OPERATIONS RESEARCH CENTER

Working Paper

Modeling and Exact Analysis of a Production Line with Two Unreliable Batch Machines and a Finite Buffer: Part I - Full Batches

by

Seok Ho Chang
Stanley B. Gershwin

OR 376-05 **June 2005**

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Modeling and Exact Analysis of a Production Line with Two Unreliable Batch Machines and a Finite Buffer:
Part I — Full Batches

Seok Ho Chang* and Stanley B. Gershwin†

Laboratory for Manufacturing and Productivity
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, Massachusetts 02139-4307 USA

June 28, 2005

Abstract

This paper considers a production line with two unreliable batch machines and a finite buffer. Batch machines process a set of parts simultaneously; the maximum number in the set is the size of the machine. The purpose of this paper is twofold: (i) to present a model of this system and its exact analysis; (ii) to present new qualitative insights and interpretations of system behavior. We demonstrate new generalized conservation of flow and flow rate-idle time relationships. We also present various performance measures of interest such as production rate, machine efficiencies, probabilities of blocking and starvation, and expected in-process inventory. We demonstrate an equivalence property and describe deadlock behavior. The effect of the sizes of machines on the performance measures is examined, new phenomena and insights are established, and possible interpretations are presented.

1 Introduction

1.1 System description

In this paper, we present a model, and its exact analysis, of a production line consisting of two unreliable batch machines and a finite buffer (Figure 1). Batch machines process many parts simultaneously. Ovens in semiconductor fabrication or composite part production are important examples. Other examples include computer numerical code (CNC) machines in computer aided manufacturing, automated guided vehicle (AGV) in material handling systems, and vehicles in dispatch stations.

*soccer9@mit.edu
†gershwin@mit.edu,http://web.mit.edu/manuf-sys

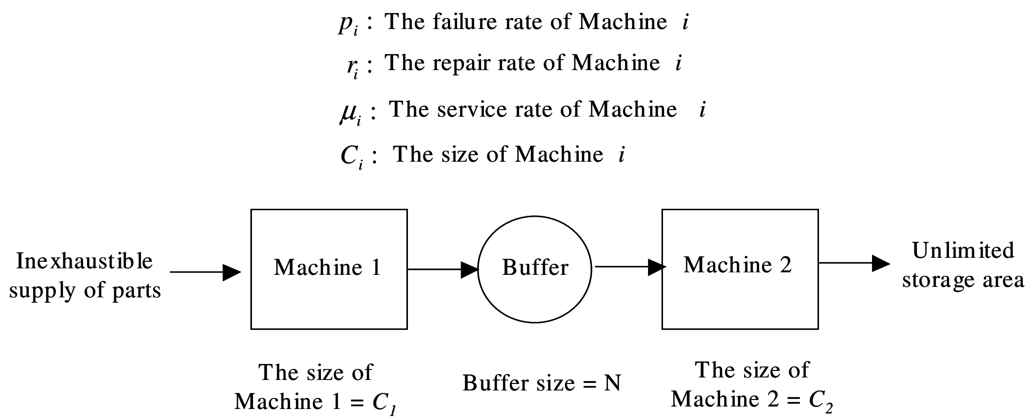

Figure 1: A production line with two unreliable batch machines and a finite buffer

Unlike earlier research on production lines, we assume that each machine has a size which is not necessarily equal to 1. The size of Machine i is C_i and it is defined as the maximum number of parts (or jobs or items) that a machine can process at a time when it is operational. The behavior of Machine i is characterized by three exponentially distributed random variables: the service time of batches (with mean $1/\mu_i$), the time to fail (with mean $1/p_i$ — abbreviated MTTF), and the time to repair (with mean $1/r_i$ — abbreviated MTTR).

1.2 Related literature

Lines with unreliable batch servers and finite buffers appear in a wide variety of settings in the areas of semiconductor manufacturing processes [2, 3, 15, 40, 55], computer aided manufacturing [36], transportation systems [41], and various related areas.

In semiconductor wafer fabrication processes, furnaces or deposition processes accumulate jobs in a buffer, and then process them as a batch of predetermined size [2, 15, 55]. In the burn-in operation in manufacturing processes of very large-scale integrated (VLSI) circuits, VLSI chips are usually processed in batches [40]. In computer aided manufacturing, items to be processed are coded and collected into groups prior to processing [36]. In transportation problems such as dispatching, products arriving at a station are dispatched in batches by a vehicle [41]. About 78 percent of all manufacturing activities in the United States fall into the classification of batch production, with batch sizes ranging from less than 10 to many thousands [24].

The analysis of two-machine one-buffer lines is of great importance because it has been used as the building block of decomposition analysis of larger lines (see, e.g., [18, 20, 29, 30, 31, 33, 39, 44, 45]), and such performance analysis is vital to the design, operation, optimization, and continuous improvement of the corresponding real systems (see, e.g., [9, 32]). We hope that the model studied here will be used in that way.

In this paper, we address both exact analytic and computational aspects of a production line with two unreliable batch machines and a finite buffer. The purpose of this paper is twofold: (i) to present a model of this system and its exact analysis; (ii) to present new qualitative insights and interpretations of system behavior. Some surprising observations are described.

This research examines the following questions: What is the effect of the sizes of machines on performance measures of interest? What are the appropriate definitions of starvation and blocking for a system with batches? Do versions of conservation of flow and flow rate-idle time relationships hold in this kind of system? What is the equivalence property of this system? Is it possible for this system to be in deadlock? Is this system always ergodic? If not, is there any way to compute the performance measures when this system is not ergodic? Are there new behavioral phenomena of this system? If so, how can we interpret them?

Let us review some related works in the following categories: (i) references that dealt with analytical evaluation of production lines (or flow lines) with machines and buffers (ii) references that dealt with batch servers or batch machines.

All previous research on flow line evaluation assume that parts are processed singly or that the sizes of batches are the same. We define *single-item machines* as machines which have size 1. Much previous research on the analytical evaluation of two-machine flow lines can be found in [5, 7, 8, 19, 21, 25, 26, 27, 28, 43, 51, 56, 57, 59, 60] and others. Many references on the analytical evaluation of production lines can be found in surveys [4, 9, 20, 31, 52]. Practical applications of lines with unreliable single-item machines and finite buffers to industry can be found in [10, 53], and elsewhere. Fundamental equivalence properties in manufacturing networks under the assumption that parts are processed singly are found in [1, 49, 50] and Chapter 5 in [31].

Buzacott and Shanthikumar (Section 7.7.6. of [9]) consider job shops with bulk job transfers. They assume that the sizes of batches are the same and present an approximate analysis of their system. Connors et al. [17] present an open queueing network model for semiconductor manufacturing processes. Though they consider batch machines in their model, they do not consider finite buffers and they do not incorporate machine failure and repair. References that dealt with various types of *single-stage* queueing models with batch arrivals and batch services can be found in [6, 13, 16, 22, 23, 54] and the references therein. The studies on optimal scheduling problems for a perfectly reliable batch machine or for a production line with perfectly reliable batch machines can be found in [3, 37, 40, 42], and their references. The literature on inventory systems with batching can be found in [11, 14], and the references therein. Haut de Sigy [38] develop a stochastic dynamic programming model to determine the loading policy of a batch machine followed by a serial machine in a single product system. Glassey and Wang [34] demonstrate how to use forecasting information to reduce the average waiting time of a lot in a batch work station where the batch sizes are the same. To the best of authors' knowledge, there has been no research which presented analytical modeling and exact analysis of the system of Figure 1.

1.3 Contribution

The primary contributions of this paper can be summarized as follows:

1. We model a new kind of system that incorporates two unreliable batch machines, which may have different sizes.

2. We present definitions of starvation and blocking (Section 2.1.2), which generalize those in [31] for lines with unreliable single-item machines and finite buffers.

3. We investigate deadlock behavior (including the phenomenon of deadlock, the condition for deadlock, and the prevention of deadlock) in this system (Section 2.1.3). Deadlock cannot be observed in lines with two unreliable single-item machines and a finite buffer.

4. We demonstrate generalized conservation of flow (Theorem 2 of Section 4.4) and flow rate-idle time relationships (Theorem 1 of Section 4.3).

5. We offer a new proof of conservation of flow (Theorem 2 of Section 4.4). We also present an interpretation of our conservation of flow theorem based on Little's law [46] (Remark 2 of Section 4.4.)

6. We demonstrate an equivalence property (Theorem 3 of Section 4.5).

7. We determine cases in which the system is not ergodic and we explain how to compute the performance measures when the system is not ergodic (Section 5).

8. We perform a numerical analysis of the system. We develop code in MATLAB that produces numerically stable performance measures of interest (e.g., production rate, efficiencies of machines, starvation and blocking probabilities, and expected in-process inventory) in a short time for most cases (Section 6).

9. Based on our code, we present numerical results and their qualitative interpretations. Based on the theoretical and numerical results presented, new relationships between the sizes of machines and performance measures in unreliable batch production line with a finite buffer, have been identified and interpreted (Sections 7 and 8).

Finally, to verify our results, we match the results of the single-item special case of this system with those of the single-item model of [31] both analytically and numerically. The results match perfectly. We also perform a simulation to verify our performance measures, which is not shown here. Our computational experience shows that the average absolute relative percentage differences between our exact mean values and simulation estimates were within 1.1%. Our computational experience further shows that all the exact numerical results obtained in this paper are consistent with our analytical formulae and results presented in this paper (Section 7).

1.4 Outline

The rest of this paper is organized as follows: In Section 2, we describe the preliminary issues and the assumptions of our system. We present a set of balance equations, stationary probabilities and the expressions of performance measures in Section 3. In Section 4, we show some theoretical results. Section 5 discusses the issue of calculating performance measures when the system is not ergodic. Section 6 briefly exhibits our computational scheme. In Section 7, we present sample numerical results and qualitative observations. Finally, Section 8 concludes this paper.

2 System and Assumptions

2.1 Preliminary issues

In this section, we present some preliminary issues which should be clarified in lines with unreliable batch machines and finite buffers.

2.1.1 Control policy of each machine

In this subsection, we present the control policy of each machine in production lines with unreliable batch machines and finite buffers (Figure 1).

By convention, we assume that an inexhaustible supply of parts is available upstream of Machine 1 in the line, and an unlimited storage area is present downstream of Machine 2. Under this setting, we assume that Machine i can process a batch of C_i parts if it is operational and that number of parts or spaces are available. However, we must establish rules for what do to with the machine when fewer than C_i parts or spaces are available. We call this kind of rule a *control policy.* There are at least two possible control policies, called *full-batch* and *partial-batch* for each machine:

Control policies for Machine 1

(1) Full-batch:

- If there are at least C_1 spaces in the buffer and Machine 1 is available, Machine 1 processes a batch of C_1 parts simultaneously.
- If there are fewer than C_1 spaces in the buffer and Machine 1 is available, Machine 1 does not operate. It is idle until there are at least C_1 spaces.

(2) Partial-batch:

- If there are at least C_1 spaces in the buffer and Machine 1 is available, Machine 1 processes a batch of C_1 parts simultaneously.
- If there are s $(0 < s < C_1)$ spaces in the buffer and Machine 1 is available, Machine 1 processes a batch of s parts simultaneously.

Control policies for Machine 2

(1) Full-batch:

- If there are at least C_2 parts in the buffer and Machine 2 is available, Machine 2 processes a batch of C_2 parts.
- If there are fewer than C_2 parts in the buffer and Machine 2 is available, Machine 2 does not operate. It is idle until there are at least C_2 parts.

(2) Partial-batch:

- If there are at least C_2 parts in the buffer and Machine 2 is available, Machine 2 processes a batch of C_2 parts.
- If there are $s < C_2$ parts in the buffer and Machine 2 is available, Machine 2 works on a batch of s parts.

In this paper, we assume the full-batch control policy for each machine. The partial-batch policy for each machine is studied in [12].

2.1.2 Generalized definitions of starvation and blocking

In this subsection, we present new generalized definitions of starvation and blocking in lines with unreliable batch machines and finite buffers. In lines with unreliable single-item machines and finite buffers, the definitions of starvation and blocking are as follows [31]: A machine is starved if its upstream buffer is empty. It is blocked if its downstream buffer is full.

However, in lines with batch machines that operated with the full-batch policy, these definitions are no longer adequate. We generalize the definitions of the starvation and blocking as follows: Machine i is *starved* if the number of parts in its upstream buffer is less than C_i. It is *blocked* if the number of parts in its downstream buffer is greater than $N - C_i$.

2.1.3 Deadlock

In this subsection, we present an important phenomenon of this system which has no counterpart in single-item systems. For this purpose, let us define the following two sets:

$$A = \{n; N - C_1 + 1 \leq n \leq N\}, \quad B = \{n; 0 \leq n \leq C_2 - 1\}.$$

From the definitions of the full-batch control policy, and the generalized definitions of blocking and starvation presented in Sections 2.1.1 and 2.1.2, Machine 1 is blocked if $n \in A$, and Machine 2 is starved if $n \in B$. Therefore, if $A \cap B$ is not empty, blocking and starvation occur simultaneously for $n \in A \cap B = \{n;\ N - C_1 + 1 \leq n \leq N, \text{ and } 0 \leq n \leq C_2 - 1\}$. If both blocking and starvation occur simultaneously, the line stops permanently. This is deadlock. When the system is in deadlock, the production rate of the system drops to zero, the blocking and starvation probabilities are equal to 1, and efficiencies of machines are equal to zero. To prevent deadlock, $A \cap B = \emptyset$. This will be true if $C_1 + C_2 \leq N + 1$.

2.2 Assumptions

In this section, we present the set of assumptions that define the system.

