some exceptions for which the analysis is not based on Markov models. These exceptions are most often encountered in flow lines with no intermediate storage.

The literature pertaining to this special case is discussed in Section 4.1. Section 4.2 describes the literature of two-machine FLRM (Flow Line with Reliable Machines) models and their solutions. The literature of two-machine FLUM (Flow Line with Unreliable Machines) models is presented in Section 4.3. Although the systems are sometimes equivalent, as we argue in Section 3.7, it is helpful to treat them separately. This is especially because different approximate methods of Section 5 were designed with different kinds of systems in mind. In Section 4.4, we describe exact methods for three-machine and larger systems.

We try to indicate which of the models in the literature are blocking-after-service systems (BAS), and which are blocking-before-service systems (BBS). This is not always straightforward, because early authors were evidently not concerned with this distinction. In the two-machine case, the distinction is not important, and does not warrant new research, because each can be transformed into the other; see Section 3.4. See the review by Perros (1988) for other papers on two-stage systems.

Notation. In the case of two-machine flow line, we drop the index of parameters pertaining to the buffer. That is, $N_{1,2}$ and $C_{1,2}$ are simply denoted by N and C, respectively.

4.1 Flow Lines with No Intermediate Storage

In this section, we restrict our attention to the case of flow lines with no intermediate storage, that is, $C_{i,i+1} = 0$, for all $i = 1, ..., K-1$. We review the part of the literature that is devoted to this special case. The aim is to determine the average production rate of the line. Most often, BAS is assumed.

Flow lines with no intermediate storage can operate under two transfer modes: **asynchronous transfer mode** and **synchronous transfer mode**. In the former case, parts can move independently as long as they are not blocked. In the latter case, there is always exactly one part on each machine and the transfer of all parts occurs simultaneously at the instant when all machines have completed their processing. Note that the asynchronous transfer mode is implicitly assumed in the asynchronous model described in Section 2.1. Let $P(AT)$ and $P(ST)$ denote the production rate of a flow line operating under the asynchronous transfer mode and synchronous transfer mode, respectively. It can be shown (Muth, 1973) that a system operating under synchronous transfer mode is less efficient than when operating under asynchronous transfer mode, i.e.:

$$P(ST) \leq P(AT) \tag{32}$$

This relationships hold for any flow line with any number of (reliable or unreliable) machines. In the special case of two-machine lines, the two production rates are equal, i.e.: $P(ST) = P(AT)$. Indeed, whether or not the part on Machine M_2 can be released independently does not affect the production rate. Consequently, in that case, it is not important to distinguish between the two transfer modes.

Two-machine FLRMs with no intermediate storage and BAS were analyzed by Rao (1975a) and Lau (1986a, 1986b). The production rate can be expressed as

$$P = \frac{1}{E[\max(S_1, S_2)]} \tag{33}$$

where S_i denotes the random variable corresponding to the processing time of Machine M_i, $i = 1, 2$. Consequently, the problem reduces to that of calculating the expected value of the maximum of two random variables. Rao (1975a) and Lau (1986a, 1986b) gives explicit expressions for calculating the production rate for different distributions of processing times, especially exponential, Erlang, uniform, and normal distributions.

For FLRMs with more than two machines we need to distinguish between the two transfer modes. In the case of synchronous transfer, the analysis is similar. Indeed, the production rate is now given by $P = 1/E[\max(S_1, ..., S_K)]$. Again, the problem reduces to that of calculating the expected value of the maximum of K random variables. In the case of asynchronous transfer, the analysis is much more complex. The case of three machine flow lines was deeply investigated by Hunt (1956), Hillier and Boling (1967), Hildebrand (1968), Muth (1973, 1977, 1984), Muth and Alkaff (1987), Rao (1976a, 1976b).

The approach introduced by Muth (1973) is fairly general. It is based on the analysis of the **holding time** distribution at the first machine. The holding time is the total time spent by a part on this machine. It is the sum of the processing time and possibly some blocking time. The holding time at Machine M_1, H_1, can be expressed as (Muth, 1984)

$$H_1 = \max(S_1, S_2, R_3) \tag{34}$$

where R_3 is the residual processing time of the part at Machine M_3 at the time at which the next part is transferred from Machine M_1 to Machine M_2. The production rate can then be obtained as

$$P = \frac{1}{E[H_1]} \tag{35}$$

Therefore, the problem mainly reduces to that of determining the distribution of R_3. Numerical solutions have been obtained by Muth (1977, 1984) and Rao (1976a, 1976b) for specific distributions of the processing times. A unifying solution is provided by Muth and Alkaff (1987) under the following assumptions: Machines M_1 and M_3 have special phase-type distributions, while Machine M_2 has a Laplace transformable distribution. Unfortunately, it does not seem that the analysis is extendible to flow lines consisting of more than three machines.

We now turn our attention to FLUMS with no intermediate buffer. First, we note that all the above results are still applicable provided that processing times are replaced by completion times. Buzacott (1968) considered the case of flow lines with synchronous transfer, deterministic processing times and general distributions of uptimes and downtimes. Let T denote the common processing time of all machines. He obtained the following simple formula for the production rate

$$P = \frac{1}{T} \frac{1}{1 + \sum_{i=1}^{K} \frac{MTTR_i}{MTTF_i}} \tag{36}$$

This formula was obtained under the following assumption (Buzacott, 1968): if two machines break down in the same cycle, the total duration of line stoppage is the same as the sum of stoppage durations of the machines when the machines break down in different cycles. This formula also holds under the following alternative assumption: when a machine goes down, all other machines stop their processing; the processing of all machines resumes as soon as the machine is repaired. Note that this assumption implies that at most one machine may be down at any time.

When neither of these assumptions satisfied, Buzacott's formula provides a good approximation of the exact production rate provided that the times to failures are large compared to the processing time T (Assumption MTS). For models for which this is not true, Commault and Dallery (1990) propose a method for calculating the production rate under the assumption that uptimes are exponentially distributed. It is shown that the analysis reduces to the calculation of the expected value of the maximum of Coxian distributions. Approximations are also provided.

4.2 Flow Lines with Two Reliable Machines

In this section, we discuss the exact solutions of two-machine FLRMs with finite buffers. In Section 4.2.1 we describe the relationship between two-machine FLRMs with finite buffers and single server queues with finite buffers. In Section 4.2.2 we describe exact methods for analyzing two-machine FLRMs with finite buffers and phase-type distributions of operation times. By analyzing, we mean calculating the steady state distribution, which can then be used to determine production rates and other performance measures. This technique is closely related to the methods described in Section 4.3.

4.2.1 Equivalence of Two-Machine Flow Lines with Finite Queues

In this section, we briefly discuss the equivalence of two-machine flow lines with finite single server queues (FSSQs). A FSSQ consists of a single server fed by an external arrival process of customers. There is a finite buffer in front of the server in order to accommodate the customers that arrive while the server is busy. Let L denote the capacity of the buffer including the space in front of the server. Thus, L is the maximum number of customers that can be present in the system, either receiving or waiting for service. We further need to define the behavior when the queue is full. There are two different assumptions and the corresponding models are referred to as the **lost model** and the **switch-off model** (Buzacott and Kostelski, 1987).

In the lost model, it is assumed that the arrival process is never stopped and that customers arriving while the queue is full are lost. In the switch-off model, it is assumed that the arrival process is switched off as soon as the queue is full. The arrival process is restarted at the instant when a space becomes available in the buffer which occurs when the next service is completed. Note that in queueing theory, the lost model is usually considered (Kleinrock, 1975). The lost and switch-off models are not equivalent, in general. However, they are equivalent if the distribution of the interarrival process of customers is exponential. This follows from the memoryless property of the exponential distribution (Kleinrock, 1975).

There are equivalences between two-machine flow lines and FSSQs with switch-off arrival process. Consider first a two-machine line with blocking-before-service and buffer capacity N.

Then, the behavior of this system is equivalent to the behavior of the following FSSQ with switch-off arrival process: the interarrival time distribution is the same as the processing time distribution of Machine M_1; the service time distribution is the same as the processing time distribution of Machine M_2; the buffer capacity is $L = N$. Now, if instead of BBS, the two-machine line operates under blocking-after-service, then the same equivalence holds provided that the buffer capacity is $L = N + 1$.

In summary, by combining the above results, we have the following equivalences: (1) for any two-machine flow line, there exists an equivalent FSSQ with switch-off arrival; and (2) for any two-machine line for which the first machine has exponential processing times, there exists an equivalent FSSQ with lost arrivals. In both cases, the queue capacity is increased by one if BAS is assumed.

Now, consider the case of a two-machine flow line with exponential and general processing time distributions for the first and second machines, respectively. As stated above, it is equivalent to a $M/G/1/L$ queue (Kleinrock, 1975). Consequently, its solution can be obtained as the solution of the $M/G/1/L$ queue which is based on the analysis of the embedded Markov chain at departure instants (Cohen, 1982). Note that in the special case where the processing time distribution of the second machine is also exponential, the analysis reduces to that of a $M/M/1/L$ queue whose solution has a very simple geometric form. Finally, we note that the case of a two-machine flow line with general and exponential processing time distributions for the first and second machines, respectively, can also be analyzed as a $M/G/1/L$ queue using the duality property of two-machine flow lines (see Section 3.6). Two-machine flow lines of the above types were analyzed by Rao (1975b).

4.2.2 Analysis of Two-Machine Flow Lines with Phase-Type Distributions

Consider a two-machine FLRMs where the processing time distribution of each machine is given in the form of phase-type distribution. (Recall that a Coxian distribution is a special case.) Let PH_i refer to the phase-type distribution of Machine M_i, for $i = 1, 2$, and let s_i be the number of phases of PH_i. The behavior of such a system can be characterized by a discrete state, continuous time Markov process. Analyzing this system then reduces to that of calculating the steady-state probabilities of this Markov process, from which all the performance parameters can be derived.

Any numerical technique for discrete space Markov processes can in principle be used (Stewart, 1978 and 1988; Philippe, Saad, and Stewart, 1989). However, it is helpful to recognize that the Markov process has a very special structure and to take advantage of it. In the following presentation, we assume that the blocking mechanism is BBS. However, the type of blocking is not at all important since we know that in the case of two-machine lines, there is equivalence. (See Section 3.4.)

The state of the Markov process can be expressed as (n, j_1, j_2), where n is the number of parts currently present in the buffer (including the part on Machine M_2, if any), and j_i is the current phase of service of Machine M_i, $i = 1, 2$. n can take on integer values from 0 to N. j_1 can take on integer values from 1 to s_1, except when Machine M_1 is blocked in which case we set $j_1 = 0$. Similarly, j_2 can take on integer values from 1 to s_2, except when Machine M_2 is starved in which case we set $j_2 = 0$. Let us partition the state space according to the values of

n. Let $\boldsymbol{p}$ denote the steady-state probability vector and let $\boldsymbol{p}_n$ denote the portion of that vector that corresponds to a buffer content of n. We can write

$$\boldsymbol{p} = \begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{p}_1 \\ \\ . \\ . \\ \boldsymbol{p}_N \end{pmatrix} \tag{37}$$

Note that $\boldsymbol{p}_n$, $n = 1, ..., N-1$, is of size $s_1 s_2$ while $\boldsymbol{p}_0$ and $\boldsymbol{p}_N$ are of size s_1 and s_2, respectively.

Let Q denote the **infinitesimal generator** of the Markov process. The steady-state probability vector $\boldsymbol{p}$ of the Markov process is the solution of the equation $\boldsymbol{p}^T Q = 0$; or, equivalently,

$$Q^T \boldsymbol{p} = 0. \tag{38}$$

(In this presentation, we choose to use this equation and therefore we deal with Q^T.) In addition, $\boldsymbol{p}$ also satisfies the **normalization equation**

$$\mathbf{1}^T \boldsymbol{p} = 1. \tag{39}$$

Matrix Q^T is a **block tridiagonal** matrix with the following special structure

$$Q^T = \begin{pmatrix} B_0 & A_0 & 0 & . & . & . & 0 \\ C_0 & B & A & 0 & . & . & . \\ 0 & C & B & A & 0 & . & . \\ & . & . & . & . & . & . \\ & . & 0 & C & B & A & 0 \\ & . & . & 0 & C & B & A_N \\ 0 & . & . & . & 0 & C_N & B_N \end{pmatrix} \tag{40}$$

where A, B, and C are square matrices of size $(s_1 s_2, s_1 s_2)$; B_0 and B_N are square matrices of size (s_1, s_1) and (s_2, s_2); A_0, C_0, A_N, and C_N are of size $(s_1 s_2, s_1)$, $(s_1, s_1 s_2)$, $(s_2, s_1 s_2)$, $(s_1 s_2, s_2)$.

Q^T has this special structure because the Markov process associated with a two-machine flow line is a **generalized birth-death process**. Transitions can only occur between states that are neighbors of each other with respect to the value of n. That is, the only possible transitions from a state (n, j_1, j_2) are to a state (n', j_1', j_2') such that either $n' = n$, or $n' = n-1$, or $n' = n+1$. In addition, transition rates are independent of n, for $1 < n < N-1$. Because of the special block tridiagonal structure of Q^T, equation (38) can be decomposed into the following set of equations

$$B_0 \boldsymbol{p}_0 + A_0 \boldsymbol{p}_1 = 0 \tag{41}$$

$$C_0 \boldsymbol{p}_0 + B \boldsymbol{p}_1 + A \boldsymbol{p}_2 = 0 \tag{42}$$

$$C \boldsymbol{p}_{n-1} + B \boldsymbol{p}_n + A \boldsymbol{p}_{n+1} = 0 \ , \ 1 < n < N-1 \tag{43}$$

$$C \boldsymbol{p}_{N-2} + B \boldsymbol{p}_{N-1} + A_N \boldsymbol{p}_N = 0 \tag{44}$$

$$C_N \boldsymbol{p}_{N-1} + B_N \boldsymbol{p}_N = 0 \tag{45}$$

Two solution techniques that make use of the special structure of the matrix Q^T have received special attention. They are known as the **recursive technique** (Herzog, Woo and Chandy, 1975, Chandy and Sauer, 1981; Stewart, 1988) and the **matrix geometric technique** (Neuts, 1981). Recursive and matrix geometric techniques to analyze two-machine FLRMs have been used, among others, by Altiok and Ranjan (1987), Buzacott and Kostelski (1987), Gun and Makowski (1987). Another approach, which has only been applied to two-machine FLUM systems, is described in Section 4.3. There is no reason why this method could not be used for the present class of systems as well.

The recursive technique can be applied to Markov processes that satisfy the following condition: there exists a subset of states, called the **boundary states**, such that the probabilities of all other states can be obtained recursively from the probabilities of the boundary states. The recursive technique is usually implemented using the following procedure (Buzacott and Kostelski, 1987). Let k be the total number of states of the Markov process. Determine a subset of m boundary states satisfying the above condition. Obtain a recursive scheme to derive the non-boundary state probabilities. This uses $k-m$ equations among the total of k balance equations. Then, express all non-boundary state probabilities as a linear function of the probabilities of the boundary states. The coefficient of a particular boundary value in the linear expression is obtained by setting that boundary value equal to 1 and all other boundary values equal to 0, and then follow the recursive scheme. The remaining m balance equations can then be used together with the normalizing equation, (39), to determine the probabilities of the boundary states, from which the other probabilities can then be derived.

It is important to note that the phrase "boundary states" has a different meaning elsewhere in this paper, and in most of the rest of the literature. Boundary states, as defined here, are usually a subset of boundary states, as defined in Section 4.3.

Buzacott and Kostelski (1987) applied this recursive technique in the case where each machine has a Coxian-2 processing time distribution. The states corresponding to an empty buffer ($n = 0$) are chosen as the boundary states. In this simple case, there are only two boundary states: $(0, 0, 1)$ and $(0, 0, 2)$. In the general case of phase-type distributions, the recursive technique can still be applied, for instance, by considering the states corresponding to a buffer level of $n = 1$ as the boundary states.

The principle of the matrix geometric solution can briefly be described as follows (Gun, 1987, Gun and Makowski, 1987). The first step is to show that the set of transition equations (41) to (45) can be transformed into an equation of the following form:

$$N \boldsymbol{p}_n + M \boldsymbol{p}_{n-1} = 0 \tag{46}$$

where the matrices N and M are of size $s_1 s_2$ and N is invertible. Let R be the matrix defined as $R = -N^{-1}M$. Then, we have:

$$\boldsymbol{p}_n = R \boldsymbol{p}_{n-1} \ , \ 1 < n < N \tag{47}$$

For the boundary states, it is also possible to define matrices S and U such that $\boldsymbol{p}_1 = S\boldsymbol{p}_0$ and $\boldsymbol{p}_N = U\boldsymbol{p}_{N-1}$. The probability vector $\boldsymbol{p}_0$ can be determined by solving an equation of the

form $Z\boldsymbol{p}_0 = \boldsymbol{x}$. This equation is again obtained from the basic set of equations, (41)-(45). The remaining probabilities can then be obtained using the above equations. See Gun (1987) and Gun and Makowski (1987) for more details.

We note that the matrix geometric solution is also based on a recursive scheme and so it has relationships with the recursive technique described above. Buzacott and Kostelski (1987) compared these two techniques in the case of Coxian-2 distributions. They showed that both techniques can be subject to numerical problems when N is large.

We note that probabilities at specific points in time, especially **completion instant probabilities**, can be derived from the steady-state probabilities; see, e.g., (Gun and Makowski, 1989). Such probabilities are useful when using decomposition methods (Section 5.1). Finally, we note that similar analyses can be done in the case of discrete phase-type distributions; see e.g., Gun (1987). In that case, the underlying model is a discrete time Markov process.

4.3 Flow Lines with Two Unreliable Machines

In this section, we discuss the exact solutions of two-machine FLUMs with finite buffers. These systems are modeled as Markov processes with discrete states and continuous times (Section 4.3.1), as Markov processes with discrete states and discrete times (Section 4.3.2), and as Markov processes with mixed states and continuous times (Section 4.3.3).

4.3.1 Asynchronous Models

Each of the models in this section is described as a Markov process with continuous time and discrete state. The state (n, α_1, α_2) represents the number of parts in the buffer (n) and the condition of the machines (α_1 and α_2). Most often, α_i can take on two values, which we may represent as 1 for operational (up) and 0 for under repair (down). When there are more than one failure mode, or when processing times, uptimes, or downtimes are represented by Coxian or phase-type random variables, or when other states (especially idle) are included, α_i takes on more than two values. In fact, it is possible to combine a Coxian model of operation time and a Coxian model of failure, and apply the methods of Section 4.2.2 and this section to it. The present models are not essentially different from those of the earlier section; α_i here is the same as j_i there.

As in Section 4.2.2, $\boldsymbol{p}$ is the steady-state probability distribution vector. The components of $\boldsymbol{p}$ are $p(n, \alpha_1, \alpha_2)$. Let $\boldsymbol{p}_n$ be the portion of that $\boldsymbol{p}$ that corresponds to a buffer content of n. The dimensionality of $\boldsymbol{p}_n$ depends on the number of different values of α_1 and α_2. When they can take on two values, the usual case in the literature, the dimensionality of $\boldsymbol{p}_n$ is 4. In that case, the dimensionality of $\boldsymbol{p}$ is roughly $4(N+1)$, since n can take on integer values from 0 to N. (We say "roughly" because there may be additional states for idle when the buffer is empty or nearly empty or full or nearly full; or there may be additional states when the buffer is full or nearly full for the piece in the machine's work area for BAS systems; or there may be fewer states if transient states are eliminated. Transient states appear when the buffer is empty or nearly empty or full or nearly full.) $\boldsymbol{p}$ can be written (37) and it satisfies (38) (with an appropriate Q) and (39).

