ANALYSIS OF ERLANG TRANSFER LINES

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Nebahat Dönmez

February 1997

ANALYSIS OF ERLANG TRANSFER LINES

A THESIS

SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL

ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Nebahat Dönmez

February, 1997

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Cemal Dinçer(Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Halim Doğrusöz

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Erdal Erel

Approved for the Institute of Engineering and Sciences:

Prof. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

ANALYSIS OF ERLANG TRANSFER LINES

Nebahat Dönmez
M.S. in Industrial Engineering
Supervisor: Assoc. Prof. Cemal Dinçer
February, 1997

Transfer lines are widely used in the modeling and analysis of complex production systems. The literature is mostly devoted to the analysis of transfer lines with exponential processing times. However, most of the time a part is processed through stages(phases) of exponential processing times. It is possible to model such systems by means of processing times that are $k-$Erlang distributed. In the modeling and solution of these systems, significant difficulties arise due to the nature of the problem. In this thesis, we propose a Markov model for transfer lines consisting of n reliable machines with $k-$Erlang processing times and finite buffers. The arrivals to the system is Poisson distributed. A program coded in C which is capable of solving the Markov model of a three machine transfer line is also developed. Besides the commonly used performance measures, such as utilization of the machines, mean throughput, mean WIP level, we calculate the variance of WIP so that it is possible to construct confidence intervals.

Key words: Transfer Lines, Markov Models, Erlang Distribution, Variance of WIP Level.

ÖZET

K-ERLANG İŞLEM ZAMANLI SERİ AKIŞLI İMALAT SİSTEMLERİNİN ANALİZİ

Nebahat Dönmez
Endüstri Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Doç. Dr. Cemal Dinçer
Şubat, 1997

Seri akışlı imalat sistemleri karmaşık imalat sistemlerinin modellenmesinde yaygınca kullanılan alt modellerdendir. Bugüne kadar büyük çoğunlukla üssel işlem zamanlı makinelerden oluşan seri akışlı sistemler incelenmiştir. Ancak, bir çok durumda işlem zamanları üssel evrelerden oluşmaktadır. Bu tür sistemleri k-Erlang dağılımı ile modellemek mümkündür. En genel haliyle bu tür sistemlerin modellenmesi oldukça büyük zorluklar içermektedir. Bu tez çalışmasında n tane k-Erlang işlem zamanlı makineden oluşan, seri akışlı ve sonlu ara stoklu, üssel talep geliş zamanlı bir sistemin Markov modellenmesi önerilmektedir. 3 makineden oluşan sistemin Markov model çözümü için C dilinde bir program kodlanmıştır. Ayrıca, geleneksel performans ölçütlerinden; makine kullanımları, ortalama üretim miktarı, ve ortalama ara stok seviyelerinin yanı sıra, ara stok seviyesinin varyansı da hesaplanmakta, böylelikle güven aralığı hesaplamalarının yapılması mümkün kılınmaktadır.

Anahtar sözcükler: Seri Akışlı Sistemler, Markov Modeller, Erlang Dağılımı, Ara Stok Seviyesinin Varyansı

To Atatürk, and the Republic of Turkey

ACKNOWLEDGEMENT

I am mostly grateful to Associate Professor Cemal Dinçer for suggesting a research topic full of enthusiasm, and for his invaluable guidance, encouragement and the enthusiasm he inspired on me during this study.

I am indebted to Professor Halim Doğrusöz and Associate Professor Erdal Erel for showing keen interest to the subject matter and accepting to read and review this thesis. Their remarks and recommendations have been invaluable.

I would like to express my deepest gratitude and thanks to my office mates. I am especially thankful to Bahar Yetiş Kara and Muhittin Hakan Demir for their encouragement, suggestions, help, friendship, patience, understanding and moral support especially during my illness. I would also like to thank Meltem Karasu Demir and Kadri Yetiş for everything.

Finally, I have to express my sincere gratitude to my family, and to all my dear friends for sharing everything.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

In this thesis, we investigate the performance measures of a tandem queueing system with three machines with k-stage Erlang processing times, and two finite storage buffers.

A queueing system can be described as customers arriving for service, waiting for service if it is not immediately available and leaving the system after being served. Such a basic system can be schematically depicted as in Figure 1.1.

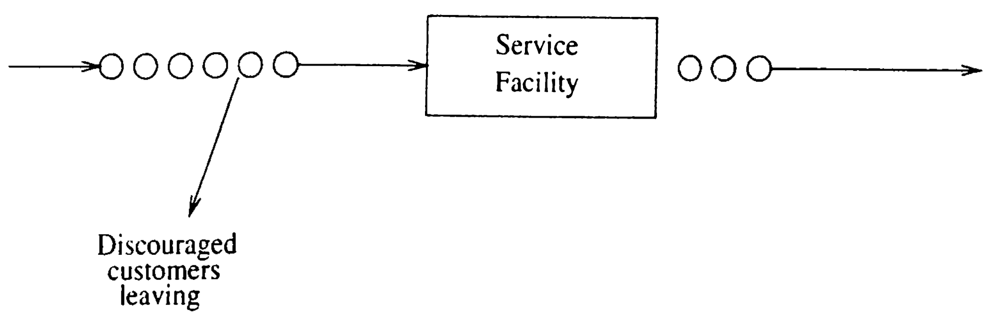

Figure 1.1: Schematic diagram of a queueing process

A *transfer line* is a manufacturing system with a very special structure. It consists of material, work stations, and storage areas. Actually, it is a linear network of service stations/machines ($M_1, M_2, ..., M_k$) separated by buffer storages ($B_1, B_2, .., B_{k-1}$). Material flows from outside the system to machine M_1 , then to buffer storage B_1, then to M_2, and so forth until it reaches machine M_k after which it leaves the system. Material visits each work station and storage area exactly once in a fixed sequence. Figure 1.2. depicts a transfer line. The squares represent machines and the triangles represent buffers.

In the language of queueing theory a transfer 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*.

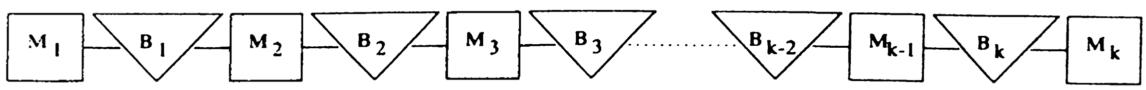

Figure 1.2: Representation of a k-machine transfer line with $k-1$ intermediate buffer storages

1.1 Related Literature

Transfer lines are studied due to their economic importance. They are used in high volume manufacturing, particularly in automobile production. In automobile production, the capital costs range from $100.000 to $30.000.000. Furthermore, transfer lines represent the simplest form of interactions of manufacturing stages, and their decoupling by means of buffers. The study of coupling and decoupling leads to application to more complex systems.