(1) We assume that an inexhaustible supply of parts is available upstream of Machine 1 in the production line, and an unlimited storage area is present downstream of Machine 2.

(2) We adopt the full-batch control policy for each machine presented in Section 2.1.1 for both machines. We also adopt the new generalized definitions of starvation and blocking defined in Section 2.1.2.

(3) We assume that Machine i can process exactly C_i parts simultaneously if it is operational, where C_i denotes the size of Machine $i (i = 1, 2)$. C_i is a positive integer. Note that C_1 and C_2 are not necessarily the same.

(4) Parts enter the system at Machine 1 in batches of fixed size C_1, then go to the buffer, then go to Machine 2, and then exit the system in batches of fixed size C_2.

(5) Service, failure and repair times of a batch for Machine $i (i = 1, 2)$ are exponential random variables with parameters μ_i, p_i, and r_i; these quantities are called the service rate, failure rate and repair rate, respectively.

(6) The buffer between the machines has a finite size N.

(7) Machine 1 never starves, and Machine 2 is never blocked.

(8) *Operational dependent failures*: A machine fails only while processing a batch.

(9) Batches are not destroyed or rejected at any stage in the line. Partly processed batches are not added into or taken out of the line. When a machine breaks down, the batch it was operating on waits for the machine to be repaired so that processing of a batch can resume.

(10) $C_1 + C_2 \leq N + 1$. (See Section 2.1.3.)

The state of the system at time t is denoted by $(n(t), \alpha_1(t), \alpha_2(t))$, where $n(t)$ represents the number of parts in the buffer at time t, $\alpha_i(t)$ $(i = 1, 2)$ denotes the repair state of Machine i at time t. If Machine i is operational (ie, not under repair) at time t, $\alpha_i(t) = 1$. Otherwise $\alpha_i(t) = 0$, $i = 1, 2$.

3 Balance equations and performance measures

In this section, we present a condition for this system to be ergodic and a set of balance equations for the stationary distribution. We also present the expressions for performance measures of the system.

3.1 Ergodicity condition

Proposition 1 *Ergodicity condition*

The system is ergodic if and only if C_1 and C_2 are relatively prime.

Proof: We define h as the maximum common divisor of C_1 and C_2. Let $\{n(t); t \geq 0\}$ be a buffer level process. Then we have

$$n(t) = n(0) + C_1 A(t) - C_2 D(t), \qquad t \geq 0, \tag{1}$$

where $A(t)$ and $D(t)$, respectively, represent the numbers of arrivals and departures of batches during interval $(0, t]$. $A(t)$ and $D(t)$ are non-negative integers.

1 Assume $h > 1$.

It then follows that

$$n(t) = n(0) + C_1 A(t) - C_2 D(t) = n(0) \mod h. \tag{2}$$

Therefore, $\{(n(t), \alpha_1(t), \alpha_2(t), t \geq 0\}$ is not ergodic because the stationary distribution of a buffer level depends on the initial buffer level $n(0)$ through equation (2).

2 Assume $h = 1$.

Consider $s(0) = (n(0), \alpha_1(0), \alpha_2(0))$ and $s^* = (n^*, \alpha_1^*, \alpha_2^*)$ where s^* is not a transient state. It is possible to construct a sequence of transitions

$$s(0) = (n(0), \alpha_1(0), \alpha_2(0)) \rightarrow s(t_1) = (n(t_1), \alpha_1(t_1), \alpha_2(t_1)) \rightarrow s(t_2) = (n(t_2), \alpha_1(t_2), \alpha_2(t_2)) \rightarrow ... \rightarrow s^*$$

for any $s(0)$ and s^*. The sequence is constructed as follows:

1. If $\alpha_1(0) \neq 1$ or $\alpha_2(0) \neq 1$, the first or the first two transitions bring (α_1, α_2) to (1,1).

2. If $n(0) < C_2$, the next set of transitions are enough Machine 1 operations such that $n \geq C_2$. If $n(0) > N - C_1$, the next set of transitions are enough Machine 2 operations such that $n \leq N - C_1$.

3. If $n > n^*$, the next transition is a Machine 2 operation, so n is decreased by C_2. If $n < n^*$, the next transition is a Machine 1 operation, so n is increased by C_1.

4. Repeat Step 3 until $n = n^*$. This will occur eventually because C_1 and C_2 are relatively prime, and because n can always be expressed as $n = n(0) + D_1 C_1 - D_2 C_2$ where D_1 is the number of times that M_1 is operated and D_2 is the number of times that M_2 is operated. From Theorem 8.6 of [48], for any n^* and $n(0)$, there exist integers D_1 and D_2 such that $n^* - n(0) = D_1 C_1 - D_2 C_2$.

5. The last one or two transitions bring (α_1, α_2) from (1,1) to (α_1^*, α_2^*).

Since the system can get from any initial state to any non-transient final state the system is ergodic. In the finite-state Markov process $\{(n(t), \alpha_1(t), \alpha_2(t)); t \geq 0\}$, there is a set of transient states, and a single final class which is irreducible. Thus, a unique stationary distribution exists.

Remark 1 When $h \neq 1$, consider the transformed process $\{(m(t), \alpha_1(t), \alpha_2(t)); t \geq 0\}$ associated with the original process $\{(n(t), \alpha_1(t), \alpha_2(t)); t \geq 0\}$ where $m(t)$ is the unique integer that satisfies

$$m(t)h \leq n(t) < (m(t) + 1)h$$

Note that

$$n(t) = m(t)h + e \tag{3}$$

where e is an integer that is determined by $n(0)$. It is given by

$$\begin{gathered} 0 \leq e < h \\ n(0) = m(0)h + e \end{gathered} \tag{4}$$

The transformed process $\{(m(t), \alpha_1(t), \alpha_2(t)); t \geq 0\}$ is equivalent to $\{(n(t), \alpha_1(t), \alpha_2(t)); t \geq 0\}$ when we replace C_i by C_i/h and N by M where M is the largest integer such that

$$N \geq Mh + e \tag{5}$$

This is because the only values that $n(t)$ can take are given by (3), and the transition rates among the states in the original system are the same as those among corresponding states in the transformed system. The transformed system can be thought of as one whose parts are sets of h parts in the original system.

The transformed system is ergodic. This is because of how we constructed the sizes of the machines. Since the original sizes were reduced by their greatest common divisor, the new sizes are relatively prime. Therefore all the results derived in this paper for ergodic systems apply to the transformed system.

In the rest of the paper, we assume that C_1 and C_2 are relatively prime so that the system is ergodic unless we explicitly state otherwise (such as in Section 5). We also assume that the system has reached its steady state.

3.2 Balance equations

We define the stationary distribution $P(n, \alpha_1, \alpha_2)$ as follows:

$$P(n, \alpha_1, \alpha_2) \equiv \lim_{t \to \infty} P\{(n(t), \alpha_1(t), \alpha_2(t)) = (n, \alpha_1, \alpha_2)\},\ 0 \leq n \leq N,\ \alpha_i = 0, 1.$$

In this subsection, we present a set of balance equations of this system. We assume

$$P(n, \alpha_1, \alpha_2) \equiv 0,\ \text{for } n < 0 \quad \text{and} \quad P(n, \alpha_1, \alpha_2) \equiv 0,\ \text{for } n > N. \tag{6}$$

We distinguish four sets of equations, corresponding to the values of α_1 and α_2:

Case 1 $\alpha_1 = 0, \alpha_2 = 0$:

$$P(n, 0, 0)(r_1 + r_2) = P(n, 1, 0)p_1 + p(n, 0, 1)p_2,\ \text{for } C_2 \leq n \leq N - C_1. \tag{7}$$

$$P(n, 0, 0)(r_1 + r_2) = P(n, 1, 0)p_1,\ \text{for } 0 \leq n \leq C_2 - 1. \tag{8}$$

$$P(n, 0, 0)(r_1 + r_2) = P(n, 0, 1)p_2,\ \text{for } N - C_1 + 1 \leq n \leq N. \tag{9}$$

Discussion: The left hand sides of these equations represent the rate at which the system leaves state $(n, 0, 0)$. They reflect the fact that the system leaves state $(n, 0, 0)$ only when the repair of a machine is complete. The right hand sides represent the rates at which the system enters state $(n, 0, 0)$. The explanations of the ranges of buffer level in the right hand sides are as follows:

Recall that we define the following two sets in Section 2.1.3:

$$A = \{n; N - C_1 + 1 \leq n \leq N\}, \quad B = \{n; 0 \leq n \leq C_2 - 1\}.$$

For $n \in A$, Machine 1 is blocked (subsection 2.1.2), so that it cannot fail (by Assumption 8). For $n \in B$, Machine 2 starves (subsection 2.1.2), so that it cannot fail (by Assumption 8). From these observations and the assumption (10) in Section 2, we get the following three cases:

(i) For $n \in A^C \cap B^C = \{n; C_2 \leq n \leq N - C_1\}$,

For this case, neither blocking (of Machine 1) nor starvation (of Machine 2) occurs. Thus, both machines can fail (by Assumption 8). From this observation, the system can reach state $(n, 0, 0)$ either from state $(n, 1, 0)$ if machine 1 fails or from state $(n, 0, 1)$ if Machine 2 fails. This argument leads to (7).

(ii) For $n \in A^C \cap B = \{n; 0 \leq n \leq C_2 - 1\}$,

For this case, Machine 1 cannot be blocked and Machine 2 is starved. Thus, Machine 1 can fail, while the Machine 2 cannot fail. From this observation, the system can reach state $(n, 0, 0)$ only from state $(n, 1, 0)$ if Machine 1 fails. This argument leads to (8).

(iii) For $n \in A \cap B^C = \{n; N - C_1 + 1 \leq n \leq N\}$,

For this case, Machine 1 is blocked and Machine 2 cannot starve. Thus, Machine 1 cannot fail, while Machine 2 can fail. From this observation, the system can reach state $(n, 0, 0)$ only from state $(n, 0, 1)$ if Machine 2 fails. This argument leads to (9).

The other sets of equations can be obtained similarly:

Case 2 $\alpha_1 = 0, \alpha_2 = 1$:

$$P(n,0,1)(r_1 + \mu_2 + p_2) = P(n,0,0)r_2 + P(n,1,1)p_1 + P(n + C_2, 0, 1)\mu_2, \quad \text{for } C_2 \leq n \leq N - C_1. \tag{10}$$

$$P(n,0,1)r_1 = P(n,0,0)r_2 + P(n,1,1)p_1 + P(n + C_2, 0, 1)\mu_2, \quad \text{for } 0 \leq n \leq C_2 - 1. \tag{11}$$

$$P(n,0,1)(r_1 + \mu_2 + p_2) = P(n,0,0)r_2 + P(n + C_2, 0, 1)\mu_2, \quad \text{for } N - C_1 + 1 \leq n \leq N. \tag{12}$$

Case 3 $\alpha_1 = 1, \alpha_2 = 0$:

$$P(n,1,0)(p_1 + \mu_1 + r_2) = P(n - C_1, 1, 0)\mu_1 + P(n,0,0)r_1 + P(n,1,1)p_2, \quad \text{for } C_2 \leq n \leq N - C_1. \tag{13}$$

$$P(n,1,0)(p_1 + \mu_1 + r_2) = P(n - C_1, 1, 0)\mu_1 + P(n,0,0)r_1, \quad \text{for } 0 \leq n \leq C_2 - 1 \tag{14}$$

$$P(n,1,0)r_2 = P(n - C_1, 1, 0)\mu_1 + P(n,0,0)r_1 + P(n,1,1)p_2, \quad \text{for } N - C_1 + 1 \leq n \leq N \tag{15}$$

Case 4 $\alpha_1 = 1, \alpha_2 = 1$:

$$P(n,1,1)(p_1 + p_2 + \mu_1 + \mu_2) = P(n - C_1, 1, 1)\mu_1 + P(n + C_2, 1, 1)\mu_2 + P(n,1,0)r_2 + P(n,0,1)r_1, \quad \text{for } C_2 \leq n \leq N - C_1. \tag{16}$$

$$P(n,1,1)(p_1 + \mu_1) = P(n - C_1, 1, 1)\mu_1 + P(n + C_2, 1, 1)\mu_2 + P(n,1,0)r_2 + P(n,0,1)r_1, \quad \text{for } 0 \leq n \leq C_2 - 1. \tag{17}$$

$$P(n,1,1)(p_2+\mu_2) = P(n-C_1,1,1)\mu_1 + P(n+C_2,1,1)\mu_2 + P(n,1,0)r_2 + P(n,0,1)r_1,$$
$$\text{for } N-C_1+1 \le n \le N. \tag{18}$$

Normalization condition

$$\sum_{n=0}^{N}\sum_{\alpha_1=0}^{1}\sum_{\alpha_2=0}^{1} P(n,\alpha_1,\alpha_2) = 1. \tag{19}$$

Special case: When $C_1 = C_2 = 1$, the above balance equations are reduced to those presented in [31] for lines with single-item machines.

Once we compute the stationary probabilities (see Section 6 for our computational scheme), we can compute the performance measures of interest using the formulas presented in Section 3.3.

3.3 Performance measures

Efficiency Efficiency E_i is defined as the probability that Machine i is operating on a batch, or the long-run fraction of time in which Machine i produces batches. E_1 and E_2 can be expressed as follows:

$$E_1 = \sum_{n=0}^{N-C_1}\sum_{\alpha_2=0}^{1} P(n,1,\alpha_2), \tag{20}$$

$$E_2 = \sum_{n=C_2}^{N}\sum_{\alpha_1=0}^{1} P(n,\alpha_1,1). \tag{21}$$

Isolated efficiency The isolated efficiency e_i of Machine i is given by

$$e_i = \frac{r_i}{r_i+p_i}, \tag{22}$$

and it represents the long-run fraction of time that Machine i is operational.