As in Section 4.2.2, it is useful to distinguish between **internal** states and equations and

boundary states and equations. The latter refers to cases when the buffer is empty or nearly empty or full or nearly full. For some models, the boundary consists only of states in which $n = 0$ and $n = N$ (in which case (42) and (44) are exactly the same as (43)); other models also include $n = 1$ and $n = N - 1$. Internal states are all states that are not on the boundary. This terminology comes from differential equations, and some of the solution techniques described below have been influenced by differential equation methods.

Also as in Section 4.2.2, this system is a generalized birth-death process, and $\boldsymbol{p}_n$ satisfies (41)-(45). The approach taken in the two-machine FLUM literature is to satisfy (43) by assuming the following analytic expression for $p(n, \alpha_1, \alpha_2)$ when n is internal:

$$p(n, \alpha_1, \alpha_2) = \sum_j D_j X_j^n \phi_j(\alpha_1, \alpha_2) \tag{48}$$

for some scalars D_j and X_j and some scalar functions ϕ_j (although not all authors write the solution in this form). Equation (48) can also be written

$$\boldsymbol{p}_n = \sum_j D_j X_j^n \boldsymbol{\phi}_j \tag{49}$$

where $\boldsymbol{\phi}_j$ is the vector whose components are $\phi_j(\alpha_1, \alpha_2)$.

Equation (48) follows from (43) when

$$\boldsymbol{p}_n = R^n \boldsymbol{\phi}. \tag{50}$$

is a solution of (43) for some matrix R and vector $\boldsymbol{\phi}$. Yeralan and Muth (1987) demonstrate conditions for this to hold. They call this the **matrix geometric property**. Substituting (50) into (43):

$$AR^{n+1}\boldsymbol{\phi} + BR^n\boldsymbol{\phi} + CR^{n-1}\boldsymbol{\phi} = 0 \tag{51}$$

or,

$$(AR^2 + BR + C)R^{n-1}\boldsymbol{\phi} = 0 \tag{52}$$

so (43) is satisfied if R satisfies

$$AR^2 + BR + C = 0. \tag{53}$$

Let $X_1, ..., X_l$ be the eigenvalues of R, and $\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_l$ be the corresponding eigenvectors. If we write

$$\boldsymbol{\phi} = D_1\boldsymbol{\phi}_1 + ... + D_l\boldsymbol{\phi}_l, \tag{54}$$

then

$$R\boldsymbol{\phi} = D_1X_1\boldsymbol{\phi}_1 + ... + D_lX_l\boldsymbol{\phi}_l, \tag{55}$$

and similarly,

$$\boldsymbol{p}_n = R^n\boldsymbol{\phi} = D_1X_1^n\boldsymbol{\phi}_1 + ... + D_lX_l^n\boldsymbol{\phi}_l, \tag{56}$$

which is equivalent to (48). When the summation has only one term, Yeralan and Muth (1987) call this the **scalar geometric property**. The coefficients $D_1, \ldots, D_l$ and the boundary probabilities are determined by satisfying the transition equations on the boundary (i.e., (41), (42), (44), and (45), and the normalization equation (39)).

As a consequence of this, all performance measures are functions of X_j^N. As indicated earlier, in all papers in which production rate is evaluated as a function of buffer size, it is a saturating function of storage space. This upper bound is the production rate of the slower — in the sense of isolated production rate — of the two machines.

Buzacott (1972) describes a two-station model with a finite buffer in which the machines are identical. Both the operation times and repair times are exponentially distributed. There is a constant probability of failure during each operation. Consequently, they occur according to a geometric distribution of the number of operations since the last failure. Note that this means that he assumes number-ODFs. A close study of the transition equations suggests that this is a blocking-after-service model. He obtains an exact solution of the model, using generating functions (i.e., z-transforms). This solution is equivalent to (48) with one or two terms. He also shows an approximate relationship between this model and (I) an exponential model with no failures and (II) his deterministic processing time model (Buzacott, 1967a and b). His numerical results demonstrate that production rate is a saturating function of storage space. Sastry (1985) obtained an analytical solution using the method of Gershwin and Berman (1981).

Gershwin and Berman (1981) study the two-machine system in which processing times, times to failure, and times to repair are all exponentially distributed. The model differs from that of Buzacott (1972) in that failure is represented by an exponential distribution in time rather than a geometric distribution in the number of parts produced. Note that this means that they assume time-ODFs. They obtain analytic solutions when the machines differ, and they prove that the model satisfies conservation of flow. The solution technique involves specifying (48) for $\boldsymbol{p}_n$ for internal states ($1 \leq n \leq N-1$), and developing other expressions for boundary states ($n = 0$ or $n = N$). They demonstrate that the summation in (48) has four terms, but one of them (corresponding to $X = 1$) has coefficient $D = 0$. They also obtain some results on the limiting behavior of the system as some of the machine parameters approach 0 or ∞. These results, which characterize the effects of bottlenecks in the system, show that the model is consistent with intuition.

Berman (1982) generalizes the Gershwin and Berman (1981) model by allowing processing times to have Erlang distributions. He also proves conservation of flow and generalizes the limiting results. In numerical examples, the graphs of production rate as a function of machine parameters appear to be increasing, saturating functions (except, of course, for production rate as a function of failure rate), but they appear to be non-convex. Both the Gershwin and Berman (1981) and the Berman (1982) models assumed blocking before service.

Sastry (1985) formulates a two-machine transfer line model in which each machine is subject to the two types of ODFs: time- and number-ODFs. One mode operates according to Buzacott's (1972) mechanism and the other is based on Gershwin and Berman's (1981). The boundary equations suggest that the system is a blocking-before-service model. Sastry obtains an analytic

solution following Gershwin and Berman's technique.

Sastry (1985) and Sastry and Awate (1988) study a second extension of Gershwin and Berman's (1981) model by including inspection and rework. A part is inspected after its operation is completed, but while it is still on the machine. If it fails inspection, it is processed again, until it passes inspection. Inspection time is assumed exponentially distributed. Machine failures are allowed during both operation and inspection phases. This system is also a blocking-before-service model. Again, Gershwin and Berman's technique is used to generate an analytic solution.

4.3.2 Synchronous Models

Each of the papers in this section models a flow line as a Markov process with discrete time and discrete state. As in Section 4.3.1, the state (n, α_1, α_2) represents the number of parts in the buffer and the condition of the machines. The approach is similar to the case of asynchronous systems except that one has to deal with a stochastic matrix instead of the infinitesimal generator. Equations similar to (41)-(45) and (48) for $\boldsymbol{p}_n$ are valid here, performance measures are still functions of X_j^N, and production rate is still a saturating function of storage space.

Buzacott's early papers, (1967a and b, 1969), while not the first to focus on flow lines, were possibly the most influential. They covered a great many topics, including several models of synchronous flow lines. In the exact two-machine analyses, both the failure and the repair processes are geometric or deterministic. He simplifies his transition equations by assuming that the probability that two events happen during the same cycle is negligible. He relaxed this assumption later in Buzacott and Hanifin (1978a). The blocking behavior is evidently blocking-after-service, since failures are allowed when the buffer is full.

Analytic solutions of special cases of the models appear in Buzacott (1967a). The solution of the geometric-repair-time-geometric-failure-time line is of the form of equation (48) (with a single term); the solution of the constant-repair-time-geometric-failure-time line may also be of that form, but it is rather complicated. Numerical examples show that the production rate for constant-repair-time-geometric-failure-time lines is greater than for geometric-repair-time-geometric-failure-time lines. In both models, production rate is a saturating function of buffer size. A more general model is also described in Buzacott (1967a): one with general repair time distributions. Because of the influence of Buzacott's work, we refer to synchronous geometric-repair-time-geometric-failure-time flow models as **Buzacott-type models** in the rest of the paper.

Okamura and Yamashina (1977) formulated a Buzacott-type model and solved it numerically. (It is not clear whether it is a blocking-before-service or blocking-after-service model.) They perform an extensive set of numerical experiments to explore the behavior of performance measures (production rate and average buffer level) as a function of system parameters (failure and repair probabilities and buffer size). Production rate is always a saturating function of buffer size. The shape of the average-buffer-level vs. buffer size curve depends on the relative values of the machine parameters.

Artamonov (1977) studied a two-machine line with deterministic processing time and geometric repair and failure times — a model similar to Buzacott's. Without making Buzacott's approximating assumption (that the probability of two events during the same cycle is negligi-

ble), he obtains a solution in the form of equation (48) with one term. Dudick (1979), Schick and Gershwin (1978), and Gershwin and Schick (1983) obtained essentially the same solution of the same system. (According to Liu (1990), the Gershwin-Schick model with a buffer of size N is close to the Buzacott model with a buffer of size $N-2$.) Schick and Gershwin (1978) also observed (43) and (50) and demonstrated (9) and conservation of flow (13). In their numerical experiments, they showed the saturation curve for production rate vs. buffer size, and they investigated the shape of the average buffer level vs. buffer size curve. Based on equation (48) (in which there is only a single term), they showed that if the first machine is slower than the second (in the sense of isolated efficiency) the average buffer level approaches a limit as the buffer size increases without limit. The same approach could have been used to demonstrate that, if the first machine is faster than the second, the average buffer level is unbounded as the buffer size increases without limit. (In fact, the part/hole duality of Section 3.6 can be used to show that the difference between the buffer size and the average buffer level approaches a constant.) Finally, Schick and Gershwin's approach could also have shown that if the two machines have $r_1 = r_2$ and $p_1 = p_2$, then the average buffer level is a constant fraction — half — of the buffer size.

Yeralan and Muth (1987) present a very general view of the two-machine transfer line. Although they assumed synchronous transfers, their approach should work equally well for asynchronous systems. By observing (43), they established a relationship with Neuts' (1981) matrix geometric systems. They demonstrated some very simple, general formulas for production rate and average buffer levels that are functions of the matrices in (43) as well as others that describe the blocked and starved behavior of the system. This allowed them to compare very easily a set of systems based on very different assumptions (concerning operating policy, the number of repair personnel, etc.). This work provides an important unifying view of these models. It may be thought of as a summary and generalization of all the previous literature on two-machine synchronous transfer lines with finite buffers.

Other work on synchronous models is described in Section 4.4.

4.3.3 Continuous Flow Models

Each of the models in this section is described as a Markov process with continuous time and mixed state (i.e., with both discrete and continuous components). The state (x, α_1, α_2) represents the amount of material in the buffer (x) and the condition of the machines (α_1 and α_2). The major feature of the continuous model is that the quantity x is a real number. As in discrete state models, α_i usually takes on two values, which we represent as 1 for operational and 0 for under repair. This is the case when uptimes and downtimes are exponentially distributed. Phase-type distributions would lead to models in which α_i would take on more than two values. See De Koster (1988a).

In these systems, the boundary is the set of states where $x = 0$ or $x = N$, and all other states are internal. The steady state is described by a probability density function in the interior and probability masses on the boundary. Let the density be given by $f(x, \alpha_1, \alpha_2)$. That is, $f(x, \alpha_1, \alpha_2)\delta x$ is the probability that the machine states are α_1 and α_2 and the amount of material in the buffer is between x and $x + \delta x$. The masses are $p(0, \alpha_1, \alpha_2)$ and $p(N, \alpha_1, \alpha_2)$ for some α_1 and α_2. There are masses at $x = 0$ and $x = N$ because sometimes, when the buffer

becomes empty, it remains empty until a repair or failure, and sometimes, when the buffer becomes full, it remains full until a repair or failure.

Because $f(x, \alpha_1, \alpha_2)$ is a function of a continuous variable, it satisfies a set of differential, rather than difference equations. If we define $f(x)$ as the vector whose components are $f(x, \alpha_1, \alpha_2)$, these equations can be written

$$\frac{df}{dx} = Af \tag{57}$$

where A is a matrix. This equation is analogous to (43). The solution to (57) can be written as a matrix exponential, or, more usefully, as

$$f(x, \alpha_1, \alpha_2) = \sum_j D_j e^{C_j x} \phi_j(\alpha_1, \alpha_2) \tag{58}$$

This expression is analogous to (48). The only difference is one of notation: now e^{C_j} takes the place of X_j. To complete the model, one must define the behavior when the buffer is empty and full. This leads to probability masses because the probabilities of finding the buffer empty or full are non-zero.

There is an important qualitative difference, in this class of models, between systems in which the machines operate at the same speed when they are operational, and systems in which the machines operate at different speeds when they are operational. This is because, when the machines operate at the same speed, the buffer level stays constant while both machines are up. However, if the machines operate at different speeds, the buffer level changes while they are both up, and the buffer can become empty or full (depending on which machine is faster).

In systems with machines of different speeds and operation dependent failures (ODFs), the boundary conditions depend on the cause of the buffer becoming empty or full. For example, if the first machine is slower than the second, the buffer will frequently be empty. However, this is different from a failure of the first machine because the second machine can still operate. It is slowed down and operates at the speed of the first machine. This is less than its normal rate, so if its failures are operation dependent, it ought to fail less often than otherwise (see the discussions in Sections 2.3 and 3.8.1). As a consequence, the boundary conditions of (57) are affected.

Zimmern (1956) was one of the first papers in the transfer line literature, and is still not as widely known as it deserves to be. Perhaps this is because it is based on the continuous material flow assumption, whereas the more fashionable models are based on discrete material. Like most papers in this section, Zimmern (1956) assumes exponentially distributed up and down times. The two machines can be completely different, so seven parameters are required to describe the system: the operating speeds, the MTTF's, and MTTR's of the machines, and the buffer size. Zimmern assumes time dependent failures. (As a result, when one machine is slowed down by the other, he does not adjust the failure rate.) Among other things in this rich paper, equation (57) is presented in detail, along with a discussion of boundary conditions, a complete solution, and a graph of production rate vs. buffer size that demonstrates the saturation shape.

Sevast'yanov (1962) analyzed a two-machine continuous material system with exponentially distributed failure and repair times. He assumed that all machines operated at the same rates and had equal repair rates. He obtained an analytic solution and extended it to an approximation

for longer lines which is described in Section 5.2.1. He assumed time-dependent failures (TDF) and that only one machine in a line is allowed to be down at once.

Winjgaard (1979) formulated the problem differently. His system is essentially the same as Zimmern's. However, instead of determining a steady state probability distribution and calculating performance measures from it, he postulates a cost for being in each state, and evaluates the average value of the cost. The production rate is found by choosing the cost functions appropriately.

Gershwin and Schick (1980) formulated a model similar to Zimmern's (1956). However, they assumed ODFs instead of TDFs. Moreover, they assumed that the failure rate of a slowed down machine was proportional to the speed that it operated at. See Section 3.8.1. They obtained a complete solution, and evaluated the average in-process inventory. They also investigated the relationship between two-machine synchronous models with geometrically distributed up- and down-times and continuous material models with exponentially distributed up- and down-times. They found conditions under which one would be a good approximation for the other. See Section 3.8.2. A similar study was performed by Dubois and Forestier (1982). They obtain the usual saturation curve for production rate as a function of buffer size. They also verified Okamura and Yamashina's (1977) observations on average buffer level.

Glassey and Hong (1986a) also analyze a continuous model with ODFs. However, as opposed to Gershwin and Schick (1980) and Dubois and Forestier (1982), they did not reduce the failure rate of a slowed down machine. This is not consistent if the model is to be used as an approximation of an asynchronous model (see Section 3.8.1).

De Koster (1989a) shows that continuous models with general up- and downtimes can also be handled provided that the distributions are of phase-type. However, the analysis becomes much more complex.

4.4 Longer Lines

Sheskin (1974, 1976), and Soyster, Schmidt, and Rohrer (1979) studied Buzacott-type synchronous models in which, for each machine, $r_i + p_i = 1$. With this restriction, the probability that a machine is down in any time step is p_i, independent of its state in the previous time step. This allowed the transition equations to be solved more easily for systems with more than two machines, and allowed Soyster, Schmidt, and Rohrer (1979) to obtain bounds on production rate. The consequences of this restriction were studied further by Lim, Meerkov, and Top (1990), Lim and Meerkov (1990), and Top (1990) when all p_i are small. Under this condition, easily computable asymptotic results can be obtained.

Buzacott (1967a) extended his two-machine synchronous system equations to a three-machine transfer line with geometric up-times and general, but identical, repair time distributions. He approximated the probabilities — and reduced the number of equations — by assuming that two or more machines would not be down at the same time. Because the problem was still too large to treat practically, he further specialized it to a system with equal deterministic repair times, in which the buffer sizes were small integer multiples of the repair time. (In the course of this, he observed a symmetry which was later studied in a more general context: if the buffers are the same, and the first and third machines are identical, then the probability of n_1 in the first buffer and n_2 in the second is the same as the probability of $N - n_1$ in the first buffer and

$N - n_2$ in the second.)

Gershwin and Schick (1983) attempted to extend their analytic solution of the two-machine Buzacott model to three machines. To do this, they had to extend the analysis of internal states, which was easy, and the analysis of boundary states, which was hard. A state is on the boundary if one or more of the buffers is on the boundary, as defined above for single-buffer systems. They analyzed the internal states by extending (48). This satisfied all the internal transition equations. However, the extension to the boundary did not satisfy all the boundary equations. The number of equations that remained was linear in the buffer sizes, and that many linear equations would have to be satisfied by some general method. As a result, the reduction in computational effort was not great enough to solve three-machine systems with large buffers. More generally, this kind of method would reduce the computational effort of solving a K-machine line to that of solving a $K-1$-machine line by some general linear equation method. This is not enough of a reduction to be practical. Wiley (1981) made a more sophisticated attempt to extend the Gershwin-Berman (1981) technique to a three-machine asynchronous line. He developed an eigenvalue-like technique that allowed him to transform the boundary equations to a smaller set of equations. However, he was only able to reduce the number of equations by a factor of approximately 2 in the three-machine case, and this does not suggest greater savings for larger systems.

Zimmern (1956) stated the internal partial differential equations for the continuous material long line. Coillard and Proth (1984) formulated and solved a three-machine continuous material flow model in which all the machines are identical.

Finally, we note that any asynchronous flow line model (with or without failures) can be described as a discrete space continuous time Markov process, provided that all distributions are given under phase-type forms. Thus, in principle, they can be analyzed by solving the steady-state probability vector equation, (38), using any appropriate numerical technique, especially iterative techniques (Stewart, 1978; Philippe, Saad and Stewart, 1989). Alternatively, the matrix-geometric approach of Neuts may be applied (Neuts, 1981). However, the number of states of the Markov chains grows very fast with the number of machines, the buffer capacities, the number of phases of the distributions. As a consequence, only models of limited sizes are tractable. A similar statement applies to synchronous models. Studies along this line include the work of Altiok and Stidham (1983), Onvural, Perros, and Altiok (1987), and Fanti, Maione, Peluso, and Truchiano (1987).

5 Approximate Analysis

It appears from Section 4 that exact solutions of two-machine flow lines (either FLRMs or FLUMs) are available for a wide range of models. However, it seems hopeless to expect to obtain exact solutions of flow lines with more machines even when more powerful computers are available. The work on three-machine lines reviewed in Section 4.4 involves models that are not tractable, or are subject to numerical problems, or are too limited to be of interest. Therefore, the use of approximate solutions is the only viable alternative. The purpose of this section is to review the literature devoted to **approximate methods**.