The earliest theoretical papers were published in 1950's in Russia. Vladzievskii [49] is the first author to use probability theory to explain the behaviour of automatic transfer lines. There are three major problems in the design and operation of production lines. These are the number of stages in the line, the location of buffers and the buffer sizes. Tools for the solution of these problems did not appear until the 1980's. Buzacott and Hanifin [9] describe physical and mechanical issues , such as the transfer mechanism, shunt versus series banks which determine the movement of material according to LIFO or FIFO, and the design of the line in order to reduce the cycle time. Smunt and

Perkins [46] focus on asynchronous flow lines with reliable machines. They are interested in line design, the problem of sizing and locating the buffers, and task allocations to work stations.

Transfer line systems can be classified into three categories. Synchronous systems are the systems in which operation times of the machines are assumed to be deterministic and equal, and when machines are not under repair, they start and stop at the same instant. In asynchronous systems, machines are not constrained to start or stop their operations at the same instant. Asynchronous systems are usually modeled with random operation times. Finally, continuous models treat material flow as continuous rather than discrete. The literature reviewed here and this study considers asynchronous transfer lines.

1.1.1 Flow Lines with No Intermediate Storage

For transfer lines with reliable machines, Rao [43] and Lau [29] provide explicit expressions for calculating the production rate for exponentially, Erlang, uniform, and normally distributed processing times.

Hunt [25], Hillier and Boling [23], Hildebrand [22] investigated the threee machine transfer lines with no buffer. Muth [33], Rao [44, 45] obtained numerical solutions for specific distributions of the processing times. Muth and Alkaff [35] provided a unifying solution under the assumptions that machines M_1 and M_3 have special phase-type distributions and machine M_2 has a Laplace transformable distribution.

Papadopoulos and O'Kelly [38] develop an exact procedure for the analysis of a transfer line with reliable machines where the processing times are exponentially distributed. The exact algorithm gives the marginal probability distribution of the number of units in each machine, mean queue length, and the throughput. Papadopoulos [37] also provides an algorithm for the efficient computation of the throughput rate of multistation reliable production lines with no intermediate buffers by extending the work of Muth [33].

For the case of transfer lines with no intermediate buffer and unreliable machines, Buzacott [7] obtained a formula for the production rate under deterministic processing times and general up and downtime distributions assumptions. Commault and Dallery [12] propose a method for calculating the production rate when uptimes are exponentially distributed.

1.1.2 Flow Lines with Finite Buffers

For transfer lines with reliable machines, Rao [44] analyzed two-machine transfer lines with exponential and general processing time distributions. When the processing time of the first machine is exponentially distributed and the distribution of the processing time of the second machine is general, the line is equivalent to an $M/G/1/L$ queue. This equivalance also holds for the case of a two-machine transfer line with general and exponential processing time distributions for the first and second machine, respectively [14]. If the distribution of the processing times of both machines is exponential, the analysis of the line reduces to that of an $M/M/1/L$ queue which has a simple geometric form.

For a two-machine flow line with reliable machines where the processing time distribution of each machine follows a continuous phase-type distribution, the behaviour is characterized by a discrete state, continuous time Markov process.

Altıok and Ranjan [2], Buzacott and Kostelski [10], Gun and Makowski [20] analyzed such systems using recursive and matrix geometric techniques which will be discussed later in Chapter 2.

Buzacott [8] analyzes a two-station model with identical unreliable machines and a finite buffer. He assumes that operation times and repair times are exponentially distributed whereas the probability of failure during each operation is constant. He provides an exact solution of the model using z-transforms and demonstrates that production rate is a saturating function of storage space.

Gershwin and Berman [16] study the same model except that they represent failure by an exponential distribution in time rather than a geometric distribution in the number of parts produced.

Berman [4] generalizes the work of Gershwin and Berman [16] by allowing Erlang distributed processing times.

Berg, Posner, and Zhao [3] investigate the effect of machine breakdowns on service and inventory levels and obtain the stationary distribution of the inventory process for different assumptions.

Di Mascolo, Frein, and Dallery [31] develop a general purpose analytical method for performance evaluation of multistage kanban controlled production system. Their approximation method can be extended to complex manufacturing systems with different assumptions.

Besides studies on computing the commonly used performance measures of transfer lines, recently much work is devoted to the optimal location and sizing of buffer inventories. Jensen, Pakath, and Wilson [26] develop a dynamic programming model and an efficient solution procedure to solve this problem. Lau [28] studies how an unpaced transfer line's utilization is affected by different patterns of allocating processing time variances among the stations. He shows that the results in the literature on variance allocation are ambiguous and often contradictory. He provides extensive results to demonstrate three desirable variance allocation characteristics he identifies: bowl; which indicates that the interior stations should be allocated less work than the end stations, symmetry, and spike; which suggests that the only variability can be concentrated into *only* one station and all the other stations have zero variability. Later, Pike and Martin [41] show that bowl phenomenon exists and determine optimal bowl configurations. Park [39] provides a two-phase heuristic algorithm for determining buffer sizes of production lines.

For the exact analysis of transfer lines with more than two machines, the literature is sparse. For asynchronous lines, Gershwin and Schick [17] extended their analytic solution of two-machine, finite buffer model with the assumptions that all machines are unreliable and they all have equal and constant service times.

Although exact solutions of two-machine transfer lines are avilable for a wide range of models, the work done up to now indicates that it seems hopeless to expect to obtain exact solutions of transfer lines with more machines except for some limited cases of three-machine transfer lines even when more powerful computers become available. Therefore, the use of approximate solutions are necessary to study longer lines.

Most approximate methods rely on *decomposition* where the idea is to partition the original system into a set of smaller subsystems which are easier to analyze. Decomposition methods will be presented in Chapter 2.

Altıok [1], Hillier and Boling [23], Perros and Altıok [40], Pollock, Birge and Alden [42], and Takahashi, Miyahara and Hasegawa [48] analyze flow lines with exponential processing times. In all these papers the subsystems are finite single server queues with lost arrivals and exponential processing times. Consequently, they are equivalent to a two-machine line decomposition with exponential characterization of the upstream machines.

Pollock, Birge and Alden [42], and Takahashi, Miyahara and Hasegawa [48] also consider an exponential characterization of the downstream machine.

Perros and Altıok [40] analyze transfer lines using decomposition where the downstream machines are characterized by phase-type distributions. Later, Altıok [1] extended this method to the case of transfer lines with phase-type processing time distributions.

Altıok and Ranjan [2],and Gun and Makowski [21] study decomposition methods for transfer lines where both the upstream and the downstream machines of each decomposed line is characterized by phase-type distributions.

Besides these, several authors derived simple approximation formulas for estimating the production rate of a flow line with reliable machines in which all stations are identical. Knott [27] provides an approximation formula for the case of two-machine flow lines with identical Erlang distribution. Muth [34] obtained a formula in the case of flow lines with any number of machines but no intermediate storage. Later, Blumenfeld [5] extends Muth's formula to flow lines with intermediate buffers.