Probabilities of blocking and starvation The stationary probabilities of blocking and starvation, P_B and P_S, respectively, are:

$$P_B = \sum_{n=N-C_1+1}^{N}\sum_{\alpha_2=0}^{1} P(n,1,\alpha_2). \tag{23}$$

$$P_S = \sum_{n=0}^{C_2-1}\sum_{\alpha_1=0}^{1} P(n,\alpha_1,1). \tag{24}$$

Production rate The production rate of Machine i, P_i, is given by

$$P_i = \mu_i C_i E_i. \tag{25}$$

Isolated production rate The isolated production rate of Machine i, ρ_i, is given by

$$\rho_i = \mu_i C_i e_i, \tag{26}$$

and it represents what the production rate of Machine i would be if it were never impeded by other machines or buffers.

Expected in-process inventory The expected in-process inventory in the buffer can be written as follows:

$$\bar{n} = \sum_{n=0}^{N} \sum_{\alpha_1=0}^{1} \sum_{\alpha_2=0}^{1} nP(n, \alpha_1, \alpha_2). \tag{27}$$

4 Theoretical results

In this section, we present some theoretical results about this system.

4.1 Transient states

We present the transient states of this system in Lemma 1, which generalize the results in p. 101 of [31].

Lemma 1 *Identification of transient states*

$$P(n, 0, 0) = P(n, 1, 0) = 0, \text{ for } 0 \le n \le C_2 - 1. \tag{28}$$

$$P(n, 0, 0) = P(n, 0, 1) = 0, \text{ for } N - C_1 + 1 \le n \le N. \tag{29}$$

Proof of (28):

(1) Combining (8) and (14), we get

$$P(n, 0, 0)r_2 + P(n, 1, 0)(\mu_1 + r_2) = P(n - C_1, 1, 0)\mu_1, \text{ for } 0 \le n \le C_2 - 1. \tag{30}$$

We consider the following two cases:

Case 1 $C_1 \ge C_2$

For this case, the term $P(n - C_1, 1, 0)\mu_1$ in the right side of (30) automatically disappears since $n - C_1 < 0$ under the condition that $0 \le n \le C_2 - 1$ and $C_1 \ge C_2$. So we get

$$P(n,0,0)r_2 + P(n,1,0)(\mu_1 + r_2) = 0, \text{ for } C_1 \ge C_2,\ 0 \le n \le C_2 - 1. \tag{31}$$

Since probabilities are nonnegative, we get

$$P(n,0,0) = P(n,1,0) = 0, \text{ for } C_1 \ge C_2,\ 0 \le n \le C_2 - 1. \tag{32}$$

Case 2 $C_1 < C_2$

For this case, (30) can be divided into the following two equations:

$$P(n,0,0)r_2 + P(n,1,0)(\mu_1 + r_2) = 0, \text{ for } 0 \le n \le C_1 - 1. \tag{33}$$

$$P(n,0,0)r_2 + P(n,1,0)(\mu_1 + r_2) = P(n - C_1, 1, 0)\mu_1, \text{ for } C_1 \le n \le C_2 - 1. \tag{34}$$

Since probabilities are nonnegative, we get the following from (33):

$$P(n,0,0) = P(n,1,0) = 0, \text{ for } C_1 < C_2\ ,\ 0 \le n \le C_1 - 1. \tag{35}$$

All that remains is to show that $P(n,0,0) = P(n,1,0) = 0$ for $C_1 \le n \le C_2 - 1$. First, suppose $C_1 \le n < \min(C_2 - 1, 2C_1)$. Then $n - C_1 < C_1$, so (34) implies that

$$P(n,0,0) = P(n,1,0) = 0 \text{ for } C_1 \le n < \min(C_2 - 1, 2C_1)$$

Suppose that we have already established that

$$P(n,0,0) = P(n,1,0) = 0 \text{ for } (K-1)C_1 \le n < \min(C_2 - 1, KC_1)$$

for integer K, and that $KC_1 \le n < \min(C_2 - 1, (K+1)C_1)$. Then, since $(K-1)C_1 \le n - C_1 < \min(C_2 - 1, KC_1)$, (34) implies that

$$P(n,0,0) = P(n,1,0) = 0 \text{ for } KC_1 \le n < \min(C_2 - 1, (K+1)C_1)$$

Eventually $KC_1 > C_2 - 1$, and the process stops. Equation (28) is proved by induction.

Proof of (29):

(2) Combining (9) and (12), we get

$$P(n,0,0)r_1 + P(n,0,1)(r_1 + \mu_2) = P(n + C_2, 0, 1)\mu_2, \text{ for } N - C_1 + 1 \le n \le N. \tag{36}$$

We consider the following two cases:

Case 1 $C_1 < C_2$

For this case, it can be seen that the term $P(n + C_2, 0, 1)\mu_2$ in the right side of (36) automatically disappears since $n + C_2 > N$ under the condition that $N - C_1 + 1 \leq n \leq N$ and $C_1 < C_2$. So we get

$$P(n,0,0)r_1 + P(n,0,1)(r_1 + \mu_2) = 0, \text{ for } C_1 < C_2, \ N - C_1 + 1 \leq n \leq N. \quad (37)$$

Since probabilities are nonnegative, we get

$$P(n,0,0) = P(n,0,1) = 0, \text{ for } C_1 < C_2, \ N - C_1 + 1 \leq n \leq N. \quad (38)$$

Case 2 $C_1 \geq C_2$

For this case, (36) can be divided into the following two equations:

$$P(n,0,0)r_1 + P(n,0,1)(r_1 + \mu_2) = 0, \text{ for } N - C_2 + 1 \leq n \leq N. \quad (39)$$

$$P(n,0,0)r_1 + P(n,0,1)(r_1 + \mu_2) = P(n + C_2, 0, 1)\mu_2, \text{ for } N - C_1 + 1 \leq n \leq N - C_2. \quad (40)$$

Since probabilities are nonnegative, we get the following from (39):

$$P(n,0,0) = P(n,0,1) = 0, \text{ for } C_1 \geq C_2, \ N - C_2 + 1 \leq n \leq N. \quad (41)$$

All that remains is to show that $P(n,0,0) = P(n,0,1) = 0$, for $N - C_1 + 1 \leq n \leq N - C_2$. First, suppose $\max(N - C_1 + 1, N - 2C_2) < n \leq N - C_2$. Then $N - C_2 < n + C_2 \leq N$, so (40) implies that

$$P(n,0,0) = P(n,0,1) = 0 \text{ for } \max(N - C_1 + 1, N - 2C_2) < n \leq N - C_2.$$

Then, suppose that we have already established that

$$P(n,0,0) = P(n,1,0) = 0 \text{ for } \max(N - C_1 + 1, N - SC_2) < n \leq N - (S-1)C_2.$$

for integer S, and that $\max(N - C_1 + 1, N - (S+1)C_2) < n \leq N - SC_2$. Then, since $\max(N - C_1 + 1, N - SC_2) < n + C_2 \leq N - (S-1)C_2$ implies that

$$P(n,0,0) = P(n,1,0) = 0 \text{ for } \max(N - C_1 + 1, N - SC_2) < n \leq N - (S-1)C_2$$

Eventually $N - SC_2 > N - C_1 + 1$, and the process stops. Equation (29) is proved by induction.

4.2 Repair frequency equals failure frequency

Lemma 2 states that the rate of transition from the set of states in which Machine 1 is under repair to the set of states in which Machine 1 is operational is equal to the rate of transitions in the opposite direction. It states the same for Machine 2. Lemma 2 generalizes similar results on p. 102 of [31] for single-item machines.

Lemma 2 *Repair frequency equals failure frequency*

$$r_1 \sum_{n=0}^{N} \sum_{\alpha_2=0}^{1} P(n,0,\alpha_2) = p_1 \sum_{n=0}^{N-C_1} \sum_{\alpha_2=0}^{1} P(n,1,\alpha_2). \tag{42}$$

$$r_2 \sum_{n=0}^{N} \sum_{\alpha_1=0}^{1} P(n,\alpha_1,0) = p_2 \sum_{n=C_2}^{N} \sum_{\alpha_1=0}^{1} P(n,\alpha_1,1). \tag{43}$$

Proof:

(i) Adding Equations (7)–(12), we get

$$\begin{aligned}\sum_{n=0}^{N} P(n,0,0)\ (r_1+r_2) + \sum_{n=0}^{N} P(n,0,1) r_1 + \sum_{n=C_2}^{N} P(n,0,1)(\mu_2+p_2) \\ = p_1 \sum_{n=0}^{N-C_1} P(n,1,0) + p_2 \sum_{n=C_2}^{N} P(n,0,1) + r_2 \sum_{n=0}^{N} P(n,0,0) \\ + p_1 \sum_{n=0}^{N-C_1} P(n,1,1) + \mu_2 \sum_{n=0}^{N} P(n+C_2,0,1).\end{aligned} \tag{44}$$

Using the convention that $P(n,0,1)=0$ for $n>N$, it can be seen that (44) is equivalent to

$$r_1 \sum_{n=0}^{N} P(n,0,0) + r_1 \sum_{n=0}^{N} P(n,0,1) = p_1 \sum_{n=0}^{N-C_1} P(n,1,0) + p_1 \sum_{n=0}^{N-C_1} P(n,1,1), \tag{45}$$

from which we get (42).

(ii) Adding Equations (7)–(9), (13)–(15), we get

$$\begin{aligned}\sum_{n=0}^{N} P(n,0,0)\ (r_1+r_2) + \sum_{n=0}^{N} P(n,1,0) r_2 + \sum_{n=0}^{N-C_1} P(n,1,0)(\mu_1+p_1) \\ = p_1 \sum_{n=0}^{N-C_1} P(n,1,0) + p_2 \sum_{n=C_2}^{N} P(n,0,1) + r_1 \sum_{n=0}^{N} P(n,0,0) \\ + p_2 \sum_{n=C_2}^{N} P(n,1,1) + \mu_1 \sum_{n=0}^{N} P(n-C_1,1,0).\end{aligned} \tag{46}$$

Using the convention that $P(n,1,0)=0$ for $n<0$, it can be seen that (46) can be written as

$$r_2\sum_{n=0}^{N}P(n,0,0)+r_2\sum_{n=0}^{N}P(n,1,0)=p_2\sum_{n=C_2}^{N}P(n,0,1)+p_2\sum_{n=C_2}^{N}P(n,1,1), \tag{47}$$

from which we get (43). This completes the proof.

4.3 Generalized flow rate-idle time relationships

In Theorem 1, we present generalized flow rate-idle time relationships.

Theorem 1 *Flow rate-idle time relationships*

$$E_1=e_1P(n\le N-C_1)=\frac{r_1}{r_1+p_1}(1-P_B). \tag{48}$$

$$E_2=e_2P(n\ge C_2)=\frac{r_2}{r_2+p_2}(1-P_S). \tag{49}$$

where P_B and P_S are defined in (23) and (24).

Proof:
(i) Let $D_1=\sum_{n=0}^{N-C_1}\sum_{\alpha_2=0}^{1}P(n,0,\alpha_2)$.
Using (20) and Lemma 1, it can be seen that

$$E_1+D_1=\sum_{n=0}^{N-C_1}\sum_{\alpha_1=0}^{1}\sum_{\alpha_2=0}^{1}P(n,\alpha_1,\alpha_2)=1-P_B, \tag{50}$$

where P_B is as defined in (23).

Taking the transient states (i.e., $P(n,0,0)=P(n,0,1)=0,$ for $N-C_1+1\le n\le N$) into account, we can express (42) as follows:

$$r_1D_1=p_1E_1. \tag{51}$$

Combining (50) and (51), we get

$$E_1=\frac{r_1}{r_1+p_1}(1-P_B). \tag{52}$$

Combining (23) and (52), we get (48).

(ii) Let $D_2=\sum_{n=C_2}^{N}\sum_{\alpha_1=0}^{1}P(n,\alpha_1,0)$.
Using (21), it can be seen that

$$E_2+D_2=\sum_{n=C_2}^{N}\sum_{\alpha_1=0}^{1}\sum_{\alpha_2=0}^{1}P(n,\alpha_1,\alpha_2)=1-P_S, \tag{53}$$

where P_S is as defined in (24).

Taking the transient states (i.e., $P(n,0,0) = P(n,1,0) = 0,\ 0 \le n \le C_2 - 1$) into account, we can express (43) using the notation E_2 and D_2 as follows:

$$r_2 D_2 = p_2 E_2. \tag{54}$$

Combining (53) and (54), we get

$$E_2 = \frac{r_2}{r_2 + p_2}(1 - P_S). \tag{55}$$

Combining (24) and (55), we get (49).

4.4 Generalized conservation of flow

Lemma 3 is helpful in demonstrating a generalized form of conservation of flow.