Most approximate methods are based on **decomposition**. The common idea is to decompose

the analysis of the original model into the analysis of a set of smaller subsystems which are easier to analyze. Each decomposition method involves three steps: (1) characterizing the subsystems; (2) deriving a set of equations that determines the unknown parameters of each subsystem; and (3) developing an algorithm to solve these equations. The aim of the first step is to define how the original line is decomposed into subsystems and to characterize each subsystem. The subsystems must have exact solutions. The purpose of the second step is to establish relationships between quantities pertaining to different subsystems so that the parameters of each subsystem can be derived from the parameters and performance measures of other subsystems. In general, the set of equations can be expressed in the following form:

$$\boldsymbol{x} = F(\boldsymbol{x}) \tag{59}$$

where $\boldsymbol{x}$ denotes the vector of unknown parameters. In other words, the problem of determining the unknown parameters reduces to that of finding a vector $\boldsymbol{x}$ that satisfies the fixed-point equation (59). Thus, the third step usually consists of deriving an iterative procedure of the following type:

$$\boldsymbol{x}^{(j)} = F(\boldsymbol{x}^{(j-1)}) \tag{60}$$

where $\boldsymbol{x}^{(j)}$ is the estimate of $\boldsymbol{x}$ at the j'th step of the iteration procedure.

Most decomposition methods in the literature decompose a K-machine flow line into a set of $K-1$ subsystems, each subsystem being associated with a buffer of the original line. Decomposition methods are approximations because (1) the subsystems are always simpler than the whole line, and so cannot exhibit the same behavior; and (2) some of the equations used to determine the parameters may be approximate, even within their assumptions. In decomposition methods, there is a trade-off between complexity and accuracy. Indeed, a more complex characterization of subsystems will generally lead to a better approximation of the behavior of the original line and, as a result, to more accurate results. However, obtaining the exact solution of subsystems will also be more complex and, since each subsystem must usually be solved several times, the overall computational complexity will be greater.

Decomposition methods have been designed either for FLRMs or for FLUMs. The basic principles of decomposition methods were devised by Zimmern (1956) and Sevast'yanov (1962) in the context of FLUMs, and by Hillier and Boling (1967) in the context of FLRMs. Since then, much work has been devoted to approximate methods. Sections 5.1 and 5.2 describe the literature of decomposition methods dedicated to FLRMs and FLUMs, respectively.

5.1 Flow Lines with Reliable Machines

5.1.1 Decomposition

As stated above, most decomposition methods decompose the original flow line with K machines into a set of $K-1$ subsystems, each subsystem being associated with a buffer of the original line. In some methods the subsystem is a two-machine line while in others the subsystem consists of a single server queue with a finite buffer. Because of the equivalences between two-machine lines and FSSQs discussed in Section 4.2.1, there are no major differences between these choices.

We first describe the principle of decomposition methods that decompose the original K machine (FLRM) line into a set of $K-1$ two-machine (FLRM) lines (Altiok and Ranjan, 1987; Dallery and Frein, 1989a; Gun and Makowski, 1989). Each two-machine line is associated with a buffer of the original line. Let L denote the original line and let $L(i,i+1)$ denote the two-machine line associated with buffer $B_{i,i+1}$. Except for the names of the buffers $B_{i,i+1}$, we use parentheses to refer to objects and parameters of the two-machine lines. Moreover, we use subscripts u and d to refer to objects and parameters of the upstream and downstream machines. Machine $M_u(i,i+1)$ is the upstream machine of line $L(i,i+1)$, and $M_d(i,i+1)$ is the downstream machine.

The basic idea of decomposition is to define upstream and downstream machines for each two-machine line $L(i,i+1)$ such that the behavior of material through its buffer is close to the behavior of material in buffer $B_{i,i+1}$ in line L. That is, an observer in the buffer of line $L(i,i+1)$ would see almost the same arrivals and departures, starvations and blockages, and buffer level dynamics as an observer in the buffer of the original line. In other words, upstream and downstream machines of each two-machine line summarize the effects of the entire upstream portion of the line and the entire downstream portion of the line, respectively, on the buffer. For instance, Machine $M_u(i,i+1)$ represents in an aggregate way the portion of line L upstream of Buffer $B_{i,i+1}$, that is, Machine M_1 to Machine M_i. Similarly, Machine $M_d(i,i+1)$ represents in an aggregate way the portion of line L downstream of Buffer $B_{i,i+1}$, that is, Machine M_{i+1} to Machine M_K. These machines are sometimes called **equivalent machines** (although they are not exactly equivalent), **pseudo-machines**, or **virtual machines**. Exact solutions of the subsystems can be obtained using an appropriate technique among those discussed in Section 4.2.

This decomposition approach is expressed symbolically in Figure 4. Note that two pseudo-machines correspond to each real machine (other than the first and last). Indeed, Machine $M_d(i-1,i)$ in Line $L(i-1,i)$ and Machine $M_u(i,i+1)$ in Line $L(i,i+1)$ correspond to Machine M_i. For the behavior of each subsystem to closely match that of the corresponding portion of the line, it is reasonable to assume that (1) the capacity of the buffer of each subsystem is the same as that of the corresponding buffer of the original line; (2) the type of blocking (BAS or BBS) of the subsystems is the same as that of the original line.

Alternatively, some methods decompose the original K-machine line into a set of $K-1$ finite single server queues (FSSQs), e.g. Altiok (1982), Hillier and Boling (1967), Perros and Altiok (1986), Takahashi, Miyahara, and Hasegawa (1980). In this approach, the idea is to determine the arrival process and service process such that the behavior of material in the buffer closely approximates that of the corresponding buffer in the original line. Let $Q(i,i+1)$ denote the FSSQ associated with Buffer $B_{i,i+1}$. The arrival process of $Q(i,i+1)$ models the portion of line L upstream of Buffer $B_{i,i+1}$ while the service process models the portion of line L downstream of Buffer $B_{i,i+1}$.

Because of the equivalences reported in Section 4.2.1, this approach is equivalent to decomposing the line into a set of two-machine lines provided that (1) a switch-off arrival process is assumed, and (2) the buffer capacity of $Q(i,i+1)$ is $N_{i,i+1}$ in the case of BBS and $N_{i,i+1}+1$ in the case of BAS. We note that the assumptions considered in the papers that use this approach may differ from these assumptions. A discussion of this issue is provided at the end of this section.

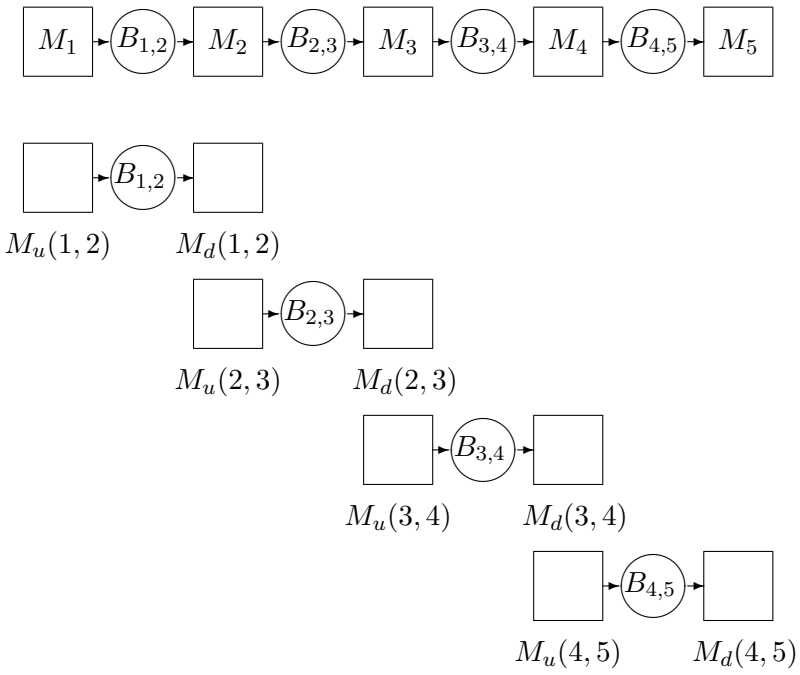

Figure 4: Flow Line Decomposition

In the following, because of the above equivalence, we only use the two-machine line decomposition approach. Also, since most methods are devoted to flow lines with blocking-after-service, we restrict our attention to this case. Decomposition methods for flow lines with blocking-before-service are based on similar approaches, although the detailed analysis is slightly different; see Gun and Makowski (1989). A unified view of decomposition methods for flow lines with blocking-after-service has been presented by Dallery and Frein (1989a). The following presentation is based on this unified view.

Consider the decomposition of a K-machine flow line with BAS into $K-1$ two-machine lines. Let f_i denote the probability density function (PDF) of the processing time of Machine M_i. Consider Subsystem $L(i,i+1)$. It has buffer capacity $N(i,i+1) = N_{i,i+1}$ and operates under blocking-after-service. We define the following quantities pertaining to Subsystem $L(i,i+1)$. Let $f_u(i,i+1)$ and $f_d(,i+1)$ denote the PDFs of the processing times of Machines $M_u(i,i+1)$ and $M_d(i,i+1)$, respectively. Let $T_u(i,i+1)$ and $T_d(i,i+1)$ denote the average processing times of Machines $M_u(i,i+1)$ and $M_d(i,i+1)$, respectively.

We also need to consider quantities related to the blocking of Machine $M_u(i,i+1)$ and to the starvation of Machine $M_d(i,i+1)$. Let $B_u^C(i,i+1)$ denote the probability that Machine $M_u(i,i+1)$ is blocked at the instant of completion of the processing of a part. Also, let $g_d^C(i,i+1)$ be the PDF of the residual processing time of Machine $M_d(i,i+1)$ at this instant. $B_u^C(i,i+1)$ is referred to as the **completion-instant blocking probability**. Similarly, let $S_d^C(i,i+1)$ denote the probability that Machine $M_d(i,i+1)$ is starved at the instant of completion of the processing of a part, and let $g_u^C(i,i+1)$ be the PDF of the residual processing time of Machine $M_u(i,i+1)$ at this instant. $S_d^C(i,i+1)$ is referred to as the **completion-instant starvation probability**. We add the superscript C to emphasize that these quantities are related to special instants in time, namely the instants of processing completions. The completion-instant blocking probability $B_u^C(i,i+1)$ is not, in general, equal to the steady-state blocking probability $B_u(i,i+1)$.

The first step of the decomposition method is to characterize the upstream and downstream servers. The processing time distribution of each machine can be characterized by exponential distributions or phase type distributions (including Coxian distributions as a special case). For a given characterization, the problem is to determine the parameters of the upstream and downstream machines of all subsystems. First, we note that Machine $M_u(1,2)$ represents the portion of the line upstream of Buffer $B_{1,2}$, which consists of only Machine M_1. Therefore, Machine $M_u(1,2)$ must be identical to Machine M_1. By a similar argument, Machine $M_d(K-1,K)$ must be identical to Machine M_K. Thus, we have the following boundary conditions:

$$f_u(1,2) = f_1 \text{ and } f_d(K-1,K) = f_K \tag{61}$$

In order to determine the remaining $2(K-2)$ machines, three major sets of equations can be used. The first set of equations is related to the service process of the downstream machines of the two-machine lines. Consider, for instance, Machine $M_d(i,i+1)$ in Line $L(i,i+1)$. Since Machine $M_d(i,i+1)$ represents the portion of line L downstream of Buffer $B_{i,i+1}$, the processing time of Machine $M_d(i,i+1)$ represents the time between the instant of beginning of a processing of a part on Machine M_{i+1} and the instant of the transfer of the part into Buffer $B_{i+1,i+2}$. This time is composed of the processing time followed by a blocking time if Buffer $B_{i+1,i+2}$ is full at the instant of processing completion. The blocking of Machine M_{i+1} in Line L is represented by

the blocking of Machine $M_u(i+1,i+2)$ in Line $L(i+1,i+2)$. Thus, the probability that Machine M_{i+1} is blocked at instant of completion of a part is approximated by the completion-instant blocking probability $B_u^C(i+1,i+2)$ of Machine $M_u(i+1,i+2)$ in Line $L(i+1,i+2)$, and the blocking time is approximated by the residual service time of Machine $M_d(i+1,i+2)$ at the instant at which blocking occurs. Consequently, with probability $1-B_u^C(i+1,i+2)$, there is a zero blocking time, and with probability $B_u^C(i+1,i+2)$, there is a non-zero blocking time whose PDF is $g_d^C(i+1,i+2)$. Therefore, the PDF of the processing time of Machine $M_d(i,i+1)$ can be expressed as

$$f_d(i,i+1) \;=\; f_{i+1} \;*\; \Big((1-B_u^C(i+1,i+2))\;0_f \;+\; B_u^C(i+1,i+2)\;g_d^C(i+1,i+2)\Big)\,, \qquad i=1,\cdots,K-2 \quad (62)$$

where $*$ denotes the convolution operator and 0_f denotes the PDF of the distribution of the random variable that is equal to 0 with probability 1. This equation will be referred to as the **blocking propagation** equation.

A similar set of equations is related to the service process of the upstream machines of the two-machine lines. Consider Machine $M_u(i,i+1)$ in Line $L(i,i+1)$. Since Machine $M_u(i,i+1)$ represents the portion of line L upstream of Buffer $B_{i,i+1}$, the processing time of Machine $M_u(i,i+1)$ represents the time between the instant of transfer of a part into Buffer $B_{i,i+1}$ and the instant of the processing completion of the next part on Machine M_i. This time is composed of a starvation time, if Buffer $B_{i-1,i}$ is empty at the instant of transfer, followed by a processing time. A starvation of Machine M_i in Line L is represented by a starvation of Machine $M_d(i-1,i)$ in Line $L(i-1,i)$. Thus, the probability that Machine M_i is starved at instant of transfer of a part is approximated by the completion-instant starvation probability $S_d^C(i-1,i)$ of Machine $M_u(i-1,i)$ in Line $L(i-1,i)$, and the starvation time is approximated by the residual service time of Machine $M_u(i-1,i)$ at the instant at which starvation occurs. Consequently, with probability $1-S_d^C(i-1,i)$, there is a zero starvation time, and with probability $S_d^C(i-1,i)$, there is a non-zero starvation time whose PDF is $g_u^C(i-1,i)$. Therefore, the PDF of the processing time of Machine $M_u(i,i+1)$ can be expressed as:

$$f_u(i,i+1) \;=\Big(1-S_d^C(i-1,i))\;0_f \;+\; S_d^C(i-1,i)\;g_u^C(i-1,i)\Big) \;*\; f_i\;, \qquad i=2,\cdots,K-1 \quad (63)$$

This equation will be referred to as the **starvation propagation** equation.

If we had considered blocking-before-service, the form of equation (62) would have been slightly different. This is because, in the case of BBS, blocking occurs before the processing starts whereas in the case of BAS, it occurs at the end of the processing. Actually, the form of equation (62) would have been similar to that of equation (63). This is simply because of the duality of starvation and blocking-before-service discussed in Section 3.6.

The third set of equations is related to the departure process from Machine M_i. A departure from Machine M_i occurs at the instant at which a part is transferred from Buffer $B_{i-1,i}$ into Buffer $B_{i,i+1}$. The departure process from Machine M_i in line L corresponds to the departure

process of Machine $M_d(i-1,i)$ in Line $L(i-1,i)$ and also to the departure process of Machine $M_u(i,i+1)$ in Line $L(i,i+1)$. Consequently, these two processes should be identical. Let $h_d(i-1,i)$ and $h_u(i,i+1)$ denote the PDFs of the interdeparture times from Machine $M_d(i-1,i)$ and Machine $M_u(i,i+1)$, respectively. Thus, we have:

$$h_d(i-1,i) = h_u(i,i+1)\ ,\ i = 2,\cdots,K-1 \tag{64}$$

This equation will be referred to as the **departure process** equation.

The above equations are sometime used in simpler forms that involve only the means of the distributions. This is because when an exponential characterization is used for certain machines, its distribution is totally determined by a single parameter (the mean). Consider first the case where the downstream machines are characterized by exponential distributions. Then, using the memoryless property of the exponential distributions, it is easy to show that equation (62) implies:

$$T_d(i,i+1) \ = \ T_{i+1} \ + \ B_u^C(i+1,i+2)\ T_d(i+1,i+2)\ ,\ i = 1,\cdots,K-2 \tag{65}$$

Similarly in the case where the upstream machines are characterized by exponential distributions, equation (63) implies:

$$T_u(i,i+1) \ = \ S_d^C(i-1,i)\ T_u(i-1,i) \ + \ T_i\ ,\ i = 2,\cdots,K-1 \tag{66}$$

(Recall that T_i, $T_d(i,i+1)$, and $T_u(i,i+1)$ are the average processing times of Machines M_i, $M_d(i,i+1)$, and $M_u(i,i+1)$, respectively.)

Consider equation (64). By just considering the means of the distributions, this equation implies that the average interdeparture time from Machines $M_d(i-1,i)$ is equal to the average interdeparture time from $M_u(i,i+1)$. The average interdeparture time from a machine is equal to the inverse of its production rate. Thus, if $P_d(i-1,i)$ and $P_u(i,i+1)$ denote the production rates of Machines $M_d(i-1,i)$ and $M_u(i,i+1)$, respectively, we have:

$$P_d(i-1,i) = P_u(i,i+1)\ ,\ i = 2,\cdots,K-1 \tag{67}$$

Because of the conservation of flow through each two-machine line (equation (13)), i.e. $P_d(i,i+1) = P_u(i,i+1) = P(i,i+1)$, this equation can simply be written as:

$$P(i-1,i) = P(i,i+1)\ ,\ i = 2,\cdots,K-1 \tag{68}$$

Equation (68) means that all subsystems should have the same production rate. Thus, any approximate method that uses this equation will be consistent with the basic conservation of flow relationship, equation (13).

We note that equation (68) holds for any characterization of both the upstream and the downstream machines, whereas equation (65) (resp. (66)) holds provided that the distributions of downstream (resp. upstream) machines are exponential. Equations (65), (66), and (68) will be referred to as the simpler version of equations (62), (63), and (64), respectively.

These equations (or their simpler versions) can be combined to determine the distributions of the processing times of the upstream and downstream machines. As shown by Dallery and Frein (1989a), there are three main approaches. A first set of equations, called $SE1$, is obtained

by combining equations (62) and (64). A second set of equations, referred to as $SE2$, is obtained by combining equations (63) and (64). Finally, a third set of equations, $SE3$, is obtained by combining equations (62) and (63).

It is important to notice that (1) System $SE2$ is the dual of System $SE1$ in the sense that $SE1$ uses the blocking propagation equation while System $SE2$ uses the starvation propagation equation; (2) any solution of $SE1$ (or $SE2$) satisfies conservation of flow since equation (64) implies equation (68); (3) System $SE3$ offers a symmetrical view of the decomposition since both the blocking and the starvation propagation equations are involved. It was proved by Dallery and Frein (1989a) that these three systems of equations are equivalent. A consequence of this result is that any solution of System $SE3$ will satisfy conservation of flow, although this equation is not explicitly involved in $SE3$.

To solve any of these three systems of equations, an iterative procedure must be used. The procedure must be appropriate for the system of equations that is chosen. There are three major types of algorithms, referred to as Algorithms $A1$, $A2$, and $A3$, which are associated with Systems $SE1$, $SE2$, and $SE3$, respectively. However, slightly different algorithms can be used for a given system of equations. (See Dallery and Frein (1989a), for more details).

An approach of type $SE1$ was used, among others, by Altiok (1982), Perros and Altiok (1986), Pollock, Birge, and Alden (1985), and Takahashi, Miyahara, and Hasegawa (1980), for flow lines with exponential processing times. In all these papers the subsystems are FSSQs with lost arrivals and exponential interarrival times. As a result, they are equivalent to a two-machine line decomposition with exponential characterization of the upstream machines. (See Section 4.2.1.) The parameter of the exponential distribution of the upstream machines is determined by using conservation of flow, equation (68).