Decomposition methods for flow lines with unreliable machines and reliable machines are based on the similar principles.

Gershwin [15] developed a decomposition method for synchronous transfer lines. Afterwards, Dallery, David and Xie [13] developed an algorithm, called the *DDX algorithm*, for Gershwin's decomposition technique.

For asynchronous transfer lines, Choong and Gershwin [11] extended Gershwin's decomposition technique for lines in which all machines could have different speeds, failure rates, and repair rates and all the distributions of processing times, uptimes and downtimes are assumed to be exponential.

Glassey and Hong [18] extend the work of Gershwin [15] and develop a decomposition method for an unreliable $n-$stage transfer line with $(n-1)$ interstage storage buffers. Their method is based on the examination of the $n-$stage line and the decomposed lines, and the relationship between the failure and repair rates of the individual stages and the aggregate stages. they show that their method performs better.

Springer [47] proposes a decomposition method for approximating the throughput rate and the WIP level of finite-buffered exponential queues in series. The approximation decomposes the network into individual finite-buffered queues which are linked together through a set of nonlinear equations.

All the literature review up to here concentrated on the methods to analyze the steady-state average production rates and steady-state average buffer levels of transfer lines. Yet, the *variance* of the throughput and of the buffer levels during a time period is also important.

This issue has been neglected so far. Only a few papers deal with the calculation of the variance of the behaviour of a transfer line over a limited time period. Miltenburg [32], and Lavenberg [30] treat two-machine transfer lines. They obtained results that are difficult to use and extend.

Variability issue is very important because of the fact that the *standard deviation* of production can be high. This variability is an inherent characteristic of production systems. Prediction of this variability is as important as the prediction of the mean since if both the mean and the variance are calculated, then an interval estimate for the actual throughput and the buffer levels during a period of time can be calculated.

Basically, this is the motivation for the work done in this thesis. Furthermore, as the literature review emphasizes, there are few and limited attempts to analyze the exact analytic solution techniques of transfer lines with more than two machines and finite buffers. Second chapter is devoted to the analysis and solution techniques of transfer lines with different characteristics. In the third chapter experimental results are discussed. Finally, last chapter covers the conclusion and future research.

Chapter 2

ANALYSIS OF TRANSFER LINES

2.1 Solution Techniques

2.1.1 Exact Solution Techniques

Exact analytic solutions are important because they are better than simulations or approximations when the models constructed fit real systems closely and they provide useful qualitative insight into the behaviour of the systems. Furthermore, they are the vital parts of the decomposition and aggregation methods that are described later in this chapter.

Most of the results pertaining to the exact analysis of the transfer line models are based on Markovian analysis. In order to be able to analyze the behaviour of the transfer line by a Markov process, the distributions have to be of special form, such as exponential or, more generally, continuous phase-type distributions in the case of continuous time models; geometric or more precisely discrete phase-type distributions in the case of discrete time models. Nevertheless, there are some exceptions most often encountered in transfer lines with no intermediate storage [14].

In order to be able to fully understand the discussion related to Markov processes, we are providing some information about the definitions and classifications.

A *stochastic process* is the mathematical abstraction of an empirical process governed by probabilistic laws. A stochastic process can be best defined as a set of random variables, $\{ X(t), t \in T \}$, defined over some index set or parameter space T. $X(t)$ represents the state of the process at time t and T is sometimes also called the time range of the process. The process is classified as a discrete-parameter or continuous-parameter process as follows:

(i.) If T is a countable sequence, for example,

$$T = \{0, \pm 1, \pm 2, \cdots\}$$

or

$$T = \{0, 1, 2, \ldots\},$$

then the stochastic process $\{ X(t), t \in T \}$ is said to be a discrete-parameter process defined on the index set T;

(ii.) If T is an interval or an algebraic combination of intervals, for example,

$$T = \{t : -\infty < t < +\infty\}$$

or

$$T = \{t : 0 < t < +\infty\},$$

then the stochastic process $\{ X(t), t \in T \}$ is called a continuous-parameter process defined on the index set T.

A discrete-parameter stochastic process $\{ X(t), t = 0, 1, 2, \ldots \}$ or a continuous-parameter stochastic process $\{ X(t), t > 0 \}$ is said to be a *Markov* *process* if, for any set of n time points $t_1 < t_2 < \cdots < t_n$ in the index set or time range of the process, the conditional distribution of $X(t_n)$, given the values of $X(t_1), X(t_2), X(t_3), \ldots, X(t_{n-1})$, depends only on $X(t_{n-1})$, the immediately preceding value; for any real numbers $x_1, x_2, \ldots, x_n$,

$$\Pr\{\ X(t_n) \le x_n \mid X(t_1) = x_1, \ldots, X(t_{n-1}) = x_{n-1}\ \} \\ = \Pr\{\ X(t_n) \le x_n \mid X(t_{n-1}) = x_{n-1}\ \}$$

Markov processes are classified according to the nature of the index set of the process and the nature of the state space of the process.

A real number x is said to be a state of a stochastic process $\{\ X(t), t \in T\ \}$ if there is a time point t in T such that the $P\{x - h < X(t) < x + h\}$ is positive for every $h > 0$. The state space is composed of the set of all possible states. If the state space is discrete, the Markov process is generally called a *Markov chain* . Table 2.1. summarizes our classification scheme for Markov processes.

Continuous time Markov processes are naturally obtained when all the distributions in the original model are *exponential distributions* due to the famous *lack-of-memory property* of the exponential distribution. Hence, exponential distribution has been widely used in the literature. Yet, it is not always an appropriate candidate for representing actual distributions of real life systems especially when the distributions encountered in real systems have coefficients of variation far from one which is the coefficient of variation for exponential distribution. In order to overcome this difficulty, non-exponential distributions are represented as a mixture of exponential distributions.

The simplest distribution of this form is the *Erlang distribution.* A *Hypo-Exponential* distribution of order k consists of a series of k exponential distributions with rates $\mu_1, \mu_2, \ldots, \mu_k$. Special case of Hypo-Exponential distribution is called $k-$Erlang distribution which consists of k exponential distributions with common rate $\mu_1 = \mu_2 = \cdots = \mu_k = \mu$. The random variable associated with the Erlang distribution is the sum of k independent and identical exponential random variables.

State Space	Type of Parameter: Discrete	Continuous
Discrete	Discrete-parameter Markov chain	Continuous-parameter Markov chain
Continuous	Discrete-parameter Markov process	Continuous-parameter Markov process

Table 2.1: Classification of Markov Processes

Another distribution used in the modeling of the stochastic systems is the ***Coxian distribution***. The Coxian distribution is more general than the Hypo-Exponential distribution since it also allows branching probabilities as shown in figure 2.1.