Lemma 3 *Balance equations*

For $0 \le n \le C_2 - 1$,

$$\mu_1 \sum_{\alpha_2=0}^{1} P(n,1,\alpha_2) = \mu_1 \sum_{\alpha_2=0}^{1} P(n - C_1, 1, \alpha_2) + \mu_2 \sum_{\alpha_1=0}^{1} P(n + C_2, \alpha_1, 1). \tag{56}$$

For $C_2 \le n \le N - C_1$,

$$\mu_1 \sum_{\alpha_2=0}^{1} P(n,1,\alpha_2) + \mu_2 \sum_{\alpha_1=0}^{1} P(n,\alpha_1,1) = \mu_1 \sum_{\alpha_2=0}^{1} P(n - C_1, 1, \alpha_2) + \mu_2 \sum_{\alpha_1=0}^{1} P(n + C_2, \alpha_1, 1). \tag{57}$$

For $N - C_1 + 1 \le n \le N$,

$$\mu_2 \sum_{\alpha_2=0}^{1} P(n,\alpha_1,1) = \mu_1 \sum_{\alpha_2=0}^{1} P(n - C_1, 1, \alpha_2) + \mu_2 \sum_{\alpha_1=0}^{1} P(n + C_2, \alpha_1, 1), \tag{58}$$

where $P(n,\alpha_1,\alpha_2) \equiv 0,\ for\ n < 0$ *and* $P(n,\alpha_1,\alpha_2) \equiv 0,\ for\ n > N.$

Proof:

i) For $0 \le n \le C_2 - 1$,

Adding Equations (8), (11), (14) and (17) and simplifying them gives (56).

ii) For $C_2 \le n \le N - C_1$,

Adding Equations (7), (10), (13), and (16) and simplifying them gives (57).

iii) For $N - C_1 + 1 \le n \le N$,

Adding Equations (9), (12), (15) and (18) and simplifying them gives (58).

In Theorem 2, we prove a statement of conservation of flow which generalizes that of [31] for single-item systems. Due to the complexity of the range of buffer levels and the existence of batch transition terms in balance equation, it is not easy to use induction as was done earlier. We therefore present a new proof of conservation of flow in the following theorem.

Theorem 2 *Conservation of flow:*
Since there is no creation or destruction of batches, flow is conserved. That is,

$$P = \mu_1 C_1 \sum_{n=0}^{N-C_1} \sum_{\alpha_2=0}^{1} P(n,1,\alpha_2) = \mu_2 C_2 \sum_{n=C_2}^{N} \sum_{\alpha_1=0}^{1} P(n,\alpha_1,1), \tag{59}$$

which can also be written

$$P = \mu_1 C_1 E_1 = \mu_2 C_2 E_2. \tag{60}$$

Proof: Multiplying (56), (57), and (58) by n and summing over n, we get

$$\begin{aligned} &\mu_1 \sum_{n=0}^{N-C_1} \sum_{\alpha_2=0}^{1} nP(n,1,\alpha_2) + \mu_2 \sum_{n=C_2}^{N} \sum_{\alpha_1=0}^{1} nP(n,\alpha_1,1) \\ &\qquad = \mu_1 \sum_{n=0}^{N} \sum_{\alpha_2=0}^{1} nP(n-C_1,1,\alpha_2) + \mu_2 \sum_{n=0}^{N} \sum_{\alpha_1=0}^{1} nP(n+C_2,\alpha_1,1), \end{aligned} \tag{61}$$

from which we get

$$\begin{aligned} &\mu_1 \sum_{n=0}^{N-C_1} \sum_{\alpha_1=0}^{1} nP(n,1,\alpha_2) + \mu_2 \sum_{n=C_2}^{N} \sum_{\alpha_1=0}^{1} nP(n,\alpha_1,1) \\ &\quad = \mu_1 \sum_{n=0}^{N} \sum_{\alpha_2=0}^{1} (n-C_1)P(n-C_1,1,\alpha_2) + \mu_2 \sum_{n=0}^{N} \sum_{\alpha_1=0}^{1} (n+C_2)P(n+C_2,\alpha_1,1) \\ &\quad +\mu_1 C_1 \sum_{n=0}^{N} \sum_{\alpha_2=0}^{1} P(n-C_1,1,\alpha_2) - \mu_2 C_2 \sum_{n=0}^{N} \sum_{\alpha_1=0}^{1} P(n+C_2,\alpha_1,1). \end{aligned} \tag{62}$$

In (62), it can be seen that the terms

$$\mu_1 \sum_{n=0}^{N} \sum_{\alpha_1=0}^{1} (n-C_1)P(n-C_1,1,\alpha_2),$$

$$\mu_2 \sum_{n=0}^{N} \sum_{\alpha_1=0}^{1} (n+C_2)P(n+C_2,\alpha_1,1),$$

$$\mu_1 C_1 \sum_{n=0}^{N} \sum_{\alpha_2=0}^{1} P(n - C_1, 1, \alpha_2),$$

and

$$\mu_2 C_2 \sum_{n=0}^{N} \sum_{\alpha_2=0}^{1} P(n + C_2, \alpha_1, 1)$$

can be simplified as follows:

$$\mu_1 \sum_{n=0}^{N} \sum_{\alpha_2=0}^{1} (n - C_1) P(n - C_1, 1, \alpha_2) = \mu_1 \sum_{n=C_1}^{N} \sum_{\alpha_2=0}^{1} (n - C_1) P(n - C_1, 1, \alpha_2) \tag{63}$$

because $P(n, 1, 0) = P(n, 1, 1) = 0,\ for\ n < 0$.

$$\mu_2 \sum_{n=0}^{N} \sum_{\alpha_1=0}^{1} (n + C_2) P(n + C_2, \alpha_1, 1) = \mu_2 \sum_{n=0}^{N-C_2} \sum_{\alpha_1=0}^{1} (n + C_2) P(n + C_2, \alpha_1, 1) \tag{64}$$

because $P(n, 0, 1) = P(n, 1, 1) = 0,\ for\ n > N$.

$$\mu_1 C_1 \sum_{n=0}^{N} \sum_{\alpha_2=0}^{1} P(n - C_1, 1, \alpha_2) = \mu_1 C_1 \sum_{n=C_1}^{N} \sum_{\alpha_2=0}^{1} P(n - C_1, 1, \alpha_2) \tag{65}$$

because $P(n, 1, 0) = P(n, 1, 1) = 0,\ for\ n < 0$.

$$\mu_2 C_2 \sum_{n=0}^{N} \sum_{\alpha_1=0}^{1} P(n + C_2, \alpha_1, 1) = \mu_2 C_2 \sum_{n=0}^{N-C_2} \sum_{\alpha_1=0}^{1} P(n + C_2, \alpha_1, 1). \tag{66}$$

because $P(n, 0, 1) = P(n, 1, 1) = 0,\ for\ n > N$.

Incorporating (63)–(66) into (62), we get

$$\begin{aligned}
&\mu_1 \sum_{n=0}^{N-C_1} \sum_{\alpha_2=0}^{1} nP(n, 1, \alpha_2) + \mu_2 \sum_{n=C_2}^{N} \sum_{\alpha_1=0}^{1} nP(n, \alpha_1, 1) \\
&= \mu_1 \sum_{n=C_1}^{N} \sum_{\alpha_1=0}^{1} (n - C_1) P(n - C_1, 1, \alpha_2) + \mu_2 \sum_{n=0}^{N-C_2} \sum_{\alpha_2=0}^{1} (n + C_2) P(n + C_2, \alpha_1, 1) \\
&+ \mu_1 C_1 \sum_{n=C_1}^{N} \sum_{\alpha_1=0}^{1} P(n - C_1, \alpha_1, 1) - \mu_2 C_2 \sum_{n=0}^{N-C_2} \sum_{\alpha_2=0}^{1} P(n + C_2, 1, \alpha_2).
\end{aligned} \tag{67}$$

In (67), it can be seen that

$$\mu_1 \sum_{n=0}^{N-C_1} \sum_{\alpha_2=0}^{1} nP(n, 1, \alpha_2) = \mu_1 \sum_{n=C_1}^{N} \sum_{\alpha_2=0}^{1} (n - C_1) P(n - C_1, 1, \alpha_2). \tag{68}$$

$$\mu_2 \sum_{n=C_2}^{N} \sum_{\alpha_1=0}^{1} nP(n, \alpha_1, 1) = \mu_2 \sum_{n=0}^{N-C_2} \sum_{\alpha_1=0}^{1} (n + C_2)P(n + C_2, \alpha_1, 1). \tag{69}$$

$$\mu_1 C_1 \sum_{n=C_1}^{N} \sum_{\alpha_1=0}^{1} P(n - C_1, \alpha_1, 1) = \mu_1 C_1 E_1. \tag{70}$$

$$\mu_2 C_2 \sum_{n=0}^{N-C_2} \sum_{\alpha_1=0}^{1} P(n + C_2, 1, \alpha_2) = \mu_2 C_2 E_2. \tag{71}$$

Incorporating (68)–(71) into (67), we get

$$0 = \mu_1 C_1 E_1 - \mu_2 C_2 E_2, \tag{72}$$

from which

$$\mu_1 C_1 E_1 = \mu_2 C_2 E_2. \tag{73}$$

This completes the proof.

Remark 2 *An interpretation of conservation of flow based on Little's law.*

If we treat Machine 2 as a system, it satisfies the assumptions of Little's law (Little [46]). Under assumption (9) in Section 2, it can be seen that $\mu_1 C_1 E_1$ is the effective arrival rate of parts (i.e., the arrival rate of parts that are accepted by Machine 2). Also the mean number of parts in Machine 2 is given by $E_2 C_2 + (1 - E_2)0 = E_2 C_2$. Under the assumptions in Section 2, the mean sojourn time in Machine 2 of a part is given by $\frac{1}{\mu_2}$. Finally, applying Little's law [46] to Machine 2, we get

$$L = \lambda W \Leftrightarrow E_2 C_2 = \mu_1 C_1 E_1 \cdot \frac{1}{\mu_2}, \tag{74}$$

from which we get the new generalized conservation of flow.

4.5 Equivalence property

In the following theorem, we demonstrate an equivalence property.

Theorem 3 *Equivalence property*

Consider the following two lines with two unreliable batch machines and a finite buffer:

Line 1

The parameters are given by N, C_1, C_2, p_1, p_2, r_1, r_2, μ_1, and μ_2. The performance measures of interest are given by P_1, P_2, E_1, E_2, P_B, P_S, and $\bar{n}$.

Line 2

The parameters are given by N, C_1^*, C_2^*, p_1^*, p_2^*, r_1^*, r_2^*, μ_1^*, *and* μ_2^*, *where* C_1^*, C_2^*, p_1^*, p_2^*, r_1^*, r_2^*, μ_1^*, *and* μ_2^* *satisfy* $C_1^* = C_2$, $C_2^* = C_1$, $p_1^* = p_2$, $p_2^* = p_1$, $r_1^* = r_2$, $r_2^* = r_1$, $\mu_1^* = \mu_2$, *and* $\mu_2^* = \mu_1$. *The performance measures are given by* P_1^*, P_2^*, E_1^*, E_2^*, P_B^*, P_S^*, *and* $\bar{n}^*$.

Then, we have the following equivalence relationships:
$P_1 = P_2 = P_1^* = P_2^*$, $E_1 = E_2^*$, $E_2 = E_1^*$, $P_B = P_S^*$, $P_S = P_B^*$, *and* $\bar{n} = N - \bar{n}^*$.

Proof:

Let $P(i, \alpha_1, \alpha_2)$ be the probability distribution of the state (i, α_1, α_2) of the line 1, where i $(0 \leq i \leq N)$ denotes the number of parts in the buffer, α_1 denotes the repair state of Machine 1, and α_2 denotes the repair state of Machine 2 in line 1. We also define $P^*(j, \alpha_1^*, \alpha_2^*)$ as the probability distribution of the state $(j, \alpha_1^*, \alpha_2^*)$ of line 2, where j $(0 \leq i \leq N)$ denotes the number of parts in the buffer, α_1^* denotes the repair state of Machine 1, and α_2^* denotes the repair state of Machine 2 in line 2.

Following arguments similar to those of Section 3, we can derive a set of balance equations for line 2. For example, the balance equation of line 2 corresponding to the equation (7) in Section 3.2 is:

$$P^*(n,0,0)(r_2 + r_1) = P^*(n,1,0)p_2 + P^*(n,0,1)p_1, \text{ for } C_1 \leq n \leq N - C_2. \tag{75}$$

By comparing two sets of balance equations, it can be shown that

$$P(n, \alpha_1, \alpha_2) = P^*(N - n, \alpha_1^*, \alpha_2^*), \quad \text{for} \quad 0 \leq n \leq N. \tag{76}$$

Using (76) and the definitions of the performance measures of line 2, we get the following:

$$E_1^* = \sum_{n=0}^{N-C_1^*} \sum_{\alpha_2^*=0}^{1} P^*(n, 1, \alpha_2^*) = \sum_{n=0}^{N-C_2} \sum_{\alpha_1=0}^{1} P(N-n, \alpha_1, 1) = \sum_{m=C_2}^{N} \sum_{\alpha_1=0}^{1} P(m, \alpha_1, 1) = E_2. \tag{77}$$

$$E_2^* = \sum_{n=C_2^*}^{N} \sum_{\alpha_1^*=0}^{1} P^*(n, \alpha_1^*, 1) = \sum_{n=C_1}^{N} \sum_{\alpha_2=0}^{1} P(N-n, 1, \alpha_2) = \sum_{m=0}^{N-C_1} \sum_{\alpha_2=0}^{1} P(m, 1, \alpha_2) = E_1. \tag{78}$$

$$P_1^* = \mu_1^* C_1^* E_1^* = \mu_2 C_2 E_2 = P_2, \; P_2^* = \mu_2^* C_2^* E_2^* = \mu_1 C_1 E_1 = P_1. \tag{79}$$

$$P_B^* = \sum_{n=N-C_1^*+1}^{N} \sum_{\alpha_2^*=0}^{1} P^*(n, 1, \alpha_2^*) = \sum_{n=N-C_2+1}^{N} \sum_{\alpha_1=0}^{1} P(N-n, \alpha_1, 1) = \sum_{m=0}^{C_2-1} \sum_{\alpha_1=0}^{1} P(m, \alpha_1, 1) = P_S. \tag{80}$$

$$P_S^* = \sum_{n=0}^{C_2^*-1} \sum_{\alpha_1^*=0}^{1} P^*(n, \alpha_1^*, 1) = \sum_{n=0}^{C_1-1} \sum_{\alpha_2=0}^{1} P(n, 1, \alpha_2) = \sum_{m=N-C_1+1}^{N} \sum_{\alpha_1=0}^{1} P(m, 1, \alpha_2) = P_B. \tag{81}$$

$$\bar{n}^* = \sum_{n=0}^{N} \sum_{\alpha_1^*=0}^{1} \sum_{\alpha_2^*=0}^{1} nP^*(n, \alpha_1^*, \alpha_2^*) = N - \sum_{m=0}^{N} \sum_{\alpha_1=0}^{1} \sum_{\alpha_2=0}^{1} mP(m, \alpha_1, \alpha_2) = N - \bar{n}^*. \tag{82}$$

This completes the proof.