Pollock, Birge, and Alden (1985), and Takahashi, Miyahara, and Hasegawa (1980) consider an exponential characterization for the downstream machine as well. In Pollock, Birge, and Alden (1985), the parameter of each exponential distribution is determined by using an equation that is equivalent to the simpler form of equation (62), that is, equation (65). Takahashi, Miyahara, and Hasegawa (1980) also use equation (65). However, they use the steady-state blocking probabilities instead of the completion-instant blocking probabilities. Also, although they are considering BAS, they do not increase the capacity of the FSSQ by 1.

Perros and Altiok (1986) assume that the downstream machines are characterized by phase-type distributions. The phase-type distribution of each downstream machine is obtained by using equation (62). This work is actually an extension of an earlier work of Altiok (1982) where it was assumed that a machine may only be blocked by its immediate successor. Altiok (1989) extended the method of Perros and Altiok (1986) to the case of flow lines with phase-type processing time distributions. For the same class of systems, Jun and Perros (1987) decompose the system into a set of FSSQs with lost arrivals and phase-type interarrival and service times. Although there is no exact equivalence in that case (because the lost model differs from the switch-off model), their approach is similar to an approach of type $SE1$.

Hillier and Boling (1967) analyzed flow lines with exponential processing times using an approach of type $SE2$. They decompose the original line into a set of FSSQs with exponential interarrival and service times, which is equivalent to a two-machine line decomposition with exponential characterizations for both machines. The parameter of the exponential distribution of each downstream server is obtained by using conservation of flow, equation (68). The parameter

of the exponential distribution of each upstream server is obtained by using an equation which differs significantly from equation (66). This equation is a reduced service rate approximation and does not seem to be extendible to more general cases, e.g., phase-type characterizations.

An approach of type $SE3$ was independently proposed by Altiok and Ranjan (1987) and Gun (1987); see also Gun and Makowski (1989). Both the upstream and the downstream machines of each two-machine line is characterized by phase-type distributions. The distributions of the upstream and downstream machines are obtained by means of equations (63) and (62), respectively. Algorithm $A3$ designed to solve System $SE3$ consists of successive forward and backward passes. During each forward (resp. backward) pass, the estimates of the parameters of the phase-type distributions of the upstream (resp. downstream) machines are improved. Because the number of phases of the distributions of upstream and downstream machines may be very large, Altiok and Ranjan (1987) suggest approximating the distributions of starvation and blocking times that appear in equations (63) and (62) by Coxian-2 distributions. (See Appendix.)

This approach is attractive since it offers a symmetrical view of the decomposition in the sense that starvation and blocking play a similar role. Let us illustrate how this symmetrical decomposition method works by means of a simple example. Consider a three-machine flow line where the processing time distribution of each machine is assumed to be a Coxian-2 distribution. Let $(\mu_{i,1}, a_{i,1}, \mu_{i,2})$ be the parameters of the Coxian-2 distribution corresponding to Machine M_i, $i = 1, 2, 3$. From the boundary conditions (equation (61)), we know that the distributions of Machines $M_u(1,2)$ and $M_d(2,3)$ are identical to those of Machines M_1 and M_3, respectively.

The distribution of Machine $M_d(1,2)$ is determined using (62), the blocking propagation equation. According to this equation, the phase-type distribution of Machine $M_d(1,2)$ has the form shown in Figure 5. In the case of phase-type distributions, the residual processing time distribution $g_d^C(2,3)$ is characterized by the parameters $\alpha_j^C(2,3)$, $j = 1, 2$, where $\alpha_j^C(2,3)$ is the probability that Machine $M_d(2,3)$ is in phase j at the instant at which blocking of Machine $M_u(2,3)$ occurs. Thus, there are three parameters that characterize the phase-type distributions of Machine $M_d(1,2)$, namely $B_u^C(2,3)$, $\alpha_1^C(2,3)$, and $\alpha_2^C(2,3)$, which can be derived from the exact analysis of Line $L(2,3)$. Note that in Figure 5, the transition probabilities of the phase-type distribution are not expressed explicitly, but they are easily obtained. For instance, the transition probability $c_{1,4}$ from the first stage to the fourth is given by $c_{1,4} = (1 - a_{2,1})B_u^C(2,3)\alpha_2^C(2,3)$.

Similarly, the distribution of Machine $M_u(2,3)$ is determined from (63), the starvation propagation equation. According to this equation, the phase-type distribution of Machine $M_u(2,3)$ has the form shown in Figure 6. The distribution $g_u^C(1,2)$ is characterized by the parameters $\beta_j^C(1,2)$, $j = 1, 2$, where $\beta_j^C(1,2)$ is the probability that Machine $M_u(1,2)$ is in phase j at the instant at which starvation of Machine $M_d(1,2)$ occurs. Thus, there are three parameters that characterize the phase-type distributions of Machine $M_u(2,3)$, namely $S_d^C(1,2)$, $\beta_1^C(1,2)$, and $\beta_2^C(1,2)$, which can be derived from the exact analysis of Line $L(1,2)$.

Very few theoretical results are available about the existence or uniqueness of the solution or about the convergence of the iterative algorithm associated with a particular decomposition. In case of an exponential characterization of both the upstream and downstream machines, Dallery and Frein (1989a) proved that the three systems of equations, $SE1$, $SE2$, $SE3$, have a unique solution, and that Algorithm $A3$ associated with System $SE3$ always converges.

All the above methods are based on the decomposition of a K-machine line into $K - 1$

Figure 5: Phase-Type Distribution of Machine $M_d(1, 2)$

Figure 6: Phase-Type Distribution of Machine $M_u(2, 3)$

two-machine lines. Alternative decompositions that involve larger subsystems are also possible. For instance, a K-machine line can be decomposed into a set of $K-2$ three-machine lines. This approach was investigated by Brandwajn and Jow (1988). It may provide more accurate results. However, it requires repetitive solutions of three-machine subsystems that, as discussed in Section 4.4, are usually complex.

5.1.2 Other Results

Several authors have derived simple approximate formulas for estimating the production rate of a FLRM where all stations are identical (that is, their processing time distributions have the same mean and coefficient of variation) and all buffers have the same capacity. A formula was obtained by Knott (1967, 1970a) in the case of two-machine flow lines with identical Erlang distributions. Haydon (1973) (cited by Buzacott, 1990) extended Knott's formula to flow lines with any number of machines. A formula was obtained by Muth (1987) (cited by Brumenfield, 1990) in the case of flow lines with any number of machines and no intermediate storage. Brumenfield (1990) extended Muth's formula to flow lines with intermediate buffers.

Recently, there have been several attempts to establish bounds for the production rate of FLRMs. Van Dijk and Lamond (1988) considered a model which is equivalent to a three-machine FLRM. All processing times are exponentially distributed. The type of blocking considered is blocking-before-service. Lower and upper bounds on the production rate are obtained by considering two different modifications of the original model. For instance, the lower bound model is the same as the original model except that there is no limitation on the capacity of each buffer but there is a global limitation on the total number of parts that can be present in the first and second buffers. This model has an exact product-form solution (Baskett et al., 1975). Shanthikumar and Jafari (1987) considered the same model. They suggested alternative lower and upper bounds. These bounds are obtained as the production rates of two different two-machine lines.

Onvural and Perros (1989) derived an upper bound on the production rate of a flow line with exponential service times. The idea is based on using an equivalent closed model. In the case of BBS, for instance, the production rate of the original flow line is identical to the production rate of a closed model whose population is equal to the total number of buffer spaces of the original model. An upper bound on the production rate of the closed model is obtained by replacing all finite buffers by infinite buffers. The production rate of the resulting closed model can simply be calculated since this model has a product-form solution (Baskett et al., 1975).

5.2 Flow Lines with Unreliable Machines

5.2.1 Decomposition

All decomposition methods for flow lines with unreliable machines decompose the original line with K machines into a set of $K-1$ two-machine (FLUM) lines, each one being associated with a buffer of the original line. The basic principle of decomposition methods presented in Section 5.1.1 in the case of FLRMs remains valid in the case of FLUMs. Here again, the upstream machine of Line $L(i,i+1)$, Machine $M_u(i,i+1)$ represents the effects of the entire portion of the line upstream of Buffer $B_{i,i+1}$, as illustrated in Figure 4. Similarly, $M_d(i,i+1))$, the downstream

machine of Line $L(i, i+1)$, represents the effects of the entire portion of the line downstream of Buffer $B_{i,i+1}$.

However, existing decomposition methods for FLRMs and FLUMs are somewhat different, even though they are based on similar principles. This is because decomposition methods for FLUMs are designed mainly to capture the effect of failures of machines. As a consequence, they are based on the assumption that starvation and blocking in FLUMS are due to failures of machines rather than variations in processing times.

Decomposition methods developed so far assume that the uptime and downtime distributions of all machines of the original flow line are memoryless, i.e., exponential for asynchronous or continuous models and geometric for synchronous models. In the case of asynchronous model the processing time distributions is also assumed to be exponential.

In all decomposition methods proposed for FLUMs that we are aware of, the characterization of the upstream and downstream machines of all two-machine FLUMs is the same as that of the machines of the original line. For instance, if the machines of the original line have exponential distributions for processing times, uptimes, and downtimes, the upstream and downstream machines of all the two-machine lines have exponential distribution for processing times, uptimes, and downtimes as well. As a result, they are also characterized by three parameters. Exact solutions of the subsystems can be obtained using an appropriate technique among those discussed in Section 4.3.

Evidence for the exponential/geometric distribution comes from the observations of Vladzievskii (1952), who studied the output processes of actual lines, and found that they were closely fitted to exponential distributions; of Buzacott (1967a), who calculated the variance of the output processes of different models, and found that the coefficient of variation was always close to 1; and of Schick (Schick and Gershwin, 1978) who studied the output processes of simulations of Buzacott-type two-machine lines, and found that both uptimes and downtimes had distributions that appeared to be very close to geometric.

As in the case of FLRMs, several equations can be used to determine the unknown parameters of the upstream and downstream machines of each two-machine line. Since all distributions are exponential (or geometric), they are determined by a single parameter. Therefore, one needs as many equations as the number of unknown distributions. Typical decomposition methods for FLUMs use some or all of the following:

Conservation of Flow One of the basic principles of decomposition, as indicated in Section 5.1.1, is that the behavior of material in the buffer of Line $L(i, i+1)$ is close to the behavior of material in Buffer $B_{i,i+1}$ in Line L. In particular, the average flow rate of material out of the buffer of Line $L(i, i+1)$ is close to the average flow rate of material out of Buffer $B_{i,i+1}$ in Line L, that is, $P(i, i+1) \approx P_{i+1}$. As a result of the conservation of flow in Line L, equation (13), this implies the following conservation of flow between the two-machine lines:

$$P(i-1, i) = P(i, i+1) \ , \ i = 2, \cdots, K-1 \tag{69}$$

Note that this equation is identical to equation (68) for FLRMs. This is not surprising since conservation of flow is not related to whether the machines are reliable or unreliable.

Flow Rate-Idle Time A second set of equations is based on (10) for systems with operation dependent failures. It can be written

$$\frac{1}{\mu_u(i,i+1)e_u(i,i+1)} = \frac{1}{\mu_i e_i} + \frac{1}{P(i-1,i)} - \frac{1}{\mu_d(i-1,i)e_d(i-1,i)} \tag{70}$$

It may also be expressed in the following alternative form:

$$\frac{1}{\mu_d(i-1,i)e_d(i-1,i)} = \frac{1}{\mu_i e_i} + \frac{1}{P(i,i+1)} - \frac{1}{\mu_u(i,i+1)e_u(i,i+1)} \tag{71}$$

These equations are obtained from equation (10) by assuming that the probability of Machine M_i being idle can be expressed as the sum of the probability of it being starved and the probability of it being blocked, i.e.:

$$I_i = S_i + B_i \tag{72}$$

In the case of discrete material systems (asynchronous or synchronous models), this relationship is only approximate because a machine can be simultaneously starved and blocked. In fact, $I_i < S_i + B_i$. If the buffers are large, the probability of being starved and blocked simultaneously is very small and therefore equation (72) is a good approximation. For continuous models, (72) is exact since in that case, a machine cannot be blocked and starved simultaneously (Dallery, David, and Xie, 1989).

The probability of starvation of Machine M_i in Line L is approximated by the probability of starvation of Machine $M_d(i-1,i)$ in Line $L(i-1,i)$, i.e., $S_i = S_d(i-1,i)$. Similarly, the probability of blocking of Machine M_i in Line L is approximated by the probability of blocking of Machine $M_u(i,i+1)$ in Line $L(i,i+1)$, i.e., $B_i = B_u(i,i+1)$. Therefore, using equation (72), equation (10) implies:

$$E_i = e_i(1 - S_d(i-1,i) - B_u(i,i+1)) \tag{73}$$

Equation (10) also holds for Machine $M_d(i-1,i)$ in Line $L(i-1,i)$. This machine is never blocked, so that the equation reduces to:

$$E_d(i-1,i) = e_d(i-1,i)(1 - S_d(i-1,i)) \tag{74}$$

Similarly, equation (10) applied to Machine $M_u(i,i+1)$ yields:

$$E_u(i,i+1) = e_u(i,i+1)(1 - B_u(i,i+1)) \tag{75}$$

Incorporating equations (74) and (75) into equation (73) yields:

$$\frac{E_i}{e_i} + 1 = \frac{E_d(i-1,i)}{e_d(i-1,i)} + \frac{E_u(i,i+1)}{e_u(i,i+1)} \tag{76}$$

Equations (70) and (71) are then obtained from equation (76) by using the following relationships: $E_i = P_i/\mu_i$; $E_d(i-1,i) = P(i-1,i)/\mu_d(i-1,i)$; $E_u(i,i+1) = P(i,i+1)/\mu_u(i,i+1)$; and $P_i = P(i-1,i) = P(i,i+1)$.

We note that equations similar to (70) and (71) can also be obtained for systems with time dependent failures, using equation (12) instead of equation (10).

Resumption of Flow These equations propagate the distribution of repair times in pseudo-machines. That is, if Machine $M_u(i, i+1)$ is down, they determine the probability distribution of when it will be up. They determine the mean time to repair of Machine $M_u(i, i+1)$, $MTTR_u(i, i+1)$, or alternatively its repair rate (or probability of repair in the case of a synchronous system), $r_u(i, i+1)$. These quantities are functions of corresponding parameters of $M_u(i-1, i)$, among other things.

These equations are based on the following arguments. Machine $M_u(i, i+1)$ represents the portion of the line upstream of Buffer $B_{i,i+1}$. Thus, Machine $M_u(i, i+1)$ is down if either Machine M_i is down, or Machine M_i is starved as a result of one of its upstream machines being down. The second condition is equivalently represented by Machine $M_u(i-1, i)$ being down. The time when $M_u(i, i+1)$ comes back up is therefore related to the time when $M_u(i-1, i)$ would come back up (if it were the culprit) and to the time when M_i would come back up. Similarly, repair parameters of $M_d(i, i+1)$ are related to those of $M_d(i+1, i+2)$.

When probabilities or rates are propagated, these equations are in the form:

$$r_u(i, i+1) = r_u(i-1, i)X(i, i+1) + r_i(1 - X(i, i+1)) \tag{77}$$

where $X(i, i+1)$ is the conditional probability, given that $M_u(i, i+1)$ is down (i.e., that there is no material entering $B_{i,i+1}$), that $B_{i-1,i}$ is empty, and $M_u(i-1, i)$ is down. Similarly,

$$r_d(i, i+1) = r_d(i-1, i)Y(i, i+1) + r_{i+1}(1 - Y(i, i+1)) \tag{78}$$

Alternatively, when MTTR's are propagated the resumption of flow equations are of the form:

$$\frac{1}{r_u(i, i+1)} = \frac{1}{r_u(i-1, i)}\gamma(i, i+1) + \frac{1}{r_i}(1 - \gamma(i, i+1)) \tag{79}$$

where $\gamma(i, i+1)$ is the fraction of failures of Machine $M_u(i, i+1)$ that are caused by a failure of Machine $M_u(i-1, i)$. Similarly,

$$\frac{1}{r_d(i, i+1)} = \frac{1}{r_d(i-1, i)}\beta(i, i+1) + \frac{1}{r_{i+1}}(1 - \beta(i, i+1)) \tag{80}$$

Recall that pseudo-machines have been characterized as having exponentially (or geometrically) distributed repair times. It appears from the above discussion that the repair time of Machine $M_u(i, i+1)$ corresponds to a repair time of either Machine M_i, or one of the machines further upstream, M_{i-1}, or M_{i-2}, ..., or M_1. Since all these machines have exponential repair times, the repair time of Machine $M_u(i, i+1)$ should be a mixture of these exponentials, a hyperexponential distribution (Kleinrock, 1975), which is a special case of phase-type distribution. Consequently, a more accurate characterization of the repair time of Machine $M_u(i, i+1)$ would be a hyperexponential distribution. Assuming an exponential characterization of repair times implies that the hyperexponential distribution is approximated by an exponential distribution having the same mean.

When all the machines of the original line have the same repair rate, the hyperexponential distribution reduces to an exponential distribution. In other words, all the repair times of the pseudo-machines are exponentially distributed with the same rate as the common repair rate of

all the machines of the original line. More generally, if the repair rates of the machines of the original line are close to one another, the hyperexponential distributions are close to exponential distributions. We therefore expect that methods based on these equations are more accurate when the repair parameters are more nearly equal.

A potentially interesting research area would be to develop approximation techniques based on hyperexponential distributions. It would be important to avoid adding unnecessary complexity to improve accuracy.

Interruption of Flow These equations propagate the distribution of failure times in pseudo-machines. That is, if Machine $M_u(i, i+1)$ is up, they determine the probability distribution of when it will next be down. They determine the mean time to failure of Machine $M_u(i, i+1)$, $MTTF_u(i, i+1)$, or alternatively its failure rate (or probability of failure in the case of a synchronous system), $p_u(i, i+1)$. These quantities are functions of parameters of $M_u(i-1, i)$, among other things.

These equations are based on the following arguments. If $M_u(i, i+1)$ is up, it may fail due to M_i going down or due to Buffer $B_{i-1,i}$ becoming empty as a result of one of the upstream machines of M_i being down. This second condition is equivalently represented by the buffer of Line $L(i-1, i)$ becoming empty as a result of a failure of $M_u(i-1, i)$. The time when $M_u(i, i+1)$ goes down is therefore related to the time when $M_u(i-1, i)$ would go down and to the time that M_i would go down. Similarly, failure parameters of $M_d(i, i+1)$ are related to those of $M_d(i+1, i+2)$.

These equations are often of the form

$$p_u(i, i+1) = p_i + V(i, i+1)S_d(i-1, i) \tag{81}$$

and

$$p_d(i, i+1) = p_{i+1} + W(i, i+1)B_u(i+1, i+2) \tag{82}$$

where $V(i, i+1)$ and $W(i, i+1)$ are appropriate parameters. Similarly, they could instead propagate MTTF, so they would be written

$$\frac{1}{p_u(i, i+1)} = \frac{1}{p_i} + \eta(i, i+1)\frac{1}{p_s(i-1, i)} \tag{83}$$

and

$$\frac{1}{p_d(i, i+1)} = \frac{1}{p_{i+1}} + \delta(i, i+1)\frac{1}{p_b(i+1, i+2)} \tag{84}$$

where $\eta(i, i+1)$ and $\delta(i, i+1)$ are appropriate parameters.