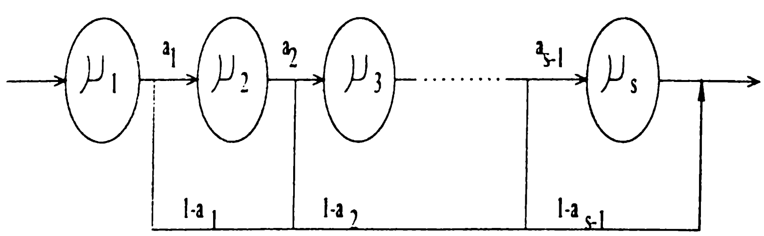

Figure 2.1: Coxian distribution with s phases

The most general form of distributions that are mixtures of exponential distributions is *phase-type distribution* . A continuous phase-type distribution with s phases(stages) is represented in Figure 2.2.

If we want to give a physical interpretation of this distribution in terms of an overall task that consists of a set of s exponential subtasks. The processing time of subtask j is exponentially distributed with rate μ_j.

The first subtask to be completed 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 $c_{0,1} + \cdots + c_{0,s} = 1$, and $c_{j,1} + \cdots + c_{j,s} + c_{j,0} = 1$.

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. The increase in the size of the state space

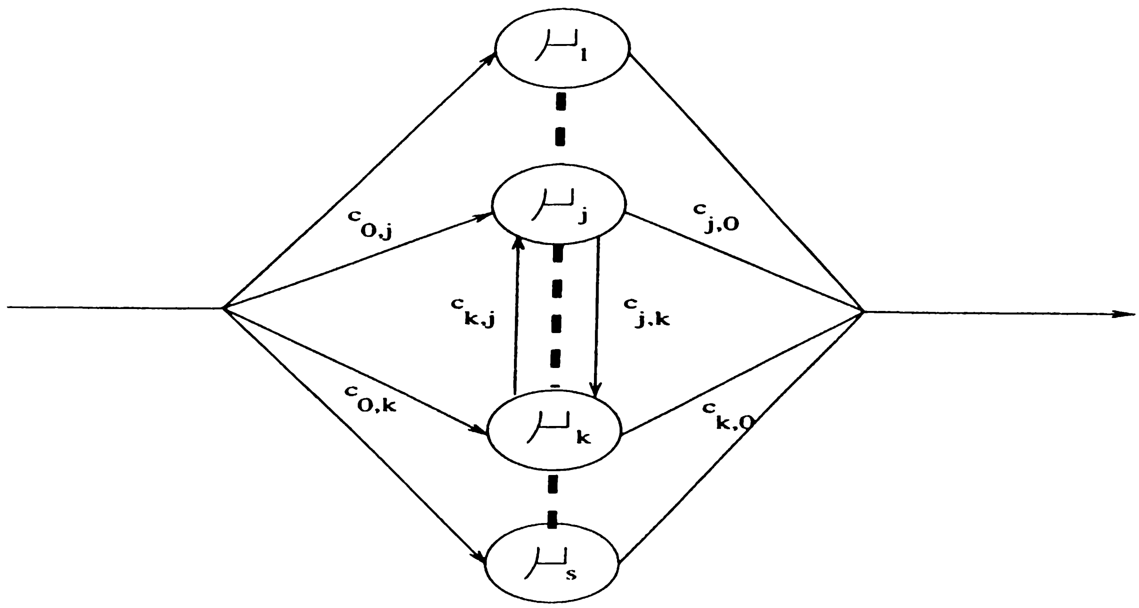

Figure 2.2: Phase-type distribution with s phases

is the price to pay to handle more realistic models having non-exponential distributions.

Similarly, discrete phase-type distributions can be defined. In that case, the geometric distribution plays the same role as the exponential distribution of continuous case.

Definition: A probability distribution $F(\cdot)$ on $[0,\infty)$ is a *distribution of phase type (PH distribution)* if and only if it is the distribution of the time until absorption in a finite Markov process having the states $\{1,\ldots,m+1\}$ with infinitesimal generator

$$\mathbf{Q} = \begin{vmatrix} \mathbf{T} & \mathbf{T^o} \\ \mathbf{0} & 0 \end{vmatrix},$$

where the non-singular $m \times m$ matrix $\mathbf{T}$ satisfies $T_{ii} < 0, for\ 1 \leq i \leq m, and\ T_{i,j} \geq 0,\ for\ i \neq j$. Moreover, $\mathbf{Te}+\mathbf{T^o} = \mathbf{0}$, and the initial probability vector of $\mathbf{Q}$ is given by $(\boldsymbol{\alpha},\alpha_{m+1})$ with $\boldsymbol{\alpha}\mathbf{e}+\alpha_{m+1} = 1$ and states $1,\ldots,m$ are all transient, so that absorption into the state $m+1$, from any initial state, is certain. The pair $(\boldsymbol{\alpha},\mathbf{T}\ )$ is called a *representation* of $F(\cdot)$.

The generalized Erlang distribution of order k with parameters $\lambda_1, \ldots, \lambda_k$ has the representation $\boldsymbol{\alpha} = (1, 0, \ldots, 0)$ and

$$T = \begin{vmatrix} -\lambda_1 & \lambda_1 & & & \\ & -\lambda_2 & \lambda_2 & & \\ & & \cdots & & \\ & & & -\lambda_{k-1} & \lambda_{k-1} \\ & & & & -\lambda_k \end{vmatrix}$$

In special k−Erlang distribution, we have $\lambda_1 = \lambda_2 = \cdots = \lambda_k$.

Discrete PH-distributions are defined by considering an $m+1$ state Markov chain P of the form

$$\mathbf{P} = \begin{vmatrix} \mathbf{T} & \mathbf{T}^o \\ \mathbf{0} & \mathbf{1} \end{vmatrix},$$

where $\mathbf{T}$ is a substochastic matrix, such that $\mathbf{I} - \mathbf{T}$ is nonsingular. The initial probability vector is $(\boldsymbol{\alpha}, \alpha_{m+1})$. The probability density $\{p_k\}$ of phase type is given by

$$\begin{aligned} P_0 &= \alpha_{m+1}, \\ P_k &= \boldsymbol{\alpha}\ \mathbf{T}^{k-1}\mathbf{T}^o\ ,\ \text{for } k \geq 1. \end{aligned}$$

In the analysis of two-machine flow lines with reliable machines having continuous phase-type distribution, any numerical technique for discrete space Markov processes can in principle be used. However, it is important to recognize that the Markov process has a very special structure and one must take advantage of it. 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. We can characterize the behaviour of such a system by a discrete state, continuous time Markov chain and then analyze this system to calculate the steady-state probabilities of the Markov chain and derive all the performance measures.

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 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 0 *to* s_1, where $j_1 = 0$ represents the case of blocking of machine M_1. Similarly, j_2 can take on integer values from 0 *to* s_2, but in that case $j_2 = 0$ represents starvation of machine M_2.