5 Computing performance measures when the system is not ergodic

The theoretical results and computational method of this paper are not applicable to non-ergodic systems. Remark 1 of Section 3.1 shows that when the system is not ergodic, ie, when h, the greatest common divisor of C_1 and C_2, is greater than 1, there is a transformed system whose behavior is identical with that of the original system, and which is ergodic. In this section, we show how to use the transformed system to obtain performance measures for the original system.

Assume that all the lines considered in this section have the same machine transition rate parameters, μ_1, r_1, p_1, μ_2, r_2, p_2. Define $P(N, C_1, C_2; n(0))$ to be the production rate of the two-machine line with those rates, with machine sizes C_1 and C_2, buffer size N, and initial buffer level $n(0)$. Define $\bar{n}(N, C_1, C_2; n(0))$ to be the average inventory of that system. When the system is ergodic, we will suppress the last argument.

5.1 Production rate

The production rate is given by

$$P(N, C_1, C_2; n(0)) = hP(M, C_1/h, C_2/h) \tag{83}$$

where M and e are defined in Remark 1 of Section 3.1. The factor of h comes from the fact that the transformed system is producing batches of parts, where each part is itself a set of h parts of the original system.

Define e^* to be the unique integer such that $0 \leq e^* < h$ and and h is a divisor of $N - e^*$. There is a unique integer M^* that satisfies

$$N = M^*h + e^* \tag{84}$$

Comparing (5) and (84), we have

$$M^*h + e^* \geq Mh + e$$

or

$$(M^* - M)h \geq e - e^*$$

If $e - e^* \geq 0$, then this inequality is satisfied only by $M = M^* - 1$. If $e - e^* < 0$, then this inequality is satisfied only by $M = M^*$. Consequently, there are two possible values for the production rate determined in (83): $hP(M^*, C_1/h, C_2/h)$ and $hP(M^* - 1, C_1/h, C_2/h)$.

5.2 Average inventory

The average inventory is given by

$$\bar{n}(N, C_1, C_2; n(0)) = h\bar{n}(M, C_1/h, C_2/h) + e \tag{85}$$

Note that e can have h different values, depending on $n(0)$. Therefore $\bar{n}(N, C_1, C_2; n(0))$ could have h different values for different $n(0)$.

5.3 Example

In this section, we provide an example to show how to compute the performance measures of this system exactly when it is not ergodic.

Let $N = 60$ and C_2=11. (This example is used in Section 7.2.) The system is not ergodic when C_1 =11, 22, 33, 44, or 55 since C_1 and C_2 are not relatively prime at these points (Section 3.1). We present our method for calculating the performance measures only when C_1 =11. The cases for C_1 = 22, 33, 44, and 55 can be handled similarly. We verified our results using simulation, which is not described here.

In this case, $h = 11$. From (84), $M^* = 5$ and $e^* = 5$.

When the initial buffer level is a multiple of 11, the buffer level is only allowed to be a multiple of 11 at any time during the evolution of this system. This is because the buffer level can only increase by 11 and it can only decrease by 11.

In general

1. When the initial buffer level $n(0)$ is given by 0 or 11 or 22 or 33 or 44 or 55, the set of all possible buffer levels during the evolution of this model is given by {0, 11, 22, 33, 44, 55}. From (3) and (4), $e = 0$ and $M = 5$.

2. When the initial buffer level $n(0)$ is given by 1 or 12 or 23 or 34 or 45 or 56, the set of all possible buffer levels during the evolution of this model is given by $\{1, 12, 23, 34, 45, 56\}$. $e = 1$ and $M = 5$.

3. When the initial buffer level $n(0)$ is given by 2 or 13 or 24 or 35 or 46 or 57, the set of all possible buffer levels during the evolution of this model is given by $\{2, 13, 24, 35, 46, 57\}$. $e = 2$ and $M = 5$.

4. When the initial buffer level $n(0)$ is given by 3 or 14 or 25 or 36 or 47 or 58, the set of all possible buffer levels during the evolution of this model is given by $\{3, 14, 25, 36, 47, 58\}$. $e = 3$ and $M = 5$.

5. When the initial buffer level $n(0)$ is given by 4 or 15 or 26 or 37 or 48 or 59, the set of all possible buffer levels during the evolution of this model is given by $\{4, 15, 26, 37, 48, 59\}$. $e = 4$ and $M = 5$.

6. When the initial buffer level $n(0)$ is given by 5 or 16 or 27 or 38 or 49 or 60, the set of all possible buffer levels during the evolution of this model is given by $\{5, 16, 27, 38, 49, 60\}$. $e = 5$ and $M = 5$.

7. When the initial buffer level $n(0)$ is given by 6 or 17 or 28 or 39 or 50, the set of all possible buffer levels during the evolution of this model is given by $\{6, 17, 28, 39, 50\}$. $e = 6$ and $M = 4$.

8. When the initial buffer level $n(0)$ is given by 7 or 18 or 29 or 40 or 51, the set of all possible buffer levels during the evolution of this model is given by $\{7, 18, 29, 40, 51\}$. $e = 7$ and $M = 4$.

9. When the initial buffer level $n(0)$ is given by 8 or 19 or 30 or 41 or 52, the set of all possible buffer levels during the evolution of this model is given by $\{8, 19, 30, 41, 52\}$. $e = 8$ and $M = 4$.

10. When the initial buffer level $n(0)$ is given by 9 or 20 or 31 or 42 or 53, the set of all possible buffer levels during the evolution of this model is given by $\{9, 20, 31, 42, 53\}$. $e = 9$ and $M = 4$.

11. When the initial buffer level $n(0)$ is given by 10 or 21 or 32 or 43 or 54, the set of all possible buffer levels during the evolution of this model is given by $\{10, 21, 32, 43, 54\}$. $e = 10$ and $M = 4$.

The total number of all possible buffer levels in Cases 1–6 is given by 6. The total number of all possible buffer levels in Cases 7–11 is given by 5. Cases 1–6 behave exactly like a two-machine, one-buffer single-item line with buffer size 5 in which the part is a batch of 11 parts; and Cases 7–11 are like such a line with buffer size 4. From this,

Production rate in Case 1 = Production rate in Case 2 = ... = Production rate in Case 5 = Production rate in Case 6 > Production rate in Case 7 = Production rate in Case 8 = ... = Production rate in Case 10 = Production rate in Case 11.

Thus, depending on the total number of all possible buffer levels, we have two groups. For each case above, we can transform the original line into a corresponding reduced line using the transformation presented in Section 3.1. Then, we can compute the exact performance measures of the original line using the results of corresponding reduced lines as follows:

(1) Production rate of original line

- For cases 1–6:

  We first calculate the production rate of a line with C_1 =1, C_2 =1, N=5, and the other parameters $(r_1, r_2, \mu_1, \mu_2, p_1, p_2)$ exactly the same as those in the original line. We then multiply it by 11 to obtain the corresponding production rate of the original line.

- For cases 7–11:

  We first calculate the production rate of a line with C_1 =1, C_2 =1, N=4, and the other parameters $(r_1, r_2, \mu_1, \mu_2, p_1, p_2)$ are exactly the same as those in the original line. We then multiply it by 11 to obtain the corresponding production rate of the original line.

  Thus depending on two groups, production rate can take two values when $C_1 = 11$. Readers are referred to Figure 6 of Section 7.

(2) Expected in-process inventory in the buffer of original line

- For cases 1-6:

  Using an argument similar to the one used in calculating production rates, we can calculate the expected in-process inventory in the buffer of original line using that of the reduced line with C_1 =1, C_2 =1, N=5, and the other parameters $(r_1, r_2, \mu_1, \mu_2, p_1, p_2)$ are exactly the same as those in the original line:

$$\bar{n} = 11\bar{n}(5, 1, 1) + e \tag{86}$$

- For cases 7-11:

  Using an argument similar to the one used in calculating production rates, we can calculate the expected in-process inventory in the buffer the original line using that of the reduced line with C_1 =1, C_2 =1, N=4, and the other parameters $(r_1, r_2, \mu_1, \mu_2, p_1, p_2)$ are exactly the same as those in the original line:

$$\bar{n} = 11\bar{n}(4, 1, 1) + e \tag{87}$$

Thus depending on $n(0)$, the expected in-process inventory can take one of 11 different values when $C_1 = 11$. Readers are referred to Figure 10 of Section 7.

6 Computational scheme

In this section, we briefly comment on a computational scheme to compute stationary probabilities when the system is ergodic. Non-ergodic cases are treated by finding an associated ergodic system, as described in the previous section.

It is not clear how to solve the balance equations using a technique like that presented in pp. 105–111 of [31]. This is because of the batch transition terms and the complexity of the ranges of buffer levels. Although we can insert a guess of a solution (the product form) into the balance equations, the resulting equations are so messy and complex (depending on the relationship between C_1, C_2, and N) that we no longer expect to find a solution of this type. Instead, we use an efficient numerical technique to solve the balanced equations and the normalization condition.

Step 1 We write the balance equations and normalization condition into the following form:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{88}$$

where $\mathbf{A}$ and $\mathbf{b}$ denote the matrix and column vector constructed from the balance equations and the normalization condition, and $\mathbf{x}$ denotes the column vector of state probabilities.

Remark 4 When C_1 and C_2 are relatively prime, the system is ergodic (Section 3). Thus under this assumption, there exists a *unique* solution which satisfies the balance equations and the normalization condition simultaneously, which means that matrix $\mathbf{A}$ has a full column rank. Otherwise, we use the transformation of Section 3.1 and obtain multiple solutions.

Remark 5 The dimensionality of $\mathbf{x}$, and therefore the number of equations and unknowns in (88) is linear in NC_1C_2.

Step 2 When matrix $\mathbf{A}$ has a full column rank, (88) has the following unique solution [58]:

$$\mathbf{x} = \mathbf{A}^{+}\mathbf{b}, \tag{89}$$

where $\mathbf{A}^{+}$ denotes the *pseudoinverse* of $\mathbf{A}$, and it is given by

$$\mathbf{A}^{+} = (\mathbf{A}^{T}\mathbf{A})^{-1}\mathbf{A}^{T}. \tag{90}$$

Remark 6. The best way to compute $\mathbf{A}^{+}$ is to use *singular value decomposition* ([58] and [47]). For this purpose, we first decompose the matrix $\mathbf{A}$ into the following form:

$$\mathbf{A} = \mathbf{USV}^{T}, \tag{91}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, and $\mathbf{S}$ is a diagonal matrix with real, non-negative singular values ([58], p. 354). We then insert (86) into (87), and get the following expression for $\mathbf{A}^{+}$:

$$\mathbf{A}^{+} = \mathbf{V}(\mathbf{S}^{T}\mathbf{S})^{-1}\mathbf{S}^{T}\mathbf{U}^{T}. \tag{92}$$

Remark 7 The primary advantage of singular value decomposition in solving simultaneous equations is that it can be applied to any matrix A (even one that is ill-conditioned), producing stable solutions without accumulating rounding errors [35].

Remark 8 Based on the scheme presented in this paper, we wrote MATLAB code. The numerical results in all figures presented in the next section were obtained very quickly. We encountered no problem in producing the numerical results even with parameters (e.g., N=120, $C_1 = 50, C_2 = 59$) that lead to very large $\mathbf{A}$ matrices. See Section 7, for example.

7 Sample numerical results and qualitative observations

We have confirmed all the theoretical results in Section 4 via extensive numerical results.

7.1 Numerical results: Effect of buffer size on performance measures

In this section, we present some numerical results. In Figures 2–5, we set $C_1 = 2$, $p_1 = 0.01$, $p_2 = 0.009$, $r_1 = 0.09$, $r_2 = 0.08$, $\mu_1 = 1.1$, $\mu_2 = 1.0$, and we varied C_2 and N.

In Figures 2–5, Machine 2 is the bottleneck when $C_2 = 1$, and Machine 1 is the bottleneck when $C_2 \geq 3$. Figure 2 shows that the production rate tends to increase and saturate as the buffer size

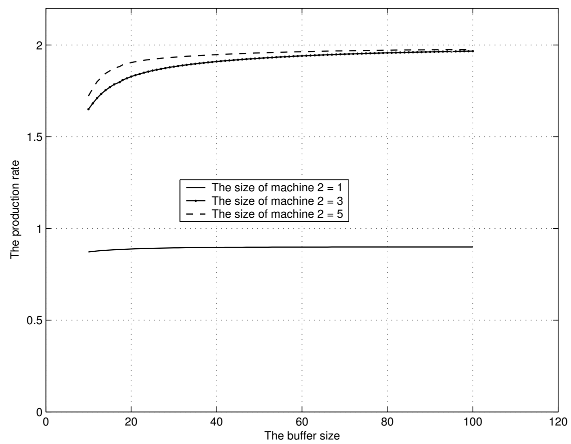

Figure 2: The effect of buffer size on the production rate.