Boundary Conditions Since the first machine is never starved, $M_u(1, 2)$ is chosen to be the same as M_1. Since the last machine is never blocked, $M_d(K-1, K)$ is chosen to be the same as M_K. This leads to a set of equations which may be summarized as

$$\text{all parameters of } M_u(1, 2) = \text{all parameters of } M_1 \tag{85}$$

$$\text{all parameters of } M_d(K-1,K) = \text{all parameters of } M_K \tag{86}$$

We now describe the different decomposition methods that have been proposed in the literature. The number of equations used by each method depends on the number of free variables in the system. For a given number of free variables, the methods differ by the precise equations that are used and/or the algorithm used to solve the set of equations. We note that all methods use equations (85) and (86).

One-Parameter Machine Sevast'yanov (1962) extended his continuous material, two-machine solution to develop an approximate decomposition technique for a long line. He assumed that the line consisted of machines that operate at the same speed and have the same exponential repair time distributions. The failure time distributions are also exponential but need not be the same. Consequently, we classify it as a one-parameter machine method because there is only one parameter that distinguishes each machine from the others in the line. For each buffer, approximately equivalent upstream and downstream machines are defined. These machines represent the effects of the entire upstream portion of the line and the entire downstream portion of the line on the buffer. They are assumed to have the speed and repair rates as the machines in the long line. Equations like (81) and (82) for the approximate values of the failure rates of the equivalent machines are developed. (The equations are constructed by making use of "anti-articles," which is the same concept used in developing duality of flow lines. See Section 3.6.) Sevast'yanov (1962) demonstrated uniqueness of the solution to his set of equations.

Zimmern (1956) earlier discussed such an approach in an informal way. Sevast'yanov was evidently unaware of this work. Buzacott (1967a) also discussed a similar method for three-stage and longer lines.

Two-Parameter Machine Gershwin (1987a) developed a decomposition method for a general Buzacott-type model of synchronous transfer lines. Since the machines may have different failure and repair parameters but the same operating speed, it is a two-parameter machine method. It makes use of (69), (70), (77), (78), (85), and (86). This paper also offered an algorithm for solving these equations.

Numerical results (1) were close to simulation results, both previously published and performed for that paper; (2) suggested that production rates decreased to a non-zero limit as the lengths of lines of identical machines and buffers increase without bound; (3) confirmed earlier work on reversibility and duality; (4) showed that the addition of a nearly perfectly reliable machine to an unreliable line affects performance in a predictable way; (5) demonstrated that buffer levels, in lines of identical machines and buffers, are high near the upstream end of the line, are nearly constant throughout the middle of the line, and are low near the downstream end of the line; (6) indicated that the production rate of a line with buffer space divided into many small buffers is greater than the production rate when the same space is divided into a few large buffers. In addition, the numerical experiments provided evidence that the method was efficient, since it was applied to lines with 20 machines.

Dallery, David, and Xie (1988) developed an algorithm (called the **DDX algorithm**) for Gershwin's (1987a) decomposition equations which was a great improvement over Gershwin's

algorithm. They repeated some examples, and obtained identical results in less time. Their numerical results showed a reduction of computer time by a factor of 4 to 10, with the greatest improvement in longer lines. They also analyzed a line with 40 machines. Finally, the DDX algorithm converged for examples on which the earlier algorithm failed. The set of equations used in the DDX algorithm consists of equations (70), (71), (77), and (78). The algorithm is in the form of an iterated downstream and upstream sweep. After initialization, the downstream sweep consists of (77) and (70). The upstream sweep consists of (78) and (71). Boundary conditions are again determined by (85) and (86).

Dallery, David, and Xie (1988) used equation (71) instead of (69). The advantage of this is to offer a symmetrical view of the decomposition method, in which upstream and downstream machines play a similar role. This symmetrical set of equations naturally leads to the simple back and forth iterative procedure described above. The new set of equations is equivalent to that used by Gershwin (1987a). As a result, conservation of flow, equation (69), although not explicitly used, is satisfied when the algorithm converges. We note that the DDX method is conceptually similar to the decomposition methods for FLRMs that use an approach of type $SE3$. (See Section 5.1.1.)

Dallery, David, and Xie (1989) derived a similar decomposition as Gershwin's (1987a) for continuous lines with equal processing rates, which they term **homogeneous** lines. Their equations were the same as those of Dallery, David, and Xie (1988): (70), (71), (77), and (78). As noted above, the Flow Rate-Idle Time equations ((70) and (71)) are now exact because, in a continuous material line, a machine cannot be starved and blocked simultaneously. They also applied the DDX algorithm to this case. Also, equations (77) and (78) were derived from equations (79) and (80). Dallery, David, and Xie (1989) implicitly showed that equations (77) (which propagates the repair rates) and (79) (which propagates the MTTR's) are equivalent. A similar statement can be made for equations (78) and (80). They concluded from their numerical and simulation experiments that the method "provides very good estimates of the production rates, and fairly good estimates of the average buffer levels". Moreover, they emphasized that this decomposition method is consistent with the duality property of continuous flow lines. (See Section 3.6.) This again results from using symmetrical equations with respect to upstream and downstream machines.

Glassey and Hong (1986b) develop essentially the same method as Gershwin's (1987a) for continuous lines with equal processing rates. They compared their results numerically with those of Gershwin (1987a) (for synchronous systems) and found close agreement. They do not indicate how they solved their equations, but they seem to use Gershwin's algorithm. They find that their version converges faster.

We note that all the above papers used the resumption of flow equations but not the interruption of flow equations. Thus, one question that naturally arises is what would be obtained by using the latter instead of the former. It is possible to show that there is an equivalence between these two sets of equations. Indeed, in the case of two-parameter machine (synchronous or continuous model), it is possible to show that equation (81), for instance, can be obtained by combining equations (70) and (77) by using the fact that the processing rates of all pseudo-machines are equal. However, informal numerical experimentation suggests that algorithms that use the interruption of flow equations may have poor convergence properties.

Xie (1990) proposed a simplified version of the decomposition method of Dallery, David,

and Xie (1989) for continuous lines with equal processing rates. The simplification is to set the repair rates of all pseudo-machines equal to the repair rates of the corresponding machines in the original line. That is, equations (77) and (78) are replaced by $r_u(i, i+1) = r_i$ and $r_d(i, i+1) = r_{i+1}$. The advantage of this simplified method is that Xie (1990) was able to prove the existence and uniqueness of the solution of the set of equations and the convergence of a simplified version of the DDX algorithm. However, this method can be expected to be less accurate than the method of Dallery, David, and Xie (1989) except when the repair rates of the machines of the original line are equal. Indeed, in the case where all the machines of the original line have the same repair rates, it is easy to check that the algorithm proposed by Xie (1990) and the DDX algorithm are identical. Thus, the results of Xie (1990) provides a proof of the existence and uniqueness of the solution and of the convergence of the DDX algorithm (Dallery, David, and Xie, 1989) in the special case where all the machines of the original line have the same repair rate.

Recently, a slightly different decomposition method was proposed by Liu and Buzacott (1989). As in Gershwin (1987a), they consider a general Buzacott-type model of synchronous transfer lines. The general principle is again to decompose the original line into a set of two-machine lines. Each equivalent machine is of the same type as the machines of the original line. Liu and Buzacott (1989) use the boundary conditions, equations (85) and (86). They also use equations similar to (70) and (71). The other equations are based on second-moment equivalence between equivalent machines corresponding to the same machine of the original line. They require that the second moment of the inter-output process of Machine $M_u(i, i+1)$ be equal to the second moment of the inter-input process of Machine $M_d(i-1, i)$. Thus, the method of Liu and Buzacott (1989) mainly differs from that of Gershwin (1987a) and Dallery, David and Xie (1988) in imposing second moment conditions instead of using equations (77) and (78). We note that the idea of using two-moment information has similarities with equation (64) used in certain decomposition methods for FLRMs.

In the derivation of their method, Liu and Buzacott (1989) introduce the concept of zero-buffer equivalence. Consider a two-machine flow line with a finite intermediate buffer. It is possible to define two different two-machine lines with no intermediate buffer that are equivalent to the original line with respect to the departure times one with the original first machine and an equivalent second machine and the other with the original second machine and an equivalent first machine.

Transformation of Three-Parameter Machine into Two-Parameter Machine A first way of handling lines with three-parameter machine is to proceed as follows. In a first step, the original line with non-identical processing times is transformed into a line with identical processing times using one of the transformations presented in Section 3.8.3. In a second step, the resulting homogeneous line is analyzed using the above decomposition methods. As a result, this approach involves two stages of approximation.

This was used by Gershwin (1987b) to extend his decomposition method (Gershwin, 1987a) to lines with different processing times. He used disaggregation (Section 3.8.3) as the transformation step. Dallery, David, and Xie (1989) pointed out, however, that these disaggregation and decomposition methods are not well suited to each other. The reason is that the parameters, especially the repair rates, of the equivalent machines of the resulting homogeneous line differ

by several orders of magnitude. As stated earlier, decomposition methods cannot be expected to be very accurate in that case.

Dallery, David, and Xie (1989) (Section 3.8.3) used the homogenization transformation coupled with the decomposition method. They found that the results are good when the original line is nearly homogeneous. If this is not the case, significant errors may be encountered since homogenization is no longer an accurate transformation. Similar results are obtained if the transformation proposed by Liu and Buzacott (1990) (Section 3.8.3) is used instead of homogenization.

Three-Parameter-Machine New difficulties arise when machine speeds (μ_i) differ from one another. This is because pseudo-machines are assumed to be like real machines in having a speed that does not change over time. On the other hand, the arrival and departure processes from buffers have different speeds at different times. They change over time as a result of buffers becoming empty or full. If Machine M_i is not starved or blocked or down, the rate that material flows into Buffer $B_{i,i+1}$ is μ_i. However, if Buffer $B_{i-1,i}$ is empty because the speed of Machine M_{i-1}, μ_{i-1}, is less than μ_i, then the rate that material flows into Buffer $B_{i,i+1}$ is μ_{i-1}. Thus, it would be more accurate to represent pseudo-machine $M_u(i, i+1)$ as having a time-varying speed.

More generally, the rate that material flows into a buffer may have many different values, depending on the relative speeds of its upstream machines, and the rate that material departs from a buffer may also change over time, depending on the distribution of speeds of machines downstream. If these speeds are not very different, the decomposition methods may work well. If they are, we should not expect great accuracy.

Choong and Gershwin (1987) extended Gershwin's (1987a) decomposition equations to asynchronous systems in which all machines could have different speeds, failure rates, and repair rates (that is, a long version of Gershwin and Berman's two-machine line). All the distributions of processing times, uptimes, and downtimes are assumed to be exponential. In addition to the equations used by Gershwin (1987a), Choong and Gershwin (1987) used the Interruption of Flow equations, that is (81) and (82). They used an algorithm similar to Gershwin's (1987a). While the method worked for some problems, it seemed to diverge for many others. Sastry (1985) performed a similar analysis using equations like (79), (80), (83), (84). See also Awate and Sastry (1987).

Gershwin (1989) transformed the set of equations derived by Choong and Gershwin (1987) into an equivalent set of equations in a similar way as Dallery, David, and Xie (1988) did for Gershwin's method (1987a). Indeed, the set of equations used by Gershwin (1989) consists of equations (70), (71), (77), (78), (81) and (82). Again, this approach is attractive since it offers a symmetrical view of the decomposition. Gershwin (1989) used an extension of the DDX algorithm to solve this set of equations. As in the case of two-parameter machines, he found that the new algorithm was substantially faster and more reliable. Sastry's (1985) earlier algorithm also sweeps up and down the line. It differs in that it uses the conservation of flow equations to update $\mu_d(i, i+1)$, and flow rate-idle time to update $\mu_u(i, i+1)$. It appears to require, as an iterated step, the determination of a parameter so that a two-machine line has a given production rate. This is the *inversion* of the exact methods described in Section 4. Such an inversion caused the method of Gershwin (1987a) to be slow, and the method of Choong and

Gershwin (1987) to be slow and unstable.

Hong and Seong (1989) proposed a slightly different decomposition method for asynchronous exponential lines. They set the processing time of each pseudo-machines equal to that of the corresponding machine in the original line. That is, they choose $\mu_u(i,i+1) = \mu_i$ and $\mu_d(i,i+1) = \mu_{i+1}$. Again, as in the case of two-parameter machines, it is possible to show that it is equivalent to use the Resumption of Flow equations or the Interruption of Flow equations.

Glassey and Hong (1986b) extended their two-parameter method to continuous lines with different processing rates. They derived a set of equations which is identical to that used by Choong and Gershwin (1987). That is, they added the Interruption of Flow equations (81) and (82). They do not indicate how they solve their set of equations. Also, it should be noted that their continuous model is the same as that of their two-machine line study (Glassey and Hong, 1986a). That is, they do not reduce the failure rate of a slowed down machine.

Semery (1987 and 1988) also considered continuous lines with different processing times. His model is the same as that of Gershwin and Schick (1980) and Dubois and Forestier (1982). He extended the decomposition method of Dallery, David, and Xie (1989). As in Hong and Seong (1989), he set the processing time of each pseudo-machine equal to that of the corresponding machine in the original line. His set of equations is again symmetrical and he used an extension of the DDX algorithm to solve the set of equations.

5.2.2 Aggregation

Several authors have independently developed approximation methods, which they refer to as **aggregation methods** (Ancelin and Semery, 1987; De Koster, 1987; Terracol and David, 1987a). The basic idea of aggregation is to replace a two-machine-one-buffer sub-line by a single equivalent machine. Most often, the equivalent machine is of the same type as the machines of the original line. Thus, an aggregation method for analyzing a line with K machines consists in applying $K-1$ single aggregation steps. The aggregation of machines can be performed in any order.

Analysis of continuous flow lines using aggregation was proposed by Ancelin and Semery (1987) and Terracol and David (1987a) in the case of operation-dependent failures and De Koster (1987) in the case of time-dependent failures. The uptimes and downtimes are exponentially distributed. The machines may have different processing rates. Thus, the machines of the original line as well as the equivalent machines have three parameters. For illustration purpose, consider the case where the aggregation is performed in the order of the machines. In that case, the aggregation method works as follows.

The first sub-line to be analyzed consists of Machines M_1 and M_2 and Buffer $B_{1,2}$. This sub-line, $L_{1,2}$, is then replaced by a single equivalent machine, M_2^a, which represents it in an aggregate way with respect to the rest of the line. Consider the flow of material out of Machine M_2 in sub-line $L_{1,2}$. The average uptime of Machine M_2 is equal to the average time during which the rate of flow out of Machine M_2 is positive in sub-line $L_{1,2}$. The average downtime of Machine M_2^a is equal to the average time during which the rate of flow out of Machine M_2 is zero in sub-line $L_{1,2}$. This corresponds to a situation where Machine M_2 is either down or starved as a consequence of Machine M_1 being down. Finally, the processing rate of Machine M_2^a is equal to the average rate of flow out of Machine M_2 in sub-line $L_{1,2}$. It is a weighted

average of the processing rates of Machines M_1 and M_2. Thus, Machine M_2^a represents in a aggregate way sub-line $L_{1,2}$. The second sub-line to be analyzed consists of Machines M_2^a and M_3 and Buffer $B_{2,3}$. This sub-line is in turn aggregated into a single equivalent machine, M_3^a. This aggregation procedure is repeated until the last machine of the line. The last step consists of aggregating the sub-line consisting of Machines M_{K-1}^a and M_K and Buffer $B_{K-1,K}$ into a single equivalent machine, Machine M_K^a.

It appears that the aggregation method can also be viewed as a decomposition of the original line into $K-1$ two-machine sub-lines. Machine M_i^a, $i = 2, ...K-1$, is the upstream machine of sub-line $L(i, i+1)$, that is Machine $M_u(i, i+1)$. However, the major difference between aggregation and decomposition methods is that in the case of the aggregation method, the parameters of Machine M_i^a do not depend on the portion of the line downstream of Machine M_i, while in the case of the decomposition method they do. In other words, the parameters of Machine M_i^a are calculated as if Machine M_i was never blocked. Another way of looking at the aggregation method is to view it as a simplified decomposition method in which the downstream machine of each sub-line is the same as the corresponding machine of the original line, i.e., $M_d(i, i+1) = M_{i+1}$, $i = 1, ..., K-1$. As a result, applying the aggregation method is equivalent to applying a single forward step of the DDX algorithm. In light of this, it is easy to find examples for which the aggregation method is not accurate. To overcome this problem, different improvements of the aggregation method have been proposed by Terracol and David (1987b) and De Koster (1988b). However, the analysis becomes much more complex.

Jafari and Shanthikumar (1987b) proposed an approximation method for synchronous flow lines with operation-dependent failures and possible scrapping of parts. Their method can be viewed as a refinement of the aggregation method. In order to present it, it is useful to view it as a decomposition method. Consider the decomposition of the original line into $K-1$ two-machine subsystems. The method involves two slightly different analyses for each subsystem. In the first case, Machine $M_d(i, i+1)$ is identical to Machine M_{i+1}, as in the original aggregation method. The corresponding subsystem will be denoted by $L^a(i, i+1)$. In the second case, Machine $M_d(i, i+1)$ also models the blocking effect of Machine M_{i+2} on Machine M_{i+1}. The corresponding subsystem will be denoted by $L^b(i, i+1)$. The method consists of a $K-2$ steps. The aim of the i'th step is to determine the parameters of Machines $M_d(i, i+1)$ in Subsystem $L^b(i, i+1)$ and the parameters of Machine $M_u(i+1, i+2)$ in Subsystem $L^a(i+1, i+2)$. In order to do this, step i consists of an iterative procedure that alternatively analyzes Subsystem $L^b(i, i+1)$ and Subsystem $L^a(i+1, i+2)$.

We note that this method can again be viewed as a simplified decomposition method in the sense that the downstream machine of Subsystem $L^b(i, i+1)$ does not model the entire portion of the line downstream of Buffer B_i. Indeed, the effect of failures of Machines $M_{i+3}, M_{i+4}, \cdots, M_K$ on the behavior of material through Buffer B_i is neglected. (A failure of one of these machines can indeed cause a blocking of Machine M_{i+1}.)

Finally, we note that the method of Jafari and Shanthikumar (1987b) also differs from the other approximation methods for FLUMs (either aggregation or decomposition methods) by the fact that the characterization of the upstream and downstream machines of each subsystem is different from that of the machines of the original line. In the model of Jafari and Shanthikumar (1987b), Machine $M_u(i, i+1)$ being down is represented by two different states depending on whether it is down as a result of Machine M_i being down or as a result of Machine M_i being

starved. Similarly Machine $M_d(i, i+1)$ being down (in Subsystem $L^b(i, i+1)$) is represented by two different states depending on whether it is down as a result of Machine M_{i+1} being down or as a result of Machine M_{i+1} being blocked.

It is interesting to note that using such a more complex characterization of subsystems could also be done in decomposition methods. It is expected that this would improve their accuracy, especially in cases where the repair rates of the machines of the original line are very different. We note however that it would significantly increase the computational complexity of decomposition methods. Nevertheless, it may be worth investigating what would be the effect of such a modification on, for instance, the Gershwin-DDX decomposition method.

6 Extensions

In this section, we briefly review some flow line papers that relax the more typical assumptions of Section 2.4, or that study issues other than steady state production rate and mean buffer levels. We also briefly survey the literature pertaining to assembly and closed loop systems.

6.1 Variance of Output

All the methods described in Sections 4 and 5 deal with steady-state average production rates and steady-state average buffer levels. However, the *variance* of the production and of the buffer levels during a time period is also important.

This issue has been entirely neglected. As far as we are aware, there are only two published papers that deal with the calculation of the variance of the behavior of a transfer line over a limited time period: Miltenburg (1987) and Lavenberg (1975). (There is also an unpublished paper by Ou and Gershwin (1989).) These papers only treat two-machine lines, and obtain results that are difficult to use and to understand intuitively. It is not clear how to extend the results of these papers either exactly or approximately.