If the state space is partitioned according to the values of n, and $\mathbf{p}$ denotes the steady-state probability vector and $\mathbf{p_n}$ denotes the portion of that vector that corresponds to a buffer content of n, we can write

$$\mathbf{p} = \begin{pmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_N \end{pmatrix}$$

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

Let $\mathbf{Q}$ denote the infinitesimal generator of the Markov chain. The steady-state probability vector $\mathbf{p}$ of the Markov chain is the solution of the equation

$$\mathbf{Q^T p} = 0.$$

In addition, $\mathbf{p}$ also satisfies the normalization equation

$$\mathbf{1^T p} = 1.$$

Matrix $\mathbf{Q^T}$ is a *block tridiagonal* matrix with the following special structure

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

where A, B, C are square matrices of size $(s_1s_2,\ s_1s_2)$; 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 $(s_1s_2,\ s_1)$, $(s_1,\ s_1s_2)$, $(s_2,\ s_1s_2)$, $(s_1s_2,\ s_2)$ respectively.

$\mathbf{Q^T}$ has this special structure because the Markov chain associated with a two-machine transfer line is a *generalized birth-death process*. Transitions occur only between states that are neigbours 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 state (n', j_1', j_2') such that $n' = n, n-1, n+1$. Moreover, transition rates are independent of n , for $1 < n < N-1$. Because of the special block tridiagonal structure of $\mathbf{Q^T}$, equation $\mathbf{Q}^T\mathbf{p} = 0$ can be decomposed into the following set of equations, which we call *transition equations*,

$$
\begin{aligned}
&B_0\mathbf{p}_0 + A_0\mathbf{p}_1 = 0, \\
&C_0\mathbf{p}_0 + B\mathbf{p}_1 + A\mathbf{p}_2 = 0, \\
&C_0\mathbf{p}_{n-1} + B\mathbf{p}_n + A\mathbf{p}_{n+1} = 0, \quad 1 < n < N-1, \\
&C\mathbf{p}_{N-2} + B\mathbf{p}_{N-1} + A_N\mathbf{p}_N = 0, \\
&C_N\mathbf{p}_{N-1} + B_N\mathbf{p}_N = 0.
\end{aligned}
$$

The special structure of the matrix $\mathbf{Q^T}$ led to two solution techniques that make use of this special structure. These are the *recursive technique* and the *matrix geometric technique.*

For matrix geometric technique the reader is referred to Neuts [36]. The principle of the matrix geometric solution can be briefly described as follows. The first step is to show that the set of transition equations can be transformed into an equation of the form

$$N\mathbf{p}_n + M\mathbf{p}_{n-1} = 0,$$

where the matrices N and M are of size(s_1s_2, s_1s_2) and N is invertible. If we define a matrix R as $R = -N^{-1}M$, we have

$$\mathbf{p}_n = R\mathbf{p}_{n-1},\ 1 < n < N.$$

For the boundary states, if we also define matrices S and U such that $\mathbf{p}_1 = S\mathbf{p}_0$ and $\mathbf{p}_N = U\mathbf{p}_{N-1}$. $\mathbf{p}_0$ can be determined by solving an equation of the form $Z\mathbf{p}_0 = \mathbf{x}$ which is obtained from the basic set of transition equations. Then, the remaining probabilities can be obtained. For more details, see Gun [19] and Gun and Makowski [20].

The recursive technique can be applied to Markov processes that satisfy the condition that there exists a subset of states, *boundary states*, such that the probabilities of all other states can be obtained recursively from the probabilities of the boundary states. The recursive technique can be implemented using the following algorithm [10].

(1) Reduction step-determine M boundaries and derive a recursive scheme, to calculate all other state probabilities from the boundary state probabilities. Then, express all state probabilities as linear combinations of the boundary values. In order to find the coefficient of a particular boundary value in the linear expression, set that boundary value to 1 and all other boundary values to 0, and then follow through the recursive scheme. This is done M times, corresponding to the M boundaries. There will be M equations not used in the calculation of the state probabilities in terms of the boundary values. $M - 1$ of these equations together with the normalizing equation give, after substituting expressions in terms of boundary state probabilities for non-boundary probabilities, M equations for the M boundary state probabilities.

(2) Solution step -determine the M boundary state probabilities by solving the set of M equations.

(3) Evaluation step -from the recursive scheme, determine the remaining state probabilities. Key to the use of this recursive algorithm is the

determination of how many boundaries to use and which specific states to be chosen as the boundary states.

2.1.2 Approximate Solution Techniques

Most approximate methods are based on ***decomposition***. Each decomposition method involves three steps. First step is the characterization of the subsystems, then a set of equations is derived to determine the unknown parameters of each subsystem. Finally, an algorithm is developed 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 second step aims to establish relationships between quantities pertaining to different subsystems so as to derive the parameters of each subsystem from the parameters and performance measures of other subsystems.

Most decomposition methods in the literature decompose a K−machine flow line into a set of $K-1$ subsystems where each subsystem is associated with a buffer of the original line. In some methods, the subsystem is a two-machine line while in others it consists of a single server queue with a finite buffer. Since the subsystems are always simpler than the whole line, they cannot exhibit the same behaviour. Moreover, some of the equations used to determine the parameters may be approximate, even within their assumptions. Thus, decomposition methods are approximations. For the principles of decomposition methods for flow lines with reliable machines, the reader is referred to Hillier and Boling [23].

The decomposition approach decomposes the original K−machine line into a set of $K-1$ two-machine lines. Each two-machine line is associated with a buffer of the original line. Let L denote the original line and $L(i,i+1)$ denote the two-machine line associated with buffer $B_{i,i+1}$. Let subscripts u and d refer to objects and parameters of the upstream and downstream machines. Machine $M_u(i,i+1)$ is the upstream machine, and $M_d(i,i+1)$ is the downstream machine of line $L(i,i+1)$. The decomposition approach is depicted in Figure 2.3.