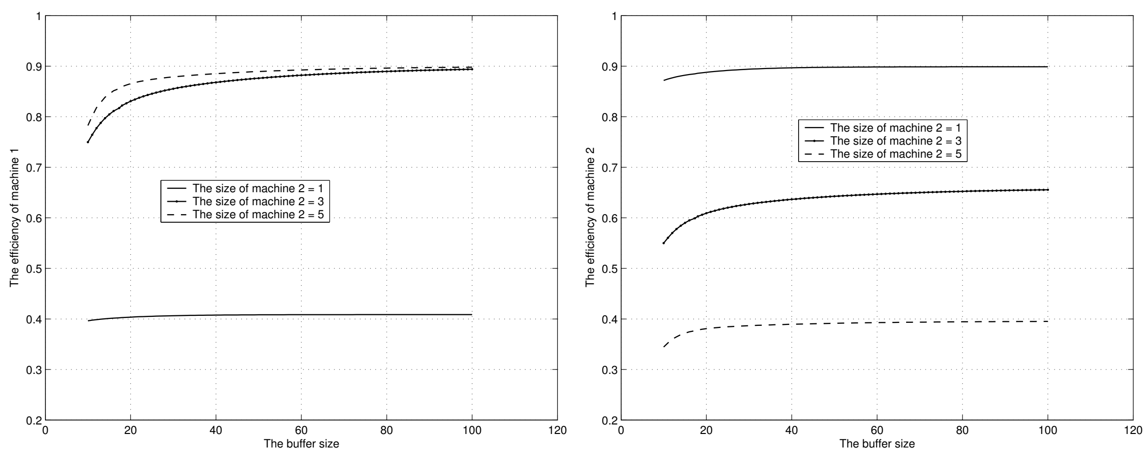

Figure 3: The effect of buffer size on the efficiencies of Machine 1 and Machine 2.

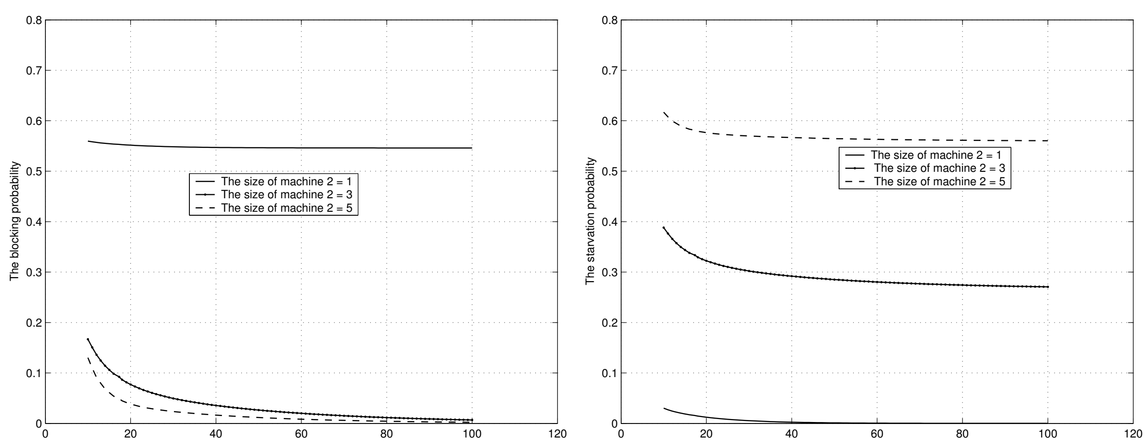

Figure 4: The effect of buffer size on the blocking and starvation probabilities.

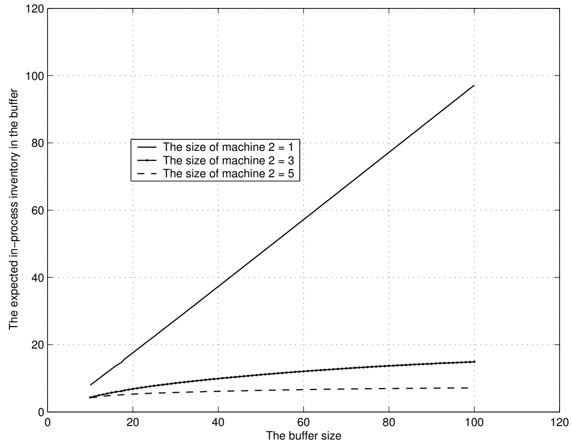

Figure 5: The effect of buffer size on the expected in-process inventory in the buffer.

increases. Figure 3 shows that the efficiencies of Machine 1 and Machine 2 tend to increase and saturate as the buffer size increases. Figure 4 shows that the blocking probability of Machine 1 and the starvation probability of Machine 2 tend to decrease and saturate as the buffer size increases. Figure 5 shows that the expected in-process inventory tends to increases as the buffer size increases.

7.2 Numerical results and qualitative interpretations: Effect of machine sizes on performance measures

In this section, we evaluate the example described in Section 5. We set $N = 60$, C_2=11, $p_1 = 0.01$, $p_2 = 0.009$, $r_1 = 0.09$, r_2= 0.08, $\mu_1 = 1.1$, $\mu_2 = 1.0$, and we vary C_1. Machine 1 is a bottleneck for $1 \le C_1 \le 9$, Machine 2 is a bottleneck for $10 \le C_1 \le 50$.

For $51 \le C_1 \le 53$, the system is in deadlock. This is because the condition $C_1 + C_2 \le N + 1$ is violated for $C_1 \ge 51$. In Figures 6–10, the production rate and efficiencies are zero, and the blocking and starvation probabilities are 1, for $C_1 \ge 51$.

The system is not ergodic when $C_1 = 11$, 22, 33, or 44. In each of those cases, the performance measures can take multiple values. The actual value depends on the initial buffer level. As mentioned in Section 5, the line has two possible production rates when C_1 is 11. These rates are 11 times those of lines with two non-batch machines and one buffer in which the buffer size is either 5 or 4 and the other parameters $(r_1, r_2, \mu_1, \mu_2, p_1, p_2)$ are the same as those in the original line. The expected in-process inventories when C_1 is 11 are calculated from equations (86) and (87) in Section 5. Depending on $n(0)$ in (86) and (87), the expected-in-process inventory can take 11 different values at non-ergodic points. The other cases can be handled similarly.

Figure 6 shows the effect of the size of Machine 1 on the production rate of the line. When Machine 1 is the bottleneck, the larger C_1, the larger the production rate. When Machine 2 is the bottleneck, however, the effect of increasing C_1 on the production rate is more complicated. When Machine 2 becomes the bottleneck, the continued increase of the size of Machine 1 leads to the increase of the production rate at a much slower rate until $C_1 = 28$. For $C_1 > 28$, the production rate tends to decrease slowly until it suddenly drops to zero due to deadlock. This decrease does not occur in systems with batches of size 1. Why does the production rate of the line tend to decrease after a certain point?

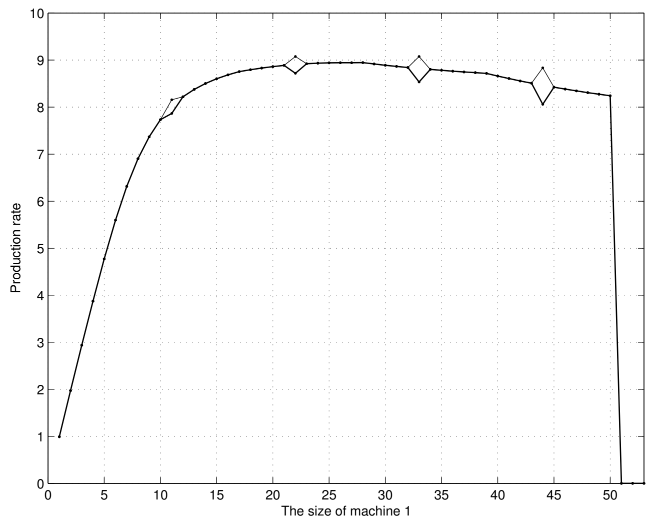

Figure 6: The effect of the size of machine 1 on the production rate.

One possible explanation is as follows: Machine 1 is forced to wait as long as there is not enough space in the buffer for the C_1 parts it will produce. Thus, the larger the size of Machine 1, the higher the blocking probability of Machine 1, and the longer the periods of blocking. Also, whenever Machine 1 is blocked, Machine 2 processes the parts in the buffer in batches until it has created enough spaces in the buffer for the blocking of Machine 1 to end. Thus, the longer the duration of a blocking period of Machine 1, the smaller the mean number of parts in the buffer right after the completion of a blocking period of Machine 1, and therefore the higher the probability that Machine 2 (the bottleneck machine) starves soon after a completion of blocking of Machine 1. Therefore, the increase of the size of Machine 1 would eventually lead to a small increase of the starvation probability of the bottleneck machine and consequently a decrease of the production rate. We confirm this interpretation via exact numerical results (Figure 9).

In Figure 7, we compare the production rate of the line with that of a line in which Machine 2 and the buffer are the same as those of the original line. Machine 1 of the new line has size 1 and the same r_1 and p_1 as that of Machine 1 of the original line. The speed μ_1' is given by $C_1\mu_1$, where C_1 and μ_1 are the size and speed of Machine 1 of the original line. As a result, the isolated

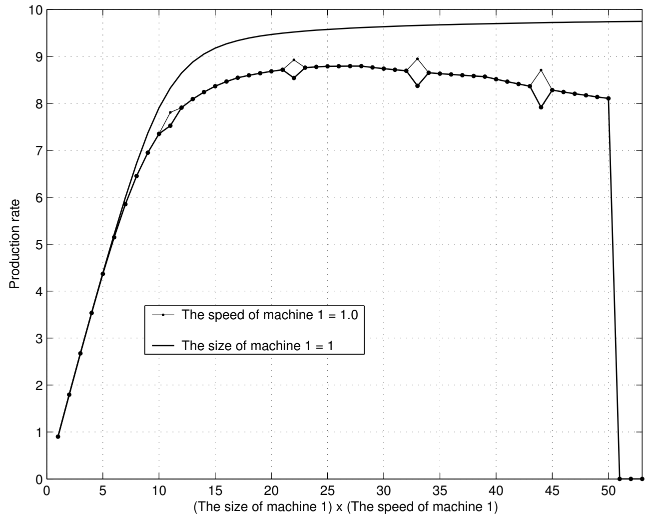

Figure 7: An approximation of the production rate

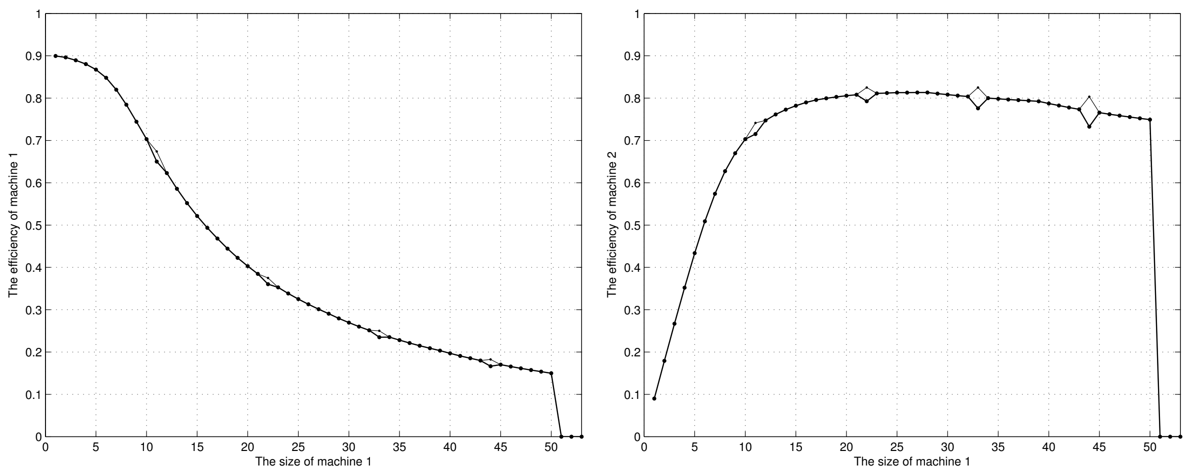

Figure 8: The effect of the size of Machine 1 on the efficiencies

production rates of the two Machines 1s are the same. Figure 7 shows that the production rate of the two lines are extremely close until $C_1 \approx 6$.

Figure 8 shows the effect of the size of Machine 1 on the efficiencies. Regardless of which machine is a bottleneck, the larger the size of Machine 1, the lower the efficiency of Machine 1, which is consistent with our comments on Figure 6. The effect of the size of Machine 1 on the efficiency of Machine 2 can similarly be explained by the comments on Figure 6.

Figure 9 presents the effect of the size of Machine 1 on the blocking and starvation probabilities. Note that blocking and starvation probabilities are negatively correlated with efficiencies of Machines 1 and 2 by Theorem 1. Thus the behavior in Figure 8 can similarly be explained by the comments presented in Figure 8.