This is a tremendously important area because manufacturers must deliver products on a daily or weekly basis. Informal numerical and simulation experimentation (Gershwin, 1991b), as well as factory observation, indicate that the *standard deviation* of weekly production can be over 10% of the mean. This implies that, over the course of a year, it is not surprising to see that the production of some weeks can be half that of other weeks. This variability is an inherent characteristic of these systems. It is striking that it is so little appreciated by researchers in this area. Perhaps it is of greater importance now than in the past because of the current emphasis on "just-in-time" production.

Prediction of this variability is no less important than that of the prediction of the mean; in fact, it may be more important. It would be reasonable for the buyer of a transfer line not only to specify its mean production rate; in addition, or instead, the firm might insist that it be able to deliver a certain amount of finished product each week with a certain probability. In order to do this, the research community must be able to provide statistical information on the number of parts produced during a given time interval other than just the mean: ideally, the probability distribution; more realistically, it can provide the standard deviation of the number of parts produced during a time interval, and the probability can be estimated based on, for instance, a normal distribution.

Furthermore, the variance of the number of parts in buffers, or in the whole line, may be just as important. Indeed, the time spent by a part to go through the line is highly influenced by the number of parts currently in the line. Since manufacturers must respond quickly to customers, a variable response time is just as undesirable as a variable production rate.

6.2 Models with Scrapping of Parts

Scrapping refers to the rejection of bad parts. When a part is rejected, it leaves the line and does not go into the machines and buffers downstream of the point where the rejection occurs. Conservation of flow must be generalized to account for scrapping.

Shanthikumar and Tien (1983) analyzed a synchronous two-machine FLUM with geometric distributions for up- and downtimes in which scrapping of workpieces occurs with a certain probability when a failure occurs; see Section 2.1.3. They presented an algorithmic solution of the transition equations based on the matrix geometric property; see Section 4.3.1. Jafari and Shanthikumar (1987a) considered the more general case where the distributions of uptimes and downtimes are discrete phase-type distributions. They also proposed an aggregation method for the approximate analysis of flow lines with any number of machines; see Section 5.2.2.

6.3 Machines in Parallel

Ignall and Silver (1977) developed an approximation for two-stage systems with multiple identical machines in each stage. The approximation is based on Buzacott's (1967b) observation that the production rate of a two-machine line (i.e., a two-stage system with a single machine in each stage) can be written (in our notation)

$$E(N) = E_0 + (E_\infty - E_0)m(N)$$

where $m(N)$ is a monotonically increasing function of the buffer size, and E_0 and E_∞ are the production rates with no buffers and with infinite buffers. Expressions for E_0 and E_∞ are given in Sections 3.5 and 4.1. To extend this formula for multiple machines per stage, E_0 and E_∞ are adjusted. (They evidently had synchronous systems in mind, but this equation applies to asynchronous and continuous systems as well.)

Elsayed and Hwang (1984) considered a two-stage synchronous system in which each stage consists of two machines in parallel. The machines may be operated in a splitting arrangement, in which the two machines at a stage are operated at half the required production rate when they are both operational. When one fails, the other operates at the production rate. Alternatively, they may be operated in a standby arrangement, in which the backup machine at a stage is only operated when the primary machine goes down. The backup machine may be in a cold, warm, or hot standby status. There is a different failure rate associated with each status. Presumably, it takes least time to switch over to a hot standby machine, and most time to switch over to a cold standby machine, but this is not stated clearly. Thus, insurance against disruption due to failures is provided not only by the buffers (as in all the other models), but also by multiple machines. Numerical solutions are provided, and the production rate follows the familiar saturating pattern.

Forestier (1980) formulated a generalizization of the Zimmern-Wijngaard-Gershwin-Schick continuous model. He replaced the single machines before and after the buffer with banks of machine operating in parallel. Thus α_i became an integer, rather than a binary variable. The differential equations and boundary conditions for the steady-state probability distribution are generalizations of Zimmern's and Gershwin and Schick's.

Mitra (1988) studied essentially the same system and obtained a solution by formulating the problem as an eigenvalue problem. He demonstrated numerically that several small machines at each station are better than a few large machines (when the production rates of an isolated station composed of the small machines is the same as that of an isolated station composed of the large machines).

Finally, Iyama and Ito (1987) analyzed a line with parallel machines and exponential processing times by solving the underlying Markov chain.

6.4 Limited Repair Personnel

Dudick (1979) studied synchronous systems in which there are limited repair personnel. He considered a set of strategies which assign priorities to machines under a variety of conditions. For example, one class of strategies is to always repair Machine i first whenever both machines are down. Another is to always repair Machine i first whenever both machines are down and the buffer level is above or below some threshold. In all graphs provided, the production rate is a saturating function of the buffer size. Buzacott (1982) extended some of Dudick's results.

Elsayed and Turley (1980) assumed that each machine has two modes of failure, and that there are a limited number of repair personnel. In one policy, the machines are treated the same when they are both down, but in the other policies, one machine is favored over another if it is in one failure mode and the other is in the other mode. They solved the Markov transition equations numerically and found, for the examples they considered, that (1) production rate is a saturating function of buffer size; (2) the policy of treating all failures equally is superior to favoring one over the other; (3) the difference between the latter policies only appears as buffer size is sufficiently large. It is not stated explicitly, but this is evidently a synchronous line. They make the same assumption as Buzacott (1967b): that the probability of two events occurring in a cycle is negligible. This is evidently a blocking-after-service model since the first machine may fail while the buffer is full.

6.5 Non-Zero Transfer Time In Buffers

Commault and Semery (1990) pointed out that for some systems, the time that it takes a part to move through a buffer, even when the buffer is empty, may be significant. In that case, the models covered in this paper are inadequate, since they assume that the transit time is 0. Commault and Semery used a two-machine line with non-zero buffer transit time to demonstrate the effects of buffer delays. They propose an approximation in which a two-machine line whose buffer is smaller than that of the original line, but which has no delays, is nearly equivalent to the original line. This issue is also investigated by Liu (1990).

6.6 Line Design

The line design problem is that of choosing buffer sizes or machine parameters to maximize performance or minimize cost subject to constraints. In order to solve this problem, it is necessary to evaluate the performance measures of lines. Existing methods ignore variability since there are essentially no methods for evaluating variability.

Sevast'yanov (1962) studied the optimal allocation of storage space for his one-parameter system. Hillier and Boling (1966, 1967, 1972, 1977, 1979) studied the allocation of machine capacity and storage space and discovered the "bowl phenomenon" and made related observations. (However, see Sections 1.4 and 3.9.)

Other analytic papers that deal with line design include Sheskin (1974, 1976), Soyster and Toof (1976), and Soyster, Schmidt, and Rohrer (1979). Coillard and Proth (1984) study a system similar to that of Dubois and Forestier (1982) and use the analysis to find the optimal location of a single buffer among a line of several machines. This is similar to the problem considered by Soyster and Toof (1976).

Ho, Eyler, Chien (1978) and Caramanis (1987) optimize line performance by using perturbation analysis techniques to calculate gradients from simulations. In fact, Ho, Eyler, Chien (1978) was the paper in which perturbation analysis was developed.

6.7 Assembly/Disassembly (Fork/Join) Networks

Throughout the paper, we have been concerned with flow line models. The special structure of these models, *i.e.,* a series of machines separated by buffers, allowed many results to be obtained. Although flow line structures are often encountered in industry, there also exist manufacturing systems exhibiting more general structures. Among these, two are of great interest and can be viewed as extensions of the flow line structure, namely **assembly** systems and **closed loop** systems.

An assembly system is a manufacturing system in which some machines perform assembly operations. There are two kinds of assembly systems (Liu, 1990): (1) those that add components to a workpiece (such as printed circuit or surface mount assembly), and (2) those that assemble different entities (workpieces) that have themselves already been processed within the manufacturing system. With respect to modeling and analysis, the first case is not different from a flow line. The second, which can form a network, is what we deal with here.

A closed loop system is a flow line in which resources such as pallets are required. There is a finite set of resources available. To be loaded into the system, a part must first get a resource. It then holds this resource during its sojourn in the system. When the part leaves the system, the resource is again available and a new part may enter the system. Resources may also correspond to a control policy, for example where they correspond to kanbans; see e.g. Di Mascolo, Frein, Dallery, and David (1990). In a kanban policy, only a limited number of parts are allowed in some portion of the system. Each part is associated with one of a limited number of kanbans (cards).

Flow lines, assembly, and closed loop structures are special cases of a general structure referred to as **assembly/disassembly** (A/D) or **fork/join** (F/J) networks. **Assembly** is the process of creating a single entity out of more than one. **Disassembly** is the reverse: it is the

process of creating more than one entity out of one. The terms **fork** and **join** are also often used for disassembly and assembly, respectively. An A/D system consists of a set of A/D machines interconnected by a set of buffers such that each buffer has exactly one upstream server and one downstream machine. An A/D machine has a set of input buffers and a set of output buffers. An A/D machine pulls one entity from each of its upstream buffers and delivers one entity to each of its downstream buffers.

Many of the results presented in Section 3 for flow line models have extensions in the case of A/D models. First, the results pertaining to a single machine (see Section 3.2) easily extend to a single A/D machine. The sample path behavior of A/D networks can still be described by means of evolution equations that are generalizations of those presented in Section 3.3; see Dallery, Liu, and Towsley (1990, 1991). As a result, properties like conservation of flow, monotonicity, reversibility, can again be established using these evolution equations; see Ammar (1980), Adan and Van der Wal (1989), Shanthikumar and Yao (1989), Liu (1990), and Dallery, Liu, and Towsley (1990 and 1991). Duality properties do also exist for general A/D networks; see Ammar and Gershwin (1989), Dallery, Liu, and Towsley (1990) and Liu (1990). Another property of interest is that of symmetry, which can be obtained by combining reversibility and duality properties (Dallery and Towsley, 1990; Dallery, Liu, and Towsley, 1990).

As for flow line systems, analysis techniques for A/D networks are mainly based on approximations. Present approximation methods are limited to two kinds of systems: tree-structured A/D networks and closed loop systems. A **tree structured** A/D system is an A/D system whose associated graph does not contain cycles. To the best of our knowledge, all approximation methods for tree structured A/D systems pertain to systems with unreliable machines. These methods are extensions of the decomposition and aggregation methods presented in Section 5.2. As in flow lines, all decomposition methods for tree structured A/D networks decompose the original system into a set of two-machine, one-buffer systems (that is, two-machine flow lines). Details can be found in Gershwin (1986a, b, 1991a), Di Mascolo, David, and Dallery (1991), Liu (1990), De Koster (1987, 1988a)

Closed loop systems are more complex to analyze than flow line systems because of the population constraint imposed by the closed loop structure, *i.e.,* the total number of entities in the different buffers is a constant. This quantity corresponds, for instance, to the total number of pallets available. Approximation methods for closed loop systems with reliable machines have been proposed by Suri and Diehl (1984, 1986), Akyildiz (1988), Onvural and Perros (1987), Dallery and Frein (1989b), Frein and Dallery (1989), Liu (1990). See also Onvural (1990) for a survey of these results. An approximation method for closed loop systems with unreliable machines has been developed by Frein, Commault, and Dallery (1991).

7 Conclusions and New Directions for Research

In this paper, we have tried to be as exhaustive as possible in describing the most important, and most widely studied, issues and problems in manufacturing flow lines. The literature pertaining to the class of models considered in Section 2 is fairly complete. Certain issues, however, require further investigations.

As described in Section 5, several approximation methods for FLRMs and FLUMs have been

developed. Although some numerical experiments have been reported in the literature, it would be useful to have a systematic study of the accuracy of the different methods. Moreover, since FLUMs can be transformed into FLRMs and vice versa (see Section 3.7), it would be useful to compare the accuracy of the methods devoted to FLRMs to that of the methods devoted to FLUMs when applied to either FLRMS or FLUMs. We suggest that these comparisons be performed using data chosen according to the guidelines in Section 3.10. Also, approximation methods for FLUMs that can handle more general distributions of times to failures and times to repair than exponential (or geometric) ones, should be developed. Another possible research direction is to apply the theory of Markov chain aggregation to the analysis of these systems. See Schweitzer and Altiok (1989).

The issues discussed in Section 6 have not yet been completely resolved, although partial results are available. One missing area is that of variability. As we describe in Section 6, there are only two published papers that deal with the question of the variance of the output of a transfer line. This class of problems should be high on the research agenda of the field. Approximation methods for flow lines with parallel machines should also be developed. The results pertaining to A/D networks are not as complete as those pertaining to flow lines. This area needs more investigation. In particular, approximation methods that can handle a larger class of A/D networks than those currently available should be developed.

Other issues include multiple part types, routing, control, correlated failures, and systems with machines of different types. An example of the latter is a system with batched continuous material, so the upstream portion of the line is continuous and the downstream portion is synchronous. Another example is a line consisting of some automated stations and some manual stations. A portion of the line might be synchronous while the rest is asynchronous.

There have been some recent papers on pull systems and kanban-operated lines. This area, and the general issue of just-in-time production, is very close to the literature surveyed here, and will prove to be important in the near future.

On the other hand, there is already enough literature on allocation of storage or machine capacity, particularly based on optimization methods. We say this because there seems to be very little difference in performance between intuitively reasonable and optimal allocations (much less than the error in estimating machine parameters). It would however be of interest to develop simple rules of thumb for resource allocation, and to prove bounds on their distance from optimality.

Acknowledgment We are grateful for the helpful comments of the editors, Professor John A. Buzacott and Professor J. George Shanthikumar, and of two anonymous reviewers. This work was partly supported by the National Science Foundation under Grant DDM-8914277.

A Appendix

A.1 Coxian and Phase-Type Distributions

In general, the only tractable models are those that can be described by Markov processes, either continuous time or discrete time (Kleinrock, 1975). Here we restrict our attention to continuous time Markov processes (CTMPs). CTMPs are naturally obtained when all the distributions in

the original model are **exponential distributions**. This is due to the famous and important **memoryless property** of the exponential distribution (Kleinrock, 1975). The probability density function (pdf) of an exponential distribution is $f(t) = \mu e^{-\mu t}$. The exponential distribution is characterized by a single parameter, μ, called the rate. Its mean is $m = 1/\mu$ and its coefficient of variation (CV) is 1.

Because it naturally gives rise to Markov processes, the exponential distribution has been widely used in the literature. However, it is not always an appropriate candidate for representing actual distributions of real systems. In particular, distributions encountered in real systems may have coefficients of variation far from 1. Fortunately, it is possible to overcome this difficulty, while keeping the tractability of Markov processes, by using the so-called **method of stages**. The idea is to represent a non-exponential distribution as a mixture of exponential distributions.

The simplest distribution of this form is the so-called **Erlang distribution**. An Erlang distribution consists of a series of s exponential distributions with common rate μ. The random variable associated with the Erlang distribution is then the sum of s independent exponential random variables with rate μ.

A more general distribution is the so-called **Coxian distribution** (Kleinrock, 1975). A Coxian distribution with s **stages**, also termed **phases**, is represented in Figure 7. The Coxian distribution is more general than the Erlang distribution since it allows non-identical rates and branching probabilities. In order to understand the meaning of this distribution, it is convenient to give it a physical interpretation. Suppose it represents the overall processing time of a task that can be decomposed into a set of s subtasks. The processing time of subtask j is exponentially distributed with rate μ_j. Subtask 1 is first performed. Upon completion of subtask 1, either subtask 2 is performed, with probability a_1, or the overall task is completed, with probability $b_1 = 1 - a_1$. More generally, upon completion of subtask j, either subtask $j + 1$ is performed, with probability a_j, or the overall task is completed, with probability $b_j = 1 - a_j$. Note that $b_s = 1$, that is, at most s subtasks are performed. Finally, we note that in the most general form of Coxian distribution, it is possible to have a zero processing time with non-zero probability (Kleinrock, 1975). This is achieved by adding a branching probability (a_0, b_0) before stage 1.

The most general form of distributions that are mixtures of exponential distributions is the so-called **phase-type distribution** (Neuts, 1981). A phase-type distribution with s stages (or phases) is represented in Figure 8. One can again give a physical interpretation of this distribution in terms of an overall task that is decomposed into a set of s exponential subtasks. (The processing time of subtask j is exponentially distributed with rate μ_j.) In that case, the first subtask to be processed is subtask j with probability $c_{0,j}$. Upon completion of subtask j, either subtask k is performed, with probability $c_{j,k}$, or the overall task is completed, with probability $c_{j,0}$. The branching and transition probabilities satisfy $c_{0,1} + ... + c_{0,s} = 1$, and $c_{j,1} + ... + c_{j,s} + c_{j,0} = 1$. Again, one may add the possibility of having a zero processing time with non-zero probability. A Coxian distribution is a special case of phase-type distributions.

Remark. We note that phase-type distributions can alternatively be described as the absorption time of a continuous time Markov process with s transient states and a single absorbing state (Neuts, 1981). A state of the CTMP is associated with each of the s stages, and the extra state, state 0, is the absorbing state. The parameters of the distribution are now the initial state probabilities and the transition rates of the CTMP. The initial state probabilities correspond to the branching probabilities $c_{0,j}$. The transitions rates from state j to state k, $\mu_{j,k}$, are given by

Figure 7: Coxian Distribution with s Phases

Figure 8: Phase-Type Distribution with s Phases

$\mu_{j,k} = \mu_j c_{j,k}$, for $k = 0, 1, ..., s$. It should be emphasized that the two descriptions are equivalent. However, in order to be consistent when dealing with Coxian and phase-type distributions, we choose to only use the first description.

Coxian and phase-type distributions give rise to Markovian processes by extending the original state space to incorporate the detailed information of which stage each distribution is currently in. This increase of the size of the state space is the price to pay to deal with models involving non-exponential distributions. As the feasibility and complexity of numerical solutions of Markov processes are very much dependent on the size of the state space, the number of stages of phase-type distributions should be kept as small as possible. (See Section 4.)

The practical question is then how to get (exact or approximate) phase type representations of non-exponential distributions. There are two ways of obtaining phase-type distributions: either by **construction** or by **identification**. The first way corresponds to the case where the stages of the phase-type distribution have a physical interpretation. For instance, suppose a machine performs an operation that takes an exponential time with rate 3.0. Moreover, at the end of the processing, some parts stay on the machine for an inspection phase that takes an exponential time with rate 4.0. Only one tenth of the parts are inspected and these parts are selected at random. Then, the overall processing time (including inspection) of the machine is exactly represented as a Coxian-2 distribution with parameters $\mu_1 = 3.0; a_1 = 0.1; \mu_2 = 4.0$.

The second way is to identify a phase-type distribution to a given distribution. In that case, the stages of the resulting phase-type distribution have no physical interpretation. They are often called **fictitious stages**. Except in a limited number of cases (see Section 3.7 for an example), it is usually not possible to identify the whole distribution. Instead, it is easy to determine a phase-type distribution that has the same first and second moments, that is the same mean and coefficient of variation, as those of a given distribution (Marie, 1980; Pierrat 1987). Identifying the first two moments is usually a good enough approximation, except for distributions with large coefficients of variation in which case significant improvements may be obtained by also identifying the third moment (Altiok, 1985a; Pierrat, 1987).

Finally, we note that similar results can be obtained in the case of discrete time Markov processes. In that case, the geometric distribution plays the same role as the exponential distribution for CTMPs. Discrete time Coxian and phase-type distributions can be defined in a similar way.