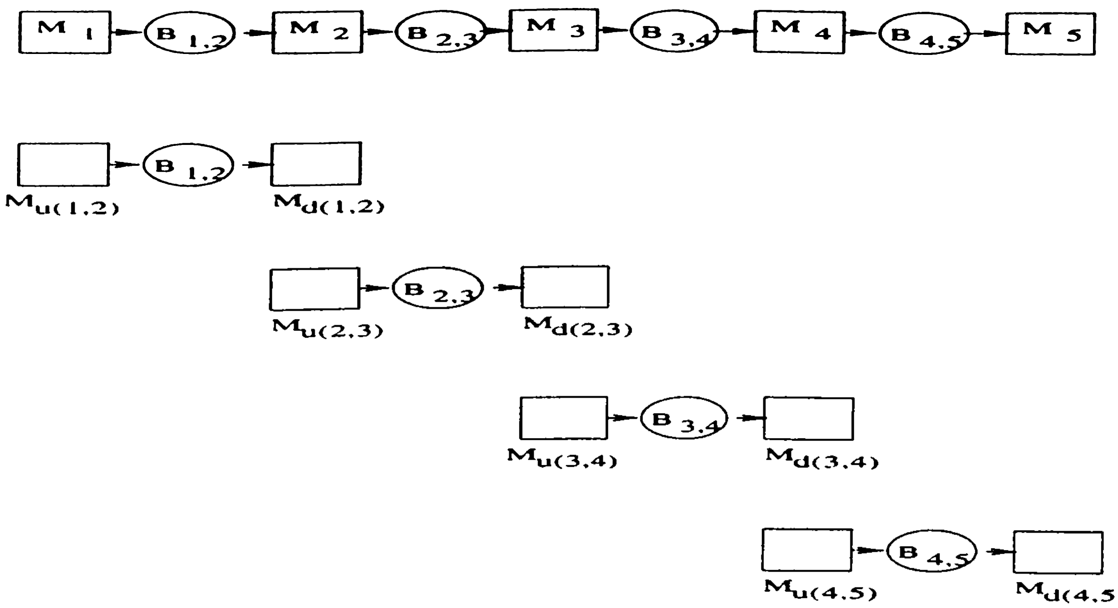

Figure 2.3: Flow line decomposition

The basic idea in decomposition is the definition of upstream and downstream machines for each two-machine line $L(i,i+1)$ such that the behaviour of material through its buffer is close to the behaviour of material in buffer $B_{i,i+1}$ in line L.

Upstream and downstream machines of each two-machine line summarize the effects of the entire up and downstream portions of the line, respectively, on the buffer. For example, machine $M_u(i,i+1)$ represents the portion of the line L upstream of buffer $B_{i,i+1}$, that is, machine M_1 to machine M_i. Likewise, machine $M_d(i,i+1)$ represents the portion of line L downstream of buffer $B_{i,i+1}$, that is, machine M_{i+1} to machine M_K. In the literature, these machines are usually called ***equivalent machines***, ***pseudo-machines***, or ***virtual machines***.

Furthermore, there are alternative decomposition methods which decompose a K−machine line into a set of $K - 2$ three-machine lines. This approach may result in more accurate results. Nevertheless, it requires repetitive solutions of three-machine subsystems which are too complex to solve in general as discussed before.

2.2 Problem Definition and Solution Procedure

We investigate the behaviour of a transfer line with three machines and two finite buffers. Since, the machines are assumed to be reliable, the only source of randomness in the system is the random machine processing times which are assumed k−stage Erlang distributed.

There are mainly four factors to be considered in choosing a distribution function. These are, factor 1, its ease of mathematical manipulation; factor 2, its ease of fitting, that is, of determining its parameters from standard summary statistics (such as mean, variance, range, etc.) of empirical data; factor 3, its resemblance to empirical distribution from actual data; and factor 4, its consistency with the *"principle of entropy maximization"*.

Regarding the last factor, information theory recommends that, for a given set of statistical conditions, a distribution function that maximizes the *'entropy'* should be adopted, subjected to the satisfaction of the given conditions. A non-technical interpretation of entropy maximization is that the selected distribution function fully reflect the information given on the random variable but should not impose on it any additional assumptions.

Exponential distribution scores very well on factors 1, 2 and 4. Regarding factor 4, if only the mean of a non-negative random variable is known, the exponential distribution maximizes the entropy.

The Erlang distribution is often used to represent processing times in unpaced line models since it can assume a wide range of different skewness and; therefore, be suitable for fitting real-life empirical distributions of processing times [29]. Moreover, since the sum of k independently and identically distributed exponential random variables with mean $1/k\mu$ yields a $k-$Erlang distribution with parameter μ, Erlang allows us to describe queueing models where the service may be a series of identical phases. Hence, the most important reason why the Erlang distribution is useful in queueing analyses is its relation to the exponential distribution which is the only continuous distribution with Markovian property.

The underlying assumptions of our model are as follows. Considering the material, it consists of discrete parts and there is only a single kind of material in the system. Each piece of material visits the machines and buffers in exactly the same sequence.

The machines are not constrained to start or stop their operations at the same instant; therefore, it is an asynchronous transfer line. Machine processing times are $k-$stage Erlang distributed and the time between part arrivals to the system is exponentially distributed with parameter λ.

Some models of flow lines have machines that can fail. When a failure occurs, machine cannot process any material, so the buffer upstream cannot lose material and the buffer downstream cannot gain material. Systems in which machines cannot fail are called *Flow Lines with Reliable Machines (FLRM)*'s. Our research assumes that the system is FLRM's.

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 takes an especially long time to process a part, and its neighbour machines work normally, the level of buffer $B_{i-1,i}$ increases and the level of buffer $B_{i,i+1}$ decreases.

If this situation persists, buffer $B_{i-1,i}$ may become full or buffer $B_{i,i+1}$ may become empty. In this case, one of the neighbour machines of M_i is not able to operate; either machine M_{i+1} is *starved* or machine M_{i-1} is *blocked*. In real life systems, it is possible that raw material is absent, or the means of removal of finished goods fail. We asume that the calling population is infinite and the last machine is never blocked.

Considering the operating policy, in our system, 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...[30]...(1982) demonstrates that this is the optimal operating policy for a two-machine line when the performance measure is the system production rate.

Quality is not treated in the model presented here. All parts are assumed perfect. There is no inspection procedure, no rework, and no rejects. Furthermore, the material in the storage buffers is assumed to be nonperishable.

The transfer line system we investigate is depicted in Figure 2.4. where ∞'s indicate that the system is saturated.

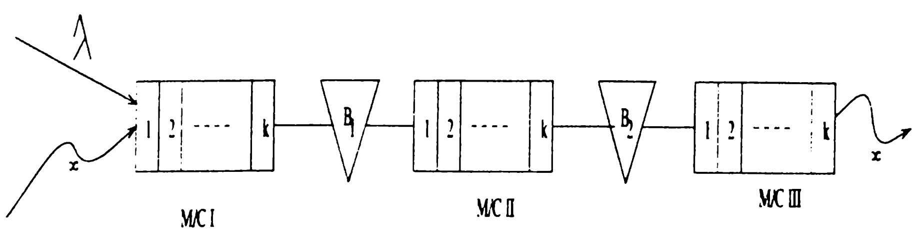

Figure 2.4: 3-machine saturated k-Erlang transfer line with Poisson (λ) arrivals

This system is modeled as a discrete state space continuous time Markov chain.

Markov chain models of transfer lines are difficult to treat due to their large state spaces and their indecomposability. When the system is modeled as a discrete state Markov chain. the number of distinct states is the product of the number of different machine states, and the number of distinct buffer levels.

Many models of queueing networks are decomposable; that is, portions of the system can be treated as if they are isolated from other portions. The mathematical models break up into smaller models, with simple relationships among them. Yet, the Markov chain models of transfer lines do not have this property. No exact decomposition exists.

Before presenting the states of Markov chain model of this transfer line, it is worth addressing the following fact again. A $k-$Erlang distribution with parameter μ/k is represented as the sum of k independently and identically distributed exponential random variables with mean $1/\mu$. Hence, this relation allows us to describe the service system as a series of identical phases that have exponential processing times with parameter μ . This is shown in Figure 2.5.