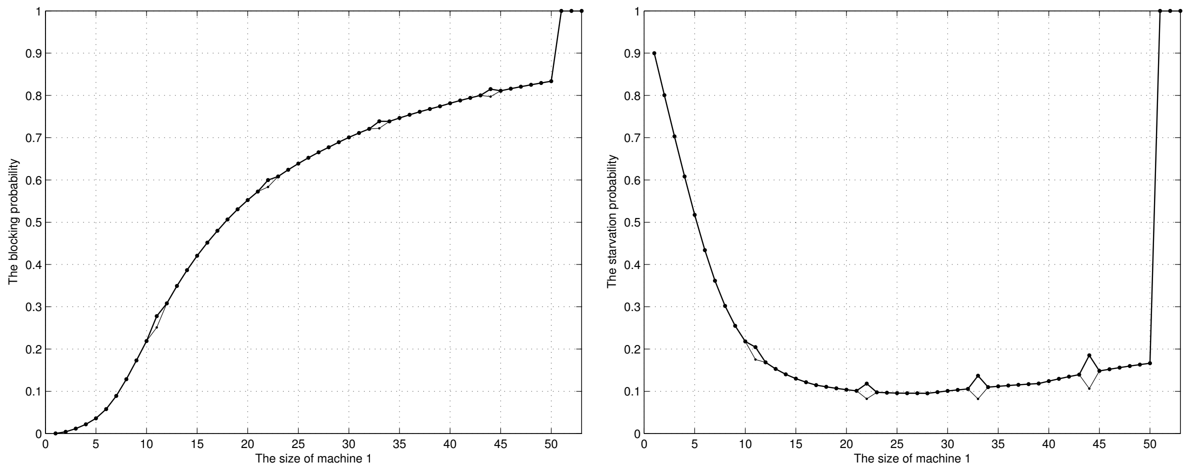

Figure 9: The effect of the size of Machine 1 on the blocking and starvation probabilities

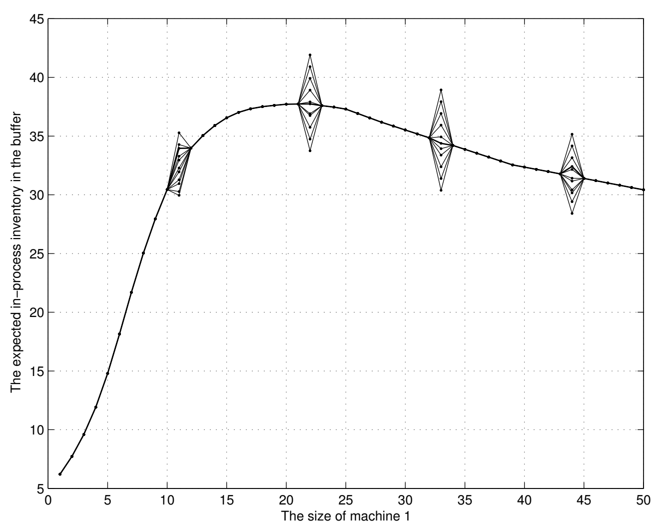

Figure 10: The effect of the size of Machine 1 on the expected in-process inventory in the buffer

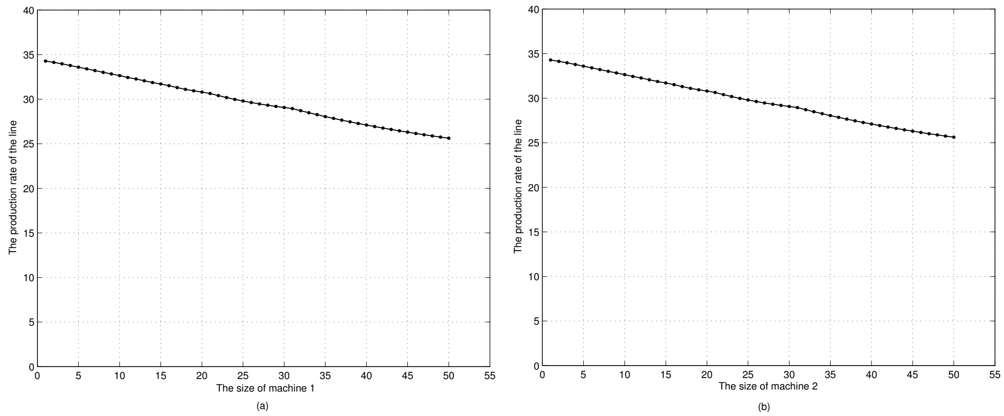

Figure 11: (a) The effect of the size of Machine 1 on the production rate of the line under the condition that $\mu_1 = 50/C_1$ (b) The effect of the size of Machine 2 on the production rate of the reversed line with $\mu_2 = 50/C_2$.

Figure 10 presents the effect of the size of Machine 1 on the expected in-process inventory in the buffer. The behavior shown in Figure 10 can be explained by the comments about Figures 7–9. The expected in-process inventories when C_1 is 11 are calculated from equations (86) and (87) in Section 5. Note that depending on $n(0)$ in (86) and (87), the expected-in-process inventory takes 11 values at non-ergodic points.

In Figures 11(a)–16(a), we set N = 120, C_2=59, p_1 = 0.009, p_2 = 0.01, r_1 = 0.08, r_2= 0.09, μ_2 = 1.0. We vary C_1 from 1 to 50 and choose $\mu_1 = 50/C_1$, so $\mu_1 C_1 = 50$. Note that the system under consideration is always ergodic in this range. Figures 11(b)–16(b) present the corresponding numerical results for the reversed system: we set N = 120, C_1=59, p_2 = 0.009, p_1 = 0.01, r_2 = 0.08, r_1= 0.09, μ_1 = 1.0. We vary C_2 from 1 to 50 and choose $\mu_2 = 50/C_2$, so $\mu_2 C_2 = 50$. Figures 11–16 imply that it is better to have a fast machine making small batches than a slow machine making large batches, if the isolated throughput $(\mu_i C_i r_i/(r_i + p_i))$ of the machine is fixed.

Figure 11(a) shows that the production rate of the line tends to decrease as we increase C_1 (and decrease μ_1). The explanation for this behavior may be the same as the explanation for Figure 6. As C_1 increases, the probability of starving Machine 2 increases. Figure 12(a) shows that the efficiency of Machine 1 tends to decrease as we increase C_1 (and decrease μ_1). Figure 13(a) shows that the efficiency of Machine 2 tends to decrease as we increase C_1 (and decrease μ_1). Figure 14(a) shows that the blocking probability tends to increase as we increase C_1 (and decrease μ_1). Figure 15(a) shows that the starvation probability tends to increase as we increase C_1 (and decrease μ_1). Figure 16(a) shows that the expected in-process inventory in the buffer tends to decrease as we increase C_1 (and decrease μ_1).

Finally, comparing Figures 11(a)–16(a) with Figures 11(b)–16(b) confirms Theorem 3.

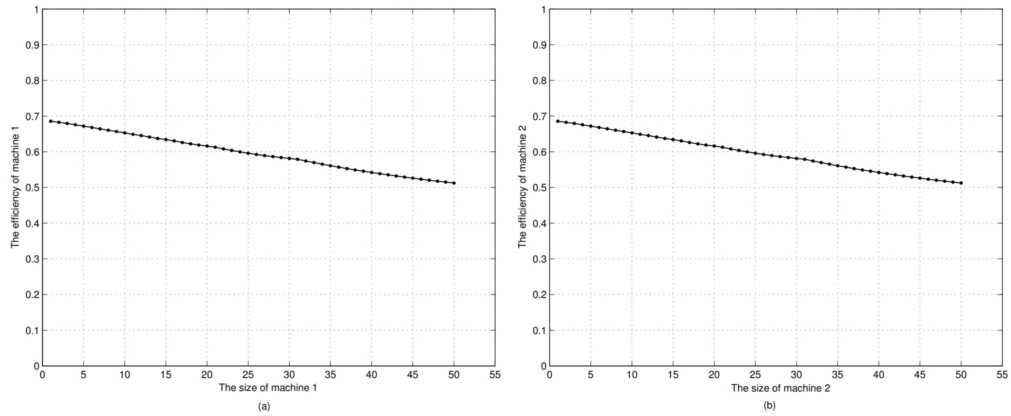

Figure 12: (a) The effect of the size of Machine 1 on the efficiency of machine 1 under the condition that $\mu_1 = 50/C_1$ (b) The effect of the size of Machine 2 on the efficiency of machine 2 of the reversed line under the condition that $\mu_2 = 50/C_2$.

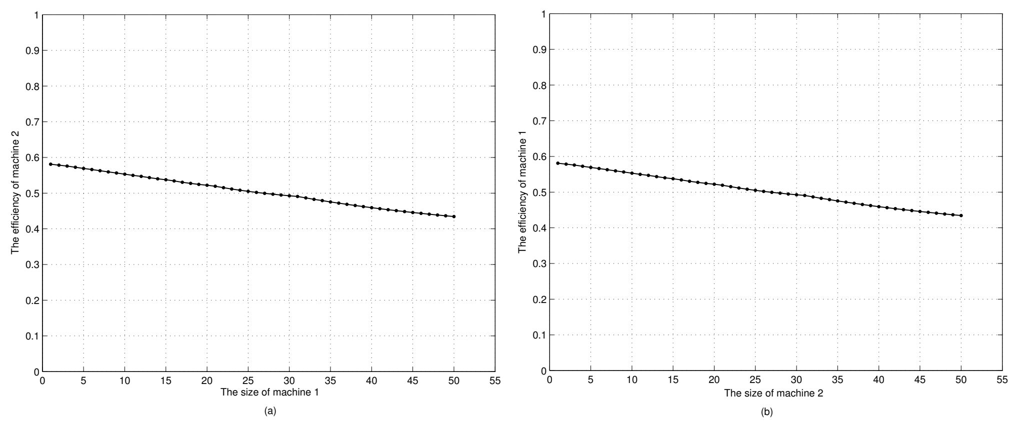

Figure 13: (a) The effect of the size of Machine 1 on the efficiency of machine 2 under the condition that $\mu_1 = 50/C_1$ (b) The effect of the size of Machine 2 on the efficiency of machine 1 of the reversed line under the condition that $\mu_2 = 50/C_2$.

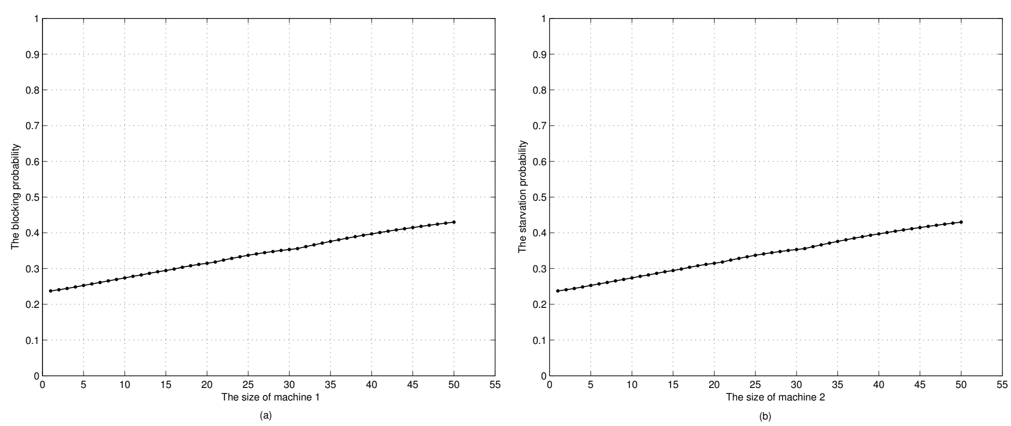

Figure 14: (a) The effect of the size of Machine 1 on the blocking probability under the condition that $\mu_1 = 50/C_1$ (b) The effect of the size of Machine 2 on the starvation probability of the reversed line under the condition that $\mu_2 = 50/C_2$.

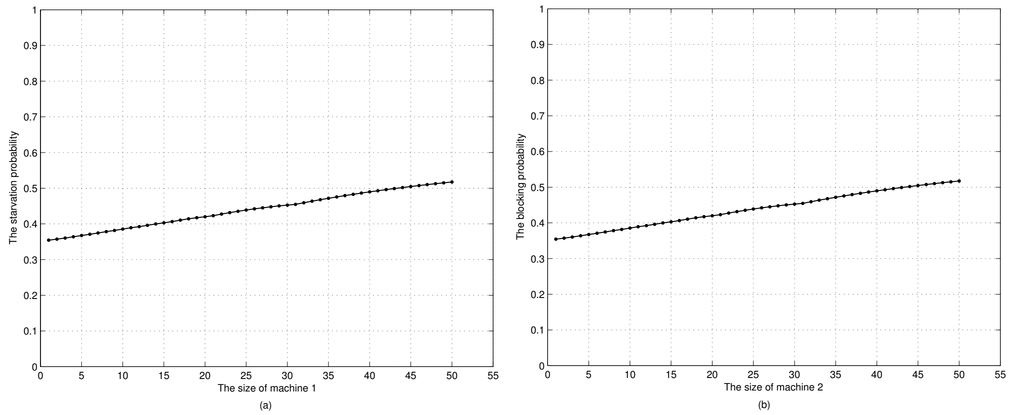

Figure 15: (a) The effect of the size of Machine 1 on the starvation probability under the condition that $\mu_1 = 50/C_1$. (b) The effect of the size of Machine 2 on the blocking probability of the reversed line under the condition that $\mu_2 = 50/C_2$.