A.2 Uptime-Downtime Relationship

Property 1 *Consider any machine of a flow line with unreliable machines. Let $MTTF$ and $MTTR$ denote its average time to failure and average time to repair, respectively. Let E, I, and D denote the proportions of time during which the machine is working, idle, and down, respectively. Then, these quantities are related by*

$$\frac{E}{D} = \frac{MTTF}{MTTR} \tag{87}$$

if the machine has operation-dependent failures, and by

$$\frac{E+I}{D} = \frac{MTTF}{MTTR} \tag{88}$$

Figure 9: Illustration of the Proof

if the machine has time-dependent failures.

Proof. The machine alternates periods during which it is up (up-periods) and periods during which it is down (down-periods). Let t_k be the time at which the k-th down-period ends and the $(k+1)$-th up period begins. During each up-period, the machine alternates periods of time during which it is working and periods of time during which it is idle. This is illustrated in Figure 9. Let us define the following quantities

δ_k^U: length of the k-th up-period.

δ_k^D: length of the k-th down-period.

δ_k^W: total working time during the k-th up-period.

δ_k^I: total idle time during the k-th up-period.

Note that δ_k^I may be equal to zero. The quantities E, I, and D can then be expressed as

$$E = \frac{E[\delta_k^W]}{E[t_k - t_{k-1}]} \; ; \; I = \frac{E[\delta_k^I]}{E[t_k - t_{k-1}]} \; ; \; D = \frac{E[\delta_k^D]}{E[t_k - t_{k-1}]} \tag{89}$$

Note that $\delta_k^W + \delta_k^I = \delta_k^U$ and as a result, $E[\delta_k^U] = E + I$. Then, the proof follows by noticing that $E[\delta_k^D] = MTTR$; $E[\delta_k^W] = MTTF$, in the case of operation-dependent failures; $E[\delta_k^U] = MTTF$, in the case of time-dependent failures. □

We note that the above proof does not require any assumptions on the distributions of processing times, up-times, and down-times.

B References

Adan, I. and Wal, J. (1989), "Monotonicity of the Throughput in Single Server Production and Assembly Networks with Respect to the Buffer Sizes," In Queueing Networks with Blocking, H. G. Perros and T. Altiok (eds.), North-Holland, Amsterdam, pp. 435- 356.

Akyildiz, I. F. (1988), "On the Exact and Approximate Throughput Analysis of Closed Queueing Networks with Blocking," IEEE Trans. on Soft. Eng., Vol. 14, pp. 62-70.

Altiok, T. (1982), "Approximate Analysis of Exponential Tandem Queues with Blocking," European Journal of Operations Research, Vol. 11, 1982.

Altiok, T. (1985a), "On the Phase-Type Approximations of Generals Distributions," IIE Transactions, Vol. 17, pp. 110-116.

Altiok, T. (1985b), "Production Lines with Phase-Type Operation and Repair Times and Finite Buffers," Int. J. Prod. Res., Vol. 23, pp. 489-498.

Altiok, T. (1989), "Approximate Analysis of Queues in Series with Phase-type Service Times and Blocking," Operations Research, Vol. 37, pp. 601-610.

Altiok, T. and Ranjan, R. (1987), "Analysis of Production Lines with General Service Times and Finite Buffers: A Two-Node Decomposition Approach, Tech. Rep., Department of Industrial and System Eng., Rutgers University, Piscataway.

Altiok, T. and Stidham, S., Jr. (1982), "A Note on Transfer Lines with Unreliable Machines, Random processing Times, and Finite Buffers," IIE Transactions, Vol. 14, No. 2, pp. 125-127.

Altiok, T. and Stidham, S., Jr. (1983), "The Allocation of Interstage Buffer Capacities in Production Lines," IIE Transactions, Vol. 15, pp. 292-299.

Alvarez, R., Dallery, Y. and David, R. (1991), "An Experimental Study of the Continuous Flow Model of Transfer Lines with Unreliable Machines and Finite Buffers," IMACS Int. Symp. on Modelling and Control of Technological Systems, Lille, May 1991.

Ammar, M. H. (1980), "Modelling and Analysis of Unreliable Manufacturing Assembly Networks with Finite Storages," MIT Laboratory for Information and Decision Systems Report LIDS-TH-1004.

Ammar, M. H. and Gershwin, S. B. (1989), "Equivalence Relations in Queuing Models of Fork/Join Queueing Networks with Blocking," Performance Evaluation, Vol. 10, pp. 233-245.

Ancelin, B. and Semery, S. (1987), "Calcul de la Productivité d'une Ligne Intégrée de Fabrication: CALIF, un Logiciel Industriel Basé sur une Nouvelle Heuristique," APII, Vol. 21, pp. 209-238.

Anderson, D. R. (1968), "Transient and Steady-State Minimum Cost In-Process Inventory Capacities for Production Lines," Ph. D. Thesis, Department of Industrial Engineering, Purdue University.

Artamonov, G. T. (1976), "Productivity of a Two-Instrument Discrete Processing Line in the Presence of Failures," Kibernetika, Vol. 3, pp. 126-130 (In Russian). (Eng. tr. Cybernetics, Vol. 12, pp. 464-468 (1977)).

Avi-Itzhak, B. (1965), "A Sequence of Service Stations with Arbitrary I and Regular Service Times," Manag. Sci., Vol. 11, pp. 565-571.

Awate, P. G. and Sastry, B. L. N. (1987), "Analysis and decomposition of Transfer and Flow Lines," Opsearch, Vol. 24, No. 3, pp. 175-196.

Baccelli, F. (1989), "Ergodic Theory of Stochastic Petri Networks," Research Report, INRIA No 1037, May 1989, to appear in Annals of Probability.

Baker, K. R., Powell, S. G., and Pyke, D. F. (1989), "Buffered and Unbuffered Assembly Systems with Variable Processing Times," Working Paper No. 246, Amos Tuck School of Business Administration, Dartmouth College, October, 1989.

Baskett, F., Chandy, K. M., Muntz, R. R., and Palacios, F. G. (1975), "Open, Closed and Mixed Networks of Queues with Different Classes of Customers," J. ACM, Vol. 22, No. 2, pp. 248-260.

Berman, O. (1982), "Efficiency and Production Rate of a Transfer Line with Two Machines and a Finite Storage Buffer," Euro. J. of Oper. Res., Vol. 9, pp. 295-308.

Bobbio, A. and Trivedi, K. S. (1988), "Computation of the Completion Time when the Work Requirement is a PH Random Variable," Int. Conf. on Analysis and Control of Large Scale Stochastic Systems, Chapel Hill, NC, May 1988.

Boxma, O. J. (1977), "Analysis of Models for Tandem Queues," Doctoral thesis, University of Utrecht.

Brandwajn, A., and Jow, Y-L. L. (1988), "An Approximation Method for Tandem Queues with Blocking," Operations Research, Vol. 36, pp. 73-83.

Brumenfield, D. E. (1990), "A Simple Formula for Estimating Throughput of Serial Production Lines with Variable Processing Times and Limited Buffer Capacity," Int. Jour. of Prod. Res., Vol. 28, No. 6, pp. 1163-1182.

Buxey, G. M., Slack, N. D. and Wild, R. (1973), "Production Flow Line System Design — A Review," AIIE Trans., Vol. 5, No. 1, pp. 37-48.

Buzacott, J. A. (1967a), "Markov Chain Analysis of Automatic Transfer Line with Buffer Stock," Ph. D. Thesis, Department of Engineering Production, University of Birmingham.

Buzacott, J. A. (1967b), "Automatic Transfer Lines with Buffer Stocks," Int. J. Prod. Res., Vol. 5, No. 3, pp. 182-200.

Buzacott, J. A. (1968), "Prediction of the Efficiency of Production Systems without Internal Storage," Int. J. Prod. Res., Vol. 6, No. 3, pp. 173-188.

Buzacott, J. A. (1969), "Methods of Reliability Analysis of Production Systems Subject to Breakdowns," in D. Grouchko ed., Operations Research and Reliability, Proc. of a NATO Conf. (Turin, Italy, June 24 -July 4, 1969), pp. 211-232.

Buzacott, J.A. (1972), "The Effect of Station Breakdowns and Random Processing Times on the Capacity of Flow Lines," AIIE Transactions, Vol. 4, pp. 308-312.

Buzacott, J. A. (1982), " 'Optimal' Operating Rules for Automatic Manufacturing Systems,", IEEE Trans. Automatic Control, Vol. AC- 27, pp. 80-86.

Buzacott, J. A. (1990), "Abandoning the Moving Assembly Line: Models of Human Operators and Job Sequencing," Int. J. Prod. Res., Vol. 28, No. 5, pp. 821-839.

Buzacott, J. A. and Hanifin, L. E. (1978a), "Models of Automatic Transfer Lines with Inventory Banks — A Review and Comparison," IIE Transactions, Vol. 10, pp. 197-207.

Buzacott, J. A. and Hanifin, L. E. (1978b), "Transfer Line Design and Analysis — An Overview," 1978 Fall IE Conference, pp. 277-286.

Buzacott, J. A. and Kostelski, D. (1987), "Matrix-geometric and Recursive Algorithm Solution of a Two-stage Unreliable Flow Line," IIE Transactions, Vol. 19, pp. 429-438.

Caramanis, M. (1987), "Production System Design: A Discrete Event Dynamic System and Generalized Benders' Decomposition Approach," International Journal of Production Research, Vol. 25, No. 8, pp. 1223-1234, 1987.

Choong, Y. F. and Gershwin, S. B. (1987), "A Decomposition Method for the Approximate Evaluation of Capacitated Transfer Lines With Unreliable Machines and Random Processing Times," IIE Transactions, Vol. 19, pp. 150-159.

Cohen, J. W. (1982), The Single Server Queue, North Holland, Amsterdam.

Coillard, P. and Proth, J.-M. (1984), "Sur L'Effet des Stocks Tampons dans une Fabrication en Ligne," Revue Belge de Statistique, d'Informatique et de Recherche Operationelle, Volume 24, pp. 3-27, 1984.

Commault, C. and Dallery, Y. (1990), "Production Rate of Transfer Lines with No Intermediate Storage, IIE Transactions, Vol. 22, No. 4.

Commault, C. and Semery, A. (1990), "Taking into Account Delays in Buffers for Analytical Performance Evaluation of Transfer Lines," IIE Transactions, Vol. 22, No. 2, pp. 133-142.

Dallery, Y., David., R., and Xie, X.-L. (1988), "An Efficient Algorithm for Analysis of Transfer Lines with Unreliable Machines and Finite Buffers," IIE Transactions, Vol. 20, pp. 280-283.

Dallery, Y., David, R., and Xie, X.-L. (1989), "Approximate Analysis of Transfer Lines with Unreliable Machines and Finite Buffers," IEEE Transactions on Automatic Control, Vol. 34, pp. 943-953.

Dallery, Y., and Frein, Y. (1989a), "On Decomposition Methods for Tandem Queueing Networks with Blocking," Technical Report MASI, Université Pierre et Marie Curie, Paris, July. To appear in Operations Research.

Dallery, Y., and Frein, Y. (1989b), "A Decomposition Method for Approximate Analysis of Closed Queueing Networks with Blocking, In Queueing Networks with Blocking, H. G. Perros and T. Altiok (eds.), North-Holland, Amsterdam, pp. 193-216.

Dallery, Y., Liu, Z., and Towsley, D. (1990), "Equivalence, Reversibility, and Symmetry Properties in Fork/Join Queueing Networks with Blocking," Technical Report MASI, No 90-32, Université Pierre et Marie Curie, Paris, June.

Dallery, Y., Liu, Z., and Towsley, D. (1991), "Reversibility in Fork/Join Queueing Networks with Blocking-After-Service," in preparation.

Dallery, Y., and Towsley, D. (1990), "Symmetry Property of the Throughout in Closed Tandem Queueing Networks with Finite Buffers," Technical Report MASI, No 90-23, Université Pierre et Marie Curie, Paris, May, to appear in Oper. Res. Letters.

Dattatreya, E. S. (1978), "Tandem Queueing Systems with Blocking," Ph. D. Thesis, Department of Industrial Engineering and Operations Research, University of California, Berkeley.

David, R., Semery, A., Ancelin, B., and Terracol, C. (1987), "Evaluation des Performances d'une Ligne de Production par des Methodes d'Agrégation," INRIA Second International Conference on Production Systems, Paris, April, 1987.

David, R., Xie, X.-L., and Dallery, Y. (1990), "Properties of Continuous Models of Transfer Lines with Unreliable Machines and Finite Buffers," IMA J. of Mathematics in Business and Industry, Vol. 6, pp. 281-308.

De Kok, A. G. (1988), "Computationally Efficient Approximations for Balanced Flowlines with Finite Intermediate Storage," Philips Centre for Quantitative Methods Report CQM-Note nr. 073.

De Koster, M. B. M. (1987), "Estimation of Line Efficiency by Aggregation," Int. Jour. of Prod. Res., Vol. 25, No. 4, pp. 615-626.

De Koster, M. B. M. (1987), "Approximation of Assembly-Disassembly Systems," Report BDK/ORS/87-02, Department of Industrial Engineering, Eindhoven University of Technology, Netherlands.

De Koster, M. B. M. (1988a), "Capacity Oriented Analysis and Design of Production Systems," Ph. D. Thesis, Department of Ind. Eng. and Manag. Sci., Eindhoven University of Technology, Eindhoven.

De Koster, M. B. M. (1988b), "An Improved Algorithm to Approximate the Behaviour of Flow Lines," Int. Jour. of Prod. Res. , Vol. 26, pp. 691-700.

De Koster, M. B. M. (1988c), "Approximate Analysis of Production Systems," Europ. J. Opnl. Res, Vol. 37, pp. 214-226.

De Koster, M. B. M. and Wijngaard, J. (1989), "Continuous vs. Discrete Models for Production Lines with Blocking," In Queueing Networks with Blocking, H. G. Perros and T. Altiok (eds.), North- Holland, Amsterdam, pp. 175-192.

Di Mascolo, M., David, R., and Dallery, Y. (1991), "Modelling and Analysis of Assembly Systems with Unreliable Machines and Finite Buffers," IIE Transactions, to appear.

Di Mascolo, M., Frein, Y., Dallery, Y., and David, R. (1990), "A Unified Modeling of Kanban Systems Using Petri Nets," Int. Jour. of Flexible Manufacturing Systems, to appear.

Dubois, D. and Forestier, J.-P. (1982), "Productivité et en Cours Moyen d'un Ensemble de Deux Machines Séparées par une Zone de Stockage," RAIRO Automatique, Vol. 16, No. 2, pp. 105-132.

Dudick, A. (1979), "Fixed-Cycle Production Systems with In-Line Inventory and Limited Repair Capability," Ph. D. Thesis, Columbia University, 1979.

Elsayed, E.A. and Turley, R.E. (1980), "Reliability Analysis of Production Systems with Buffer Storage," Int. Jour. of Prod. Res., Vol. 18, No. 5.

Elsayed and Hwang (1984), "Analysis of Manufacturing Systems with Buffer Storage and Redundant Machines," IE Working Paper 84-101, Department of Industrial Engineering, Rutgers University.

Erpsher, Y. B. (1952), "Losses of Working Time and Division of Automatic Lines into Sections," Stanki i Instrument, Vol. 23, No. 7, pp. 7-11 and pp. 12-16. (In Russian). (English translation DSIR CTS 631 and CTS 634).

Fanti, M. P., Maione, B., Peluso, R., and Turchiano, B. (1977), "Large Markov Chain Modelling and Analysis of Transfer Lines with Unreliable Work Stations and Finite Buffers," Working Paper, Dipartimento di Elettrotecnica ed Ellectronica, Universita di BARI, Italy.

Forestier, J.-P. (1980), "Modélisation Stochastique et Comportement Asymptotique d'un Systeme Automatisé de Production," RAIRO Automatique, Vol. 14, No. 2, pp. 127-144.

Frein, Y., Commault, C., and Dallery, Y. (1989), "Analytical Performance Evaluation of Closed Transfer Lines with Limited Number of Pallets," Laboratoire d'Automatique de Grenoble Technical Report, September, 1991.

Frein, Y. and Dallery, Y. (1989), "Analysis of cyclic queueing networks with finite buffers and service blocking," Performance Evaluation, Vol. 10.

Gaver, D. P. (1962), "A Waiting Line with Interrupted Service, Including Priorities," J. Roy. Stat. Soc., Vol. B24, pp. 73-90.

Gershwin, S. B. (1986a), "Assembly/Disassembly Systems: An Efficient Decomposition Algorithm for Tree-Structured Networks," Massachusetts Institute of Technology Laboratory for Information and Decision Systems Report LIDS-P-1579, July, 1986.

Gershwin, S. B. (1986b), "Assembly/Disassembly Systems: An Efficient Decomposition Algorithm for Tree-Structured Networks," IEEE SMC.

Gershwin, S. B. (1987a), "An Efficient Decomposition Algorithm for The Approximate Evaluation of Tandem Queues with Finite Storage Space and Blocking," Opns. Res., Vol. 35, pp. 291-305.

Gershwin, S. B. (1987b), "Representation and Analysis of Transfer Lines with Machines that have Different Processing Rates," Annals of Operations Research, Volume 9, pp. 511-530, 1987.

Gershwin, S. B. (1989), "An Efficient Decomposition Algorithm for Unreliable Tandem Queuing Systems with Finite Buffers," In Queueing Networks with Blocking, H.G. Perros and T. Altiok (eds.), North-Holland, Amsterdam, pp. 127-146.

Gershwin, S. B. (1991a), "Assembly/Disassembly Systems: An Efficient Decomposition Algorithm for Tree-Structured Networks," IIE Transactions, to appear, 1991.

Gershwin, S. B. (1991b), Manufacturing Systems Engineering, in preparation.

Gershwin, S. B. and O. Berman (1981), "Analysis of Transfer Lines Consisting of Two Unreliable Machines with Random Processing Times and Finite Storage Buffers," AIIE Transactions, Vol. 13, No. 1, March 1981.

Gershwin, S. B. and Schick, I. C. (1980), "Continuous Model of an Unreliable Two-Stage Material Flow System with a Finite Interstage Buffer, Report LIDS-R-1039, Massachusetts Institute of Technology, Cambridge.

Gershwin, S. B. and Schick, I. C. (1983), "Modelling and Analysis of Three-Stage Transfer Lines with Unreliable Machines and Finite Buffers," Operations Research, Vol. 31, No. 2, pp. 354-380.

Glassey C. R. and Hong Y. (1986a), "The Analysis of Behavior of an Unreliable Two-Stage Automatic Transfer Line with Inter-Stage Buffer Storage," Tech. Rep., Dept. of Ind. Eng. and Oper. Res., University of California, Berkeley, CA.

Glassey, C. R. and Hong Y. (1986b), "The Analysis of Behavior of an Unreliable N-Stage Automatic Transfer Line with (N-1) Inter-Stage Buffer Storages," Tech. Rep., Dept. of Ind. Eng. and Oper. Res., University of California, Berkeley, CA.

Gordon, W. J. and Newell, G. F. (1967), "Cyclic Queueing Systems with Restricted Length Queues," Op. Res., Vol. 15, No. 2, pp. 266-277.

Gun, L. (1987), "Tandem Queueing Systems Subject to Blocking with Phase-Type Servers: Analytic Solutions and Approximations," M.S. Thesis, Department of Elec. Eng., University of Maryland, College Park.

Gun, L. and Makowski, A. M. (1987), "Matrix-Geometric Solution for Finite Capacity Queues with Phase-Type Distributions," Stochastic Models, Performance'87, P.-J. Courtois and G. Latouche (eds.), North-Holland, Amsterdam, pp. 269-282.

Gun, L. and Makowski, A. M. (1989), "An Approximation Method for General Tandem Queueing Systems Subject to Blocking," In Queueing Networks with Blocking, H.G. Perros and T. Altiok (eds.), North-Holland, Amsterdam, pp. 147-174.