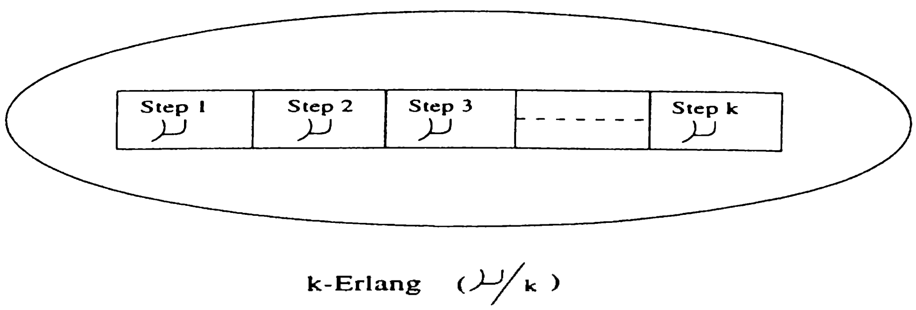

Figure 2.5: Use of the Erlang for phased service

By the help of this observation, even though the service may not actually consist of phases, in the state representation of the Markov chain, we also denote the phases of the services so that we can exploit the Markovian property of exponential distribution by means of this phased service idea. Therefore, the state representation is as follows:

$$(n_1; i_1, B_1, n_2; i_2, B_2, n_3; i_3)$$

where n_1, n_2, and n_3 denote the states of the machines respectively. n_1, and n_2 can take on values 1, 0, and b where 1 denotes that machine is up and working, 0 denotes that machine is idle, and b denotes that machine is blocked. Yet, n_3 can take on values 1 and 0, but not b because of the fact that it is a saturated transfer line. i_1, i_2, and i_3 represent the stages of the services for the machines respectively. B_1 is the size of the finite buffer in front of the second machine, and B_2 is the size of the finite buffer in front of the third machine.

The following is the all possible states of the Markov chain model of the transfer line.

1. $(0;0,0,0;0,0,0;0)$
2. $(1;k,0,0;0,0,0;0)$ $\quad k=1,2,\ldots,K$
3. $(1;k,B_1,1;l,0,0;0)$ $\quad k,l=1,2,\ldots,K\,,\ B_1=0,1,\ldots,B$
4. $(0;0,B_1,1;l,0,0;0)$ $\quad l=1,2,\ldots,K\,,\ B_1=0,1,\ldots,B$
5. $(0;0,0,0;0,B_2,1;j)$ $\quad j=1,2,\ldots,K\,,\ B_2=0,1,\ldots,C$
6. $(1;k,0,0;0,B_2,1;j)$ $\quad k,j=1,2,\ldots,K\,,\ B_2=0,1,\ldots,C$
7. $(0;0,B_1,1;l,B_2,1;j)$ $\quad l,j=1,2,\ldots,K\,,\ B_1=0,1,\ldots,B$, $B_2=0,1,\ldots,C$
8. $(1;k,B_1,1;l,B_2,1;j)$ $\quad k,l,j=1,2,\ldots,K\,,\ B_1=0,1,\ldots,B$, $B_2=0,1,\ldots,C$
9. $(b;0,B,1;l,0,0;0)$ $\quad l=1,2,\ldots,K$
10. $(b;0,B,1;l,B_2,1;j)$ $\quad l,j=1,2,\ldots,K\,,\ B_2=0,1,\ldots,C$
11. $(0;0,B_1,b;0,C,1;j)$ $\quad j=1,2,\ldots,K\,,\ B_1=0,1,\ldots,B$
12. $(1;k,B_1,b;0,C,1;j)$ $\quad k,j=1,2,\ldots,K\,,\ B_1=0,1,\ldots,B$
13. $(b;0,B,b;0,C,1;j)$ $\quad j=1,2,\ldots,K$

In order to calculate the performance measures of the system, the steady-state probabilities of the Markov chain are calculated. The Markov chain has a steady-state distribution because it is an ergodic Markov chain. Miltenburg [32] proves that for finite-size buffer inventories, the states of a K station transfer line constitute an ergodic Markov chain. Our state representation differs from the others in the literature that we also keep track of the phases. However, this representation of phases do not violate the ergodicity and there is also an embedded pure birth process if we just consider the process of passing from one phase to the other.

To obtain the steady-state distribution, instead of studying the system by means of the transition equations, we use the balance equations, which equate the rate of leaving a state with the rate of entering it.

For a discrete state, continuous time Markov chain, the balance equation is in the from

$$\frac{d\mathbf{p}_i(t)}{dt} = \sum_j \lambda_{ij}\mathbf{p}_{ij}(t)$$

where $\lambda_{ij} > 0$, $j \neq i$, and

$$\lambda_{ii} = -\sum_{j \neq i} \lambda_{ji}$$

In steady state $d\mathbf{p}_i/dt = 0$, and the negative term of the balance equation can be moved to the left side, so it becomes

$$\mathbf{p}_i \sum_{j \neq i} \lambda_{ji} = \sum_{j \neq i} \lambda_{ji}\mathbf{p}_j$$

The left side is the rate of the system leaving state i, and the right side is the rate of entering it.

While writing the balance equations, one has to consider two events. These are the arrival process, and the service process.

For the arrival process, we identify the following probabilities:

$$P\{An\ arrival\ occurs\ in\ \Delta t\} = \frac{e^{-\lambda \Delta t}(\lambda \Delta t)^1}{1!}$$

$$P\{No\ arrivals\ occur\ in\ \Delta t\} = \frac{e^{-\lambda \Delta t}(\lambda \Delta t)^0}{0!}$$

In the above equations, for the term $e^{-\lambda\Delta t}$ we can make the following approximation by using Taylor's series expansion:

$$e^{-\lambda\Delta t} = \sum_{k=0}^{\infty} \frac{(\lambda\Delta t)^k(-1)^k}{k!} = 1 - \lambda\Delta t + \frac{(\lambda\Delta t)^2}{2!} + \frac{(\lambda\Delta t)^3}{3!} + \cdots$$
$$\simeq 1 - \lambda\Delta t$$

Hence, if we approximate $e^{-\lambda\Delta t}$ by $1 - \lambda\Delta t$, we get the following probabilities.

$$P\{An\ arrival\ occurs\ in\ \Delta t\} = \lambda\Delta t(1 - \lambda\Delta t)$$

$$P\{No\ arrivals\ occur\ in\ \Delta t\} = 1 - \lambda\Delta t$$

For the service process, for each phase of the service we identify the following probabilities:

$$P\{Service\ completion\ in\ \Delta t\} = \frac{e^{-\mu\Delta t}(\mu\Delta t)^1}{1!}$$
$$P\{No\ service\ completion\ in\ \Delta t\} = \frac{e^{-\mu\Delta t}(\mu\Delta t)^0}{0!}$$

where μ is the rate of exponential distribution.