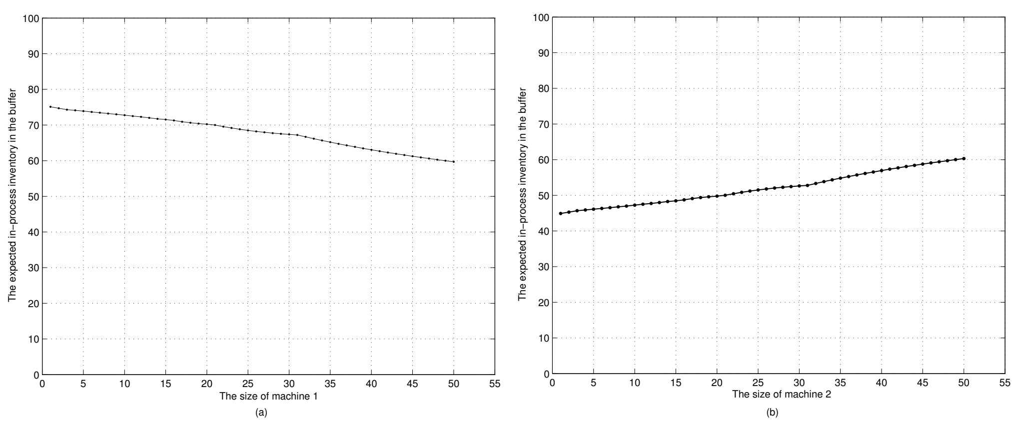

Figure 16: (a) The effect of the size of Machine 1 on the expected in-process inventory in the buffer under the condition that $\mu_1 = 50/C_1$. (b) The effect of the size of Machine 2 on the expected in-process inventory in the buffer of the reversed line under the condition that $\mu_2 = 50/C_2$.

8 Conclusions

In this paper, we have discussed the analytical modeling and exact analysis of a production line with two unreliable batch machines and a finite buffer when the machines may have different batch sizes. We have presented new conservation of flow and flow rate-idle time relationships, which generalize those in pp. 99–100 of [31] for the single-item two-machine line. We have also presented various performance measures of interest such as production rate, efficiencies of machines, probabilities of blocking and starvation and expected in-process inventory. We have established formulas for calculating the limiting values of performance measures of this system. We have demonstrated an equivalence property and described deadlock behavior. Numerical results and their qualitative interpretations have been presented. We have established relationships between the sizes of machines and performance measures of this system. New phenomena and insights are investigated and interpreted.

The important relationships between the batch sizes and some performance measures that were observed in the examples in this paper can be summarized as follows:

1. The effect of the sizes of machines on the production rate:

   1) If Machine i is a bottleneck, the larger the size of Machine i, the larger the production rate of the line.

   2) If Machine 2 is a bottleneck, and the size of Machine 1 is relatively small (e.g., less than $N/2$), the increase of the size of Machine 1 leads to the increase of the production rate up to a certain point. After that, the production rate tends to decrease slowly until it suddenly drops to zero due to deadlock. This behavior is investigated and interpreted in Section 7.

2. The effect of the sizes of machines on the blocking and starvation probabilities:

   1) Regardless of which machine is a bottleneck, the larger the size of Machine 1, the higher the blocking probability. (Similarly, regardless of which machine is a bottleneck, the larger the size of Machine 2, the higher the starvation probability).

   2) If Machine 2 is a bottleneck, and the size of Machine 1 is relatively small (e.g., less than $N/2$), an increase of the size of Machine 1 leads to a decrease of the starvation probability of Machine 1 up to a certain point. After that, the starvation probability of Machine 1 tends to increase slowly until it suddenly becomes 1 due to deadlock. This behavior is investigated and interpreted in Section 7.

3. Under the condition that isolated throughput $\frac{\mu_i C_i r_i}{r_i+p_i}$ of Machine i is fixed, the production rate of the line decreases as we increase C_i (and decrease μ_i), i=1,2. That is, it is better to have small fast machines than large slow machines. This behavior is interpreted in Section 7.

Acknowledgments

This was partially supported by the Korea Research Foundation (KRF) under Grant No. R05-2004-000-10393-0. It was also partially supported by the Singapore-MIT Alliance (SMA).

References

[1] M.H. Ammar, S.B. Gershwin, Equivalence relations in queuing models of fork/join networks, Performance Evaluation, Special Issue on Queueing Networks with Finite Capacities, 10 (1989), 233-245.

[2] F. Avram, W.M. Wein, A product design problem in semiconductor manufacturing, Operations Research 40 (1992), 986-998.

[3] J.H. Ahmadi, R.H. Ahmadi, S. Dasu, C.S. Tang, Batching and scheduling jobs on batch and discrete processors, Operations Research 40(4) (1992), 750-763.

[4] T. Altiok, Performance Analysis of Manufacturing Systems, Springer-Verlag, New York, 1997.

[5] O. Berman, Efficiency and production rate of a transfer line with two machines and a finite storage buffer, European Journal of Operations Research 9 (1982), 295-308.

[6] G.R. Bitran, D. Tirupati, Approximations for product departures from single-server station with batch processing and multi-product queues, Management Science 35(7) (1989), 851-878.

[7] J.A. Buzacott, Markov chain analysis of automatic production line with buffer stock, Ph.D. thesis, University of Birmingham, 1967.

[8] J.A. Buzacott, Methods of reliability analysis of production systems subject to breakdowns, in: Operations Research and Reliability, ed. D. Grouchko, Turin, Italy (1969), 211-232.

[9] J.A. Buzacott, J.G. Shanthikumar, Stochastic Models of Manufacturing Systems, Prentice Hall, Englewood Cliffs, New Jersey 1993.

[10] M. Burman, S.B. Gershwin, C. Suyematsu, Hewlett-Packard uses Operations Research to improve the design of a printer production line, INTERFACES 28(1), January-February, (1998), 24-36.

[11] G.P. Cachon, Exact evaluation of batch-ordering inventory policies in two-echelon supply chains with periodic review, Operations Research 49(1), (2001), 1446-1460.

[12] S.H. Chang, S.B. Gershwin, Modeling and exact analysis of a production line with unreliable batch machines and finite buffers: Part 2 – Partial batches, Working paper, Massachusetts Institute of Technology, Cambridge, MA (2005).

[13] S.H. Chang, T. Takine, Factorization and stochastic decomposition properties in bulk queues with generalized vacations, to appear in Queueing Systems 50(2-3), (2005), 165-183.

[14] F. Chen, Y. Feng, D. Simchi-Levi, Uniform distribution of inventory positions in two-echelon periodic review systems with batch-ordering policies and independent demand, European Journal of Operations Research 140(3), (2002), 648-654.

[15] H. Chen, J.M. Harrison, A. Mandelbaum, A. van Ackere, L.M. Wein, Empirical evaluation of a queueing network model for semiconductor wafer fabrication, Operations Research 36(2), (1988), 202-215.

[16] J.W. Cohen, The Single Server Queue, North Holland, Amsterdam, 1982.

[17] D.P. Connors, G.E. Feigin, D.D. Yao, A queueing network model for semiconductor manufacturing, IEEE Transactions on Semiconductor Manufacturing 9, (1996), 412-427.

[18] Y. Dallery, R. David., X.L. Xie. An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers, IIE Transactions 20, (1988), 280-283.

[19] Y. Dallery, R. David, X.L. Xie, Approximate analysis of transfer lines with unreliable machines and finite buffers, IEEE Transactions on Automatic Control 34(9), (1989), 943-953.

[20] Y. Dallery, S.B. Gershwin, Manufacturing flow line systems: A review of models and analytical results, Queueing Systems Theory and Applications, Special Issue on Queueing Models of Manufacturing Systems 12(1-2), (1992), 3-94.

[21] Y. Dallery, On modeling failure and repair times in stochastic models of manufacturing systems with generalized exponential distributions, Queueing Systems 15, (1994), 199-209.

[22] R. Deb, R.F. Serfozo, Optimal control of batch service queues, Advances in Applied Probability, 5(2), (1973), 340-361.

[23] J.H. Dshalalow, Queueing systems with state dependent parameters, In Frontiers in Queueing: Models and Applications in Science and Engineering, ed. J.H. Dshalalow, CRC press, Amsterdam, 1997, 61-116.

[24] P.J. Egbelu, Unit load design and its impact on manufacturing systems performance, J.M.A. Tanchoco (Ed.), Material Flow Systems in Manufacturing, Chapman and Hall, London, 1994.

[25] S.B. Gershwin, I.C. Schick, Modeling and analysis of two- and three-stage transfer lines with unreliable batch machines and finite buffers, Technical Report LIDS-R-979, Massachusetts Institute of Technology, Cambridge, MA (1980a).

[26] S.B. Gershwin, I.C. Schick, Continuous model of an unreliable two-stage material flow system with a finite interstage buffer, Technical Report LIDS-R-1039, Massachusetts Institute of Technology, Cambridge, MA (1980b).

[27] S.B. Gershwin, O. Berman, Analysis of production lines consisting of two unreliable machines with random processing times and a finite storage buffer, AIIE Transactions 13(1), 2-11, March (1981).

[28] S.B. Gershwin, I.C. Schick, Modeling and analysis of three-stage transfer lines with unreliable machines and finite buffers, Operations Research 31, (1983), 354-380.

[29] S.B. Gershwin, An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking, Operations Research 35(2), (1987), 291-305.

[30] S.B. Gershwin, Assembly/Disassembly systems: An efficient decomposition algorithm for tree structured networks, IIE Transactions 23(4), (1991), 302-314.

[31] S.B. Gershwin, Manufacturing Systems Engineering, Englewood Cliffs, New Jersey: PTR Prentice Hall, 1994.

[32] S.B. Gershwin, Design and operations of manufacturing system: the control-point policy, IIE Transactions 32, (2000), 891-906.

[33] S.B. Gershwin, M.H. Burman, A decomposition method for analyzing inhomogeneous assembly/disassembly systems, Annals of Operations Research 93, (2000), 91-116.

[34] C.R. Glassey, W.W. Weng, Dynamic batching heuristic for simultaneous processing, IEEE Transactions on Semiconductor Manufacturing 4(2), (1991), 77-82.

[35] G.H. Golub, C.F. van Loan, Matrix Computations, 2nd edn., The John Hopkins University Press, 1989.

[36] M.P. Groover, Automation, Production Systems, and Computer Integrated Manufacturing, Second Edition, Prentice Hall, 2001.

[37] N.G. Hall, C.N. Potts, Supply chain scheduling: batching and delivery, Operations Research, 51, (2003), 566-584.

[38] R.J. de Haut de Sigy, Loading control policy for a batch machine, Technical report LMP90-001, Massachusetts Institute of Technology, Cambridge, MA (1990).

[39] S. Helber, H. Jusi, A new decomposition approach for non-cyclic continuous material flow lines with a merging flow of material, Annals of Operations Research, 125(1-4), (2004), 117-139.

[40] D.S. Hochbaum, D. Landy, Algorithms and heuristics for scheduling semiconductor burn-in operations, Operations Research 45(6), (1997), 874-885.

[41] E. Ignall, P. Kolesar, Operating characteristics of a simple shuttle bus under local dispatching rule, Operations Research 20, (1972), 1077-1088.

[42] S.M.R. Iravani, J.A. Buzacott, M J.M. Posner, Operations and shipment scheduling of a batch on a flexible machine, Operations Research 51(4), (2003), 585-601.

[43] M.A. Jafari, J.G. Shanthikumar, Exact and approximate solutions to two-stage production lines with general uptime and downtime distributions, IIE Transactions 19(4), (1987), 412-419.

[44] V.S. Kouikoglou, Sensitivity analysis and decomposition of unreliable production lines with blocking, Annals of Operations Research 93, (2001), 245-264.

[45] R. Levantesi, A. Matta, T. Tolio, Performance evaluation of continuous production lines with machines having different processing times and multiple failure modes, Performance Evaluation 51(2-3), (2003), 247-268.

[46] J.D.C. Little, A proof for the queueing formula: $L = \lambda W$, Operations Research 9, (1961), 383-387.

[47] MATLAB, The Language of Technical Computing, The Math Works, Inc, 1999.

[48] G. D. Mostow, J. H. Sampson, and J.-P. Meyer, Fundamental Structures of Algebra, McGraw-Hill, 1963.

[49] E. J. Muth, The reversibility property of production lines, Management Science 25(2), (1979), 152-158.

[50] G.F. Newell, Approximate Behavior of Tandem Queues, Springer-Verlag, 1979.

[51] Y. Pan, S.B. Gershwin, Modeling and analysis of production lines with and without preventative maintenance, Laboratory for Information and Decision Systems Report LIDS-P-1294, Massachusetts Institute of Technology, Cambridge, MA, April, 1983.

[52] H.T. Papadopoulous, C. Heavy, J. Browne, Queueing Theory in Manufacturing System Analysis and Design, London, Chapman & Hall, 1993.

[53] A. Patchong, T. Lemoine, G. Kern, Improving car body production at PSA Peugeot Citroen, INTERFACES 33(1), Jan-Feb, (2003), 36-49.

[54] W.B. Powell, P. Humblet, The bulk service queue with a general control strategy: theoretical analysis and a new computational procedures, Operations Research 34, (1986), 267-275.

[55] SEMATECH, 2002, Semiconductor manufacturing process. July, http://www.sematech.org/public/news/mfgproc/mfgproc.htm

[56] I.C. Schick, S.B. Gershwin, Modeling and analysis of unreliable transfer lines with finite interstage buffers, In Complex Material Handling and Assembly Systems, Vol. IV, Report ESL-FR-834-6, Electronic Systems Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 1978.

[57] J.G. Shanthikumar, C.C. Tien, An algorithmic solution to two-stage production lines with possible scrapping of units, Management Science 29(9), (1983), 1069-1086.

[58] G. Strang, Introduction to Linear Algebra, Third Edition, Wellesley-Cambridge Press, Box 812060, Wellesley, MA 02482, 2003.

[59] T. Tolio, S.B. Gershwin, A. Matta, Analysis of two-machine lines with multiple failure modes, IIE Transactions 34(1), January, (2002), 51-62.

[60] M.H. Veatch, L.M. Wein, Optimal control of two-station tandem production/inventory system, Operations Research 42(2), March-April, (1994), 337-350.