Gun, L. and Makowski, A. M. (1990), "Matrix-Geometric Solution for Two-Node Tandem Queueing Systems with Phase-Type Servers Subject to Blocking and Failures," Stochastic Models, to appear.

Harrison, J. M. (1973),"Assembly-Like Queues," Journal of Applied Probability, Volume 10, pp. 354-367, 1973.

Haydon, B. J. (1973), "The Behavior of Systems of Finite Queues," Unpublished Ph.D. thesis, University of New South Wales, Kensington.

Herzog, U., Woo, R., and Chandy, K. M. (1975), "Solution of Queueing Problems by a Recursive Technique," IBM J. Res. Develop., Vol. 19, pp. 295-300.

Hildebrand, D. K. (1967), "Stability of Finite Queue, Tandem Server Systems," J. Appl. Prob., Vol. 4, pp. 571-583.

Hildebrand, D. K. (1968), "On the Capacity of Tandem Server, Finite Queue, Service Systems," Op. Res., Vol. 16, pp. 72-82.

Hillier, F. S. and Boling, R. W. (1966), "The Effect of Some Design Factors on the Efficiency of Production Lines with Variable Operation Times," J. Ind. Eng., Vol. 17, No. 12, pp. 651-658.

Hillier, F. S. and Boling, R. W. (1967), "Finite Queues in Series with Exponential or Erlang Service Times — a Numerical Approach," Op. Res., Vol. 16, No. 2, pp. 286-303.

Hillier, F. S. and Boling, R. W. (1972), "Optimal Allocation of Work in Production Line Systems with Variable Operation Times," Stanford University, Department of Operations Research, Technical Report No. 33, December 30, 1972.

Hillier, F. S. and Boling, R. W. (1977), "Toward Characterizing the Optimal Allocation of Work in Production Lines with Variable Operation Times," in Advances in Operations Research, Proceedings of EURO II, Marc Reubens, Editor; North-Holland, Amsterdam.

Hillier, F. S. and Boling, R. W. (1979), "On the Optimal Allocation of Work in Symmetrically Unbalanced Production Line Systems with Variable Operation Times," Management Systems, Vol.

Ho, Y. C., Eyler, M. A., and Chien, T. T. (1979), "A Gradient Technique for General Buffer Storage Design in a Production Line," International Journal of Production Research, Vol. 17, No. 6, pp 557-580.

Hong, Y. and Seong, D. (1989), "The Analysis of an Unreliable N-Machine Flow-Line Manufacturing System with Random Processing Times, Tech. Rep. 89-03, Dept. of Ind. Eng., Pohang Institute of Science and Technology.

Hopp, W. J., and Simon, J. T. (1989), "Bounds and Heuristics for Assembly-Like Queues," Queueing Systems, Volume 4, pp. 137-156, 1989.

Hunt, G. C. (1956), "Sequential Arrays of Waiting Lines," Op. Res., Vol. 4, No. 6, pp. 674-683.

Ignall, E. and Silver, A. (1977), "The Output of a Two-Stage System with Unreliable Machines and Limited Storage," AIIE Trans., Vol. 9, No. 2, pp. 183-188.

Jafari, M. A. (1982), Ph. D. Thesis proposal, Syracuse University.

Jafari, M. and Shantikumar, J. G. (1987a), "Exact and Approximate Solutions to Two-Stage Transfer Lines with General Uptime and Downtime Distributions," IIE Transactions, Vol. 19, pp. 412-420.

Jafari, M. and Shanthikumar, J. G. (1987b), "An Approximate Model of Multistage Automatic Transfer Lines with Possible Scrapping of Workpieces," IIE Transactions, Vol. 19, pp. 252-265.

Jun, K. P. and Perros, H. G. (1987), "An Approximate Analysis of Open Tandem Queueing Networks with Blocking and General Service Times, Computer Science Report 87-17, North Carolina State University.

Kleinrock, L. (1975), Queueing Systems, Vol. 1, John Wiley.

Knott, A. D. (1967), "The Efficiency of Series Production Lines," Unpublished Ph. D. thesis, University of New South Wales.

Knott, A. D. (1970), "The Inefficiency of a Series of Work Stations — A Simple Formula," Int. J. Prod. Res., Vol. 8, No. 2, pp. 109-119.

Koenigsberg, E. (1959), "Production Lines and Internal Storage -A Review," Manag. Sci., Vol. 5, pp. 410-433.

Lau, H.-S. (1986a), "The Production Rate of a Two-Stage System with Stochastic Processing Times," Int. J. of Prod. Res., Vol. 24, No. 2, pp. 401-412.

Lau, H.-S. (1986b), "A Directly-Coupled Two-Stage Unpaced Line," IIE Transactions, Vol. 18, pp. 304-312.

Lavenberg, S. S. (1975), "The Steady-State Queueing Time Distribution for the $M/G/1$ Finite Capacity Queue," Management Science, Vol. 21, No. 5, pp. 501-506, January, 1975.

Lim, J.-T., Meerkov, S. M., and Top, F. (1990), "Homogeneous, Asymptotically Reliable Serial Production Lines: Theory and a Case Study," IEEE Transactions on Automatic Control, Vol. 35, No. 5, May, 1990.

Little, J. D. C., (1961), "A Proof of the Queueing Formula $L = \lambda W$," Operations Research, Vol. 9, pp. 383-387, 1961.

Lim, J.-T., and Meerkov, S. M. (1990), "Analysis and Synthesis of Asymptotically Reliable Closed Serial Production Lines," Proceedings of the 29th IEEE Conference on Decision and Control, Honolulu, Hawaii, December, 1990

Liu, X.-G. (1990), "Toward Modeling Assembly Systems: Applications of Queueing Networks with Blocking," Ph. D. Thesis, Department of Management Sciences, University of Waterloo, Waterloo.

Liu, X.-G. and Buzacott, J. A. (1989), "A Zero-buffer Equivalence Technique for Decomposing Queueing Networks with Blocking," In Queueing Networks with Blocking, H. G. Perros and T. Altiok (eds.), North-Holland, Amsterdam, pp. 87-104.

Liu, X.-G. and Buzacott, J. A. (1990), "Approximate Models of Assembly Systems with Finite Inventory Banks," Europ. J. Opnl. Res., Vol. 45, pp. 143-154.

Marie, R. (1980), "Calculating Equilibrium Probabilities for $\lambda(n)/C_k/1/N$ Queues," Proceedings of Performance 80 International Symposium on Computer Modelling, pp. 117-125.

Melamed B. (1986) "A Note on the Reversibility and Duality of some Tandem Blocking Queueing Systems," Management Science, Vol. 32, pp. 1648-1650.

Miltenburg, G. J. (1987), "Variance of the Number of Units Produced on a Transfer Line with Buffer Inventories During a Period of Length T," Naval Research Logistics, Vol. 34, pp. 811-822.

Mitra, D. (1988), "Stochastic Theory of a Fluid Model of Multiple Failure-Susceptible Producers and Consumers Coupled by a Buffer," Advances in Applied Probability, September 1988.

Muth, E. J. (1973), "The Production Rate of a Series of Work Stations with Variable Service Times," Int. J. Prod. Res., Vol. 11, No. 2, pp. 155-169.

Muth, E. J. (1977), "Numerical Methods Applicable to a Production Line," In Algorithmic Methods in Probability, North-Holland/TIMS Studies in the Management Sciences, M. F. Neuts (ed.), North- Holland, Amsterdam, pp. 143-160.

Muth, E. J. (1979), "The Reversibility Property of Production Lines," Mgmt. Sci. 25, 152-158.

Muth, E. J. (1984), "Stochastic Processes and their Network Representations Associated with a Production Line Queueing Model," Europ. J. Oper. Res., Vol. 15, pp. 63-83.

Muth, E. J. (1987), "An Update on Analytic Models of Serial Transfer Lines," Research Report No. 87-15, Dept. of Ind. and Syst. Eng., University of Florida, Gainesville, FL.

Muth, E. J. and Alkaff, A. (1987), "The Throughput Rate of Three- Station Production Lines: A Unifying Solution," Int. J. Prod. Res., Vol. 25, pp. 1405-1413.

Neuts, M. F. (1981), Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach, The John Hopkins Univ. Press.

Newell, G. F. (1979), "Approximate Behavior of Tandem Queues," Springer-Verlag Lecture Notes in Economics and Mathematical Systems Number 171.

Nicola, V. F. (1986), "A Single Server Queue with Mixed Types of Interruptions," Acta Informatica, Vol. 23, pp. 465-486.

Okamura, K. and Yamashina, H. (1977), "Analysis of the Effect of Buffer Storage Capacity in Transfer Line Systems," AIIE Trans., Vol. 9, No. 2, pp. 127-135.

Onvural, R. (1990), "A Survey of Closed Queueing Networks with Finite Buffers," ACM Computing Surveys, Vol. 22, No. 2, pp. 83-121.

Onvural, R. and Perros, H. G. (1986), "On Equivalencies of Blocking Mechanism in Queueing Networks with Blocking," Oper. Res. Lett., Vols. 5-6, pp. 293-298.

Onvural, R. and Perros, H. G. (1987), "Throughput Analysis of Cyclic Queueing Networks with Blocking," Technical Report, CS Dept., North Carolina State University, Raleigh.

Onvural, R. and Perros, H. G. (1989), "Some Equivalencies between Open and Closed Queueing Networks with Finite Buffers," Performance Evaluation, Vol. 9, pp. 111-118.

Onvural, R., Perros, H. G., and Altiok, T. (1987), "On the Complexity of the Matrix-Geometric Solution of Exponential Open Queueing Networks with Blocking," Int. Workshop on Modelling Techniques and Performance Evaluation (Pujolle et al., Eds), North Holland, 1987.

Otero, E. (1987) "A New Algorithm for the Analysis of Tree-Structured Assembly/Disassembly Networks," unpublished private communication, July, 1987.

Ou, J. and Gershwin, S. B. (1989), "The Variance of the Lead Time Distribution of a Two-Machine Transfer Line With a Finite Buffer," MIT Laboratory for Manufacturing and Productivity Report LMP-89-028, August, 1989.

Perros, H. G. (1986), "Queueing Networks with Blocking: A Bibliography," Perf. Eval. Rev., Vol. 12, pp. 8-14.

Perros, H. G. (1988), "A Survey of Two-Node Queueing Networks with Blocking," Technical Report No. 88-06, CS Dept., North Carolina State University, Raleigh.

Perros, H. G. (1989), "Open Queueing Networks with Blocking," in Stochastic Analysis of Computer and Communication Systems, H. Takagi (ed.), North Holland, Amsterdam.

Perros, H. G. and Altiok, T. (1986), "Approximate Analysis of Open Networks of Queues with Blocking: Tandem Configurations," IEEE Trans. Software Eng., Vol. 12, pp. 450-461.

Perros, H. G. and Altiok, T. (1989), Queuing Networks with Blocking, Proceedings of the First International Workshop held in Raleigh, North Carolina, May 20-21, 1988, Edited by H. G. Perros and T. Altiok (1989), Elsevier, 1989.

Philippe, B., Saad, Y., and Stewart, W. J., (1989), "Numerical Methods in Markov Modeling," Tech. Rep. INRIA No. 495, September 1989.

Pierrat, J.-J. (1987), "Modélisation de Systemes a Evénements Discrets Sujets a des Pannes," Doctoral thesis, Laboratoire d'Automatique de Grenoble, Institut National Polytechnique de Grenoble, Grenoble.

Pollock, S. M., Birge, J. R., and Alden J. M. (1985), "Approximation Analysis for Open Tandem Queues with Blocking," IOE Report 85-30, University of Michigan.

Rao, N. P. (1975a), "On the Mean Production Rate of a Two-Stage Production System of the Tandem Type," Int. J. Prod. Res., Vol. 13, No. 2, pp. 207-217.

Rao, N. P. (1975b), "Two-Stage Production Systems with Intermediate Storage," AIIE Trans., Vol. 7, No. 4, pp. 414-421.

Rao, N. P. (1976a), "A Generalization of the Bowl Phenomenon in Series Production Systems," Int. J. Prod. Res., Vol. 14, pp. 437-443.

Rao, N. P. (1976b), "A Viable Alternative to the Method of Stages Solution of Series Production Systems with Erlang Service Times, Int. J. Prod. Res., Vol. 14, pp. 699-702.

Sastry, B.L.N. (1985), "Analysis of Two-Machine Markovian Production Lines and Decomposition of Longer Lines," Ph. D Thesis, Ind. Eng. and Oper. Res. Group, Indian Institute of Technology, Bombay.

Sastry, B.L.N. and Awate, P.G. (1988), "Analysis of a Two Station Flow-Line with Machine Processing Subject to Inspection and Rework," Opsearch, Vol. 25, No. 2, pp. 89-97.

Sauer, C.H. and Chandy, K.M. (1981), Computer Systems Performance Modeling, Prentice-Hall.

Schick, I. C. and Gershwin, S. B. (1978), "Modelling and Analysis of Unreliable Transfer Lines with Finite Interstage Buffers," Massachusetts Institute of Technology Electronic Systems Laboratory Report ESL-FR-834-6.

Schweitzer, P. J. and Altiok, T. (1989), "Aggregate Modelling of Tandem Queues without Intermediate Buffers," In Queueing Networks with Blocking, H. G. Perros and T. Altiok (eds.), North-Holland, Amsterdam, pp. 47-72.

Semery, A. (1987), "Modélisation des LIF: Intégration du Modele de Dubois et Forestier a l'Heuristique de Gershwin, pour les Lignes a Temps de Cycle Différents, Rapport de Recherche AS/619-87, Regie Renault, Direction de la Recherche.

Semery, A. (1988), "Modélisation des LIF: Comportement et Limites des Modeles Basés sur la Methode de Decomposition de Gershwin," Rapport de Recherche AS/641-88, Regie Renault, Direction de la Recherche, April 1988.

Sevast'yanov, B. A. (1962), "Influence of Storage Bin Capacity on the Average Standstill Time of a Production Line," Teoriya Veroyatnostey i ee Primeneniya, Vol. 7, No. 4, pp. 438-447 In Russian). (Eng. tr. Theory of Probability and its Applications, Vol. 7, No. 4 , pp. 429-438 (1962)).

Shanthikumar, J. G. and Tien, C. C. (1983), "An Algorithmic Solution to Two-Stage Transfer Lines with Possible Scrapping of Units," Mgmt. Sci., Vol. 29, pp. 1069-1086.

Shanthikumar, J. G. and Jafari, M. A. (1987), "Bounding the Performance of Tandem Queues with Finite Buffer Spaces," Working Paper, University of California at Berkeley, October.

Shanthikumar, J. G. and Yao, D. D. (1989), "Monotonicity and Concavity Properties in Cyclic Queueing Networks with Finite Buffers," In Queueing Networks with Blocking, H. G. Perros and T. Altiok (eds.), North-Holland, Amsterdam, pp. 325-345.

Sheskin, T. J. (1974), "Allocation of Interstage Storage Along an Automatic Transfer Production Line with Discrete Flow," Ph. D. Thesis, Department of Industrial and Management Engineering, Pennsylvania State University.

Sheskin, T. J. (1976), "Allocation of Interstage Storage Along an Automatic Production Line," AIIE Trans., Vol. 8, No. 1, pp. 146-152.

Smunt, T. L. and Perkins, W. C. (1985), "Stochastic Unpaced Line Design: Review and Further Experimental Results," Journal of Operations Management, Vol. 5, pp. 351-373.

Soyster, A. L. and Toof, D. I. (1976), "Some Comparative and Design Aspects of Fixed Cycle Production Systems," Naval Res. Logistics. Quart., Vol. 23, No. 3, pp. 437-454.

Soyster, A. L., Schmidt, J. W., and Rohrer M. W.(1979), "Allocations of Buffer Capacities for a Class of Fixed Cycle Production Systems," AIIE Transactions, Vol. 11, No. 2.

Stewart, W. J. (1978), A Comparison of Numerical Techniques in Markov Modeling," Comm. of ACM, Vol. 21, No. 2, pp. 144-152.

Stewart, W. J. (1988), "Numerical Solution of Markov Chains: Block Hessenberg Matrices and Solution by Recursion, Tech. Rep. No. 88-29, Department of Computer Science, North Carolina State University.

Suri, R. and Diehl, G. (1984), "A New Building Block for Performance Evaluation of Queueing Networks with Finite Buffers," Proc. ACM Sigmetrics Conf. on Measurement and Modeling of Computer Systems, pp. 134-142.

Suri, R. and Diehl, G. (1986), "A Variable Buffer-Size Model and Its Use in Analyzing Closed Queueing Networks with Blocking," Mgmt. Sci., Vol. 32, pp. 206-224.

Takahashi, Y., Miyahara, H., and Hasegawa, T., (1980), "An Approximation Method for Open Restricted Queuing Networks," Opns. Res., Vol. 28, No. 3, Part I.

Terracol, C. and David, R. (1987a), "Performances d'une Ligne Composée de Machines et de Stocks Intermédiaires," APII, Vol. 21, pp. 239-262.

Terracol, C. and David, R. (1987b), "An Aggregation Method for Performance Evaluation of Transfer Lines with Unreliable Machines and Finite Buffers," IEEE Int. Conf. on Robotics and Automation, Raleigh, NC, April.

Top, F. (1990), "Asymptotic Analysis and Synthesis of Serial Production Systems," Ph. D. Thesis, University of Michigan, 1990.

Van Dijk, N. M. and Lamond, B. F. (1988), "Simple Bounds for Finite Single-Server Exponential Tandem Queues," Opns. Res., Vol. 36, pp. 470-477.

Vladzievskii, A. P. (1952), "Probabilistic Law of Operation and Internal Storage of Automatic Lines," Avtomatika i Telemekhanika, Vol. 13, No. 3, pp. 227-281. (In Russian).

Vladzievskii, A. P. (1953), "Losses of Working Time and the Division of Automatic Lines into Sections," Stanki i Instrument, Vol. 24, No. 10, pp. 9-15 (In Russian). (English translation DSIR CTS 632).

Vladzievskii, A.P. (1958), Avtomaticheskii Linii v Mashinostroenii, Mashgiz.

Wijngaard, J. (1979), "The Effect of Interstage Buffer Storage on the Output of Two Unreliable Production Units in Series with Different Production Rates," AIIE Trans. Vol. 11, pp. 42-47.

Wiley, R. P. (1981), "Analysis of a Tandem Queue Model of a Transfer Line," Massachusetts Institute of Technology Laboratory for Information and Decision Systems Report LIDS-P-1150, September 1981.

Xie, X.-L. (1989), "An Efficient Algorithm for Performance Analysis of Transfer Lines and Its Convergence," Working Paper, INRIA-LORRAINE, France.

Yamazaki, G., Sakasegawa, H. (1975), "Properties of Duality in Tandem Queueing Systems," Annals of the Institute of Statistical Mathematics," Vol. 27, pp. 201-212.

Yamazaki, G., Kawashima, T., and Sakasegawa, H. (1985), "Reversibility of Tandem Blocking Queueing Systems," Mgmt. Sci., Vol. 31, pp. 78-83.

Yeralan, S., and Muth, E. J. (1987), "A General Model of a Production Line with Intermediate Buffer and Station Breakdown," IIE Trans. Vol. 19, pp. 130-139.

Yu Retsker, I. and Bunin A. A. (1964), "Determining the Main Parameters of Transfer Lines", Stanki i Instrument Volume 35, No. 6, 17.

Zimmern, B. (1956), "Etudes de la Propagation des Arrets Aleatoires dans les Chaines de Production," Revue de statistique appliquée, Vol. 4, p. 85-104.