For these equations, an approximation similar to that of the arrival process is made. Thus, if we approximate $e^{-\mu\Delta t}$ by $1 - \mu\Delta t$, we get the following probabilities:

$$P\{Service\ completion\ in\ \Delta t\} = \mu\Delta t(1 - \mu\Delta t)$$

$$P\{No\ service\ completion\ in\ \Delta t\} = 1 - \mu\Delta t$$

The balance equations are generated by an algorithm coded in the programming language *C*. This system of linear equations is solved by the optimization software CPLEX because of its speed. Then, performance measures are calculated by another program again coded in C. The codes of these programs are not provided with the thesis work and they can be obtained directly from the author.

Chapter 3

EXPERIMENTAL RESULTS

This chapter covers the experimental results of our study. We solve different problems so as to observe the effect of K; the number of phases in the Erlang service, the effect of buffer sizes B; the size of the first buffer and C; the size of the second buffer, and the effect of the machine processing rates: μ_1, μ_2, μ_3, on the performance measures. The size of the balance equations generated for several problems varies between 200 and 6000.

We are interested in mainly four performance measures. These are the utilization of the machines, mean throughput, mean Work-In-Process inventory (WIP), variance of Work-In-Process inventory.

Machine utilization is calculated as the percentage of time that a machine is working; that is neither blocked nor idle. High machine utilization is assumed to be good because it amortizes the cost of the machinery faster. Nevertheless, by forcing a machine to run so as to amortize its cost and increase its utilization, one is simply transferring a machine asset into an inventory asset. Hence, it is important to differentiate the most beneficial policy, whether to increase utilization or decrease inventory.

Work-In-Process is the amount of semi-finished product currently resident on the factory floor. A semi-finished product is either being processed or is waiting for the next processing operation. We investigate both the mean value and the variance of Work-In-Process inventory so that confidence intervals for WIP can be constructed.

The throughput is the number of parts produced per unit time. The reciprocal of the throughput is the production time per unit of the product. For transfer lines, the throughput approximates the reciprocal of the cycle time. Mean value of the throughput is the expected number of parts produced per unit time in the system. In the long run, when the system achieves a steady state, the mean throughput is equal to the effective arrival rate which is the product of arrival rate, λ, and the percent idle time of the first machine. It is also equal to the product of the processing rate and the utilization of the last machine of the line.

First of all, the effect of machine processing rates is investigated for K=2 and K=3. The graphs of the performance measures are shown in figures 3.1. to 3.6. For both K=2 and K=3 cases, if the rates of the machines are increased simultaneously, utilization of the machines decreases as expected. It is also important to note that utilization of the three machines are almost equal since their processing times are independent and identically distributed with the same rate. Furthermore, utilizations of the machines stay almost constant with respect to changing buffer sizes. This can be explained as that for these cases, rather than the buffer sizes, machine processing rates play significant role in determining the utilizations. Later, we observe the same thing for different K values as well. For mean throughput, the graph is not given. However, the tables that summarize the results of all experiments are provided at the end of this chapter. For both cases, mean throughput increases when the processing rate increases for all machines. For the expected value of WIP, as the processing rate increases, the expected value of WIP decreases. This is intuitive because when machines are faster, the part travels through the transfer line faster. Moreover, it can be observed that expected value of WIP is a linear function of the buffer size. Although variance of WIP is the same for processing rates

$\mu_1 = \mu_2 = \mu_3 = 2$/unit time and $\mu_1 = \mu_2 = \mu_3 = 3$/unit time for K=2, the variance is larger when $\mu_1 = \mu_2 = \mu_3 = 3$/unit time for the transfer line where the machine processing rates are 3−Erlang distributed. Variance of WIP tends to increase as processing rate of the machines increases. Another important observation is that variance of WIP is an exponential function of the buffer sizes. Keeping these observations in mind, although high processing rate gives smaller mean WIP values, one should try to balance the effect of increasing processing rate on the variance of WIP and the mean value of it when the concern is to keep the interval that the value of WIP lies small.

As a next step, to observe the effect of K, performance measures for K=2 and K=3 are compared. These are illustrated in figures 3.7 to 3.9. For the case of K=3, utilization of the machines is higher than that of K=2 case. This is expected because as the number of phases is increased, the part spends more time on each machine keeping it busy. Furthermore, utilizations of all machines are almost equal again for both cases. Mean throughput drops when we increase K since the time it takes for a part to be processed through the transfer line increases, causing the number of parts produced per time to fall. Expected value of WIP is greater when K increases. On the other hand, variance of WIP decreases as the number of stages of the Erlang distribution is increased. That is, when K is increased, more WIP is carried, but the deviation from this value is less. Hence, if the aim is to keep a stable WIP level, although it may be a little bit high, smaller K values should be chosen.

In order to see the effect of buffer sizes, we investigate the performance measures for the case when the size of one of the buffers is fixed and the other is varied. The results of different problems are represented in tables 3.1. to 3.22. Whereas, we only provide the graphs for the cases B=4, and C is varied, C=4, and B is varied; B=5, and C is varied, C=5, and B is varied; B=6, and C is varied, C=6, and B is varied. These graphs are given in figures 3.10. to 3.18. For all cases, utilization is almost equal for the case where C is fixed, B is varying and the case where B is fixed, C is varying. For all performance measures, the following is observed. When the size of one of the buffers is set to a constant value, c, and the size of the other buffer is varied, the performance measure is better until the point where $B = C = c$ for the case where B is fixed and C is changing, and after that point performance measures are better for the case where C is constant and B varies. Another observation is that for the case where C is constant, and B is varied, performance measures are always around a stationary value. Whereas, for the opposite case, although the utilization and mean throughput are almost constant, expected WIP increases linearly while variance of WIP exhibits an exponentially increasing behaviour as discussed previously. Moreover, although buffer sizes are increased, throughput stays almost the same for both cases. Hillier and So [24] prove that percentage increases in throughput decrease as buffer capacities increase. Hence, our finding also supports this observation.

These observations lead to the following observation. If the processing rates of the machines are equal and if there is restricted available space for buffers. i.e. when a total amount must be allocated between the two buffers, the first buffer always should get more if we want to reduce the expected value of WIP and the variance of it. For example, if the total available space is 12 parts for K=3, λ=1/unit time, $\mu_1 = \mu_2 = \mu_3$=2/unit time case, B=9. C=3 combination gives an expected WIP value of 3.73741 and a variance value of 0.25579. However, B=8, C=4 gives 3.95295 and 1.35911 respectively. Finally, B=7, C=5 gives 4.16362 and 2.79884 respectively.

Finally, we try 0 processing rate combinations, such as $\mu_1 < \mu_2 < \mu_3$, $\mu_1 > \mu_2 > \mu_3$, $\mu_1 > \mu_2 < \mu_3$, and $\mu_1 < \mu_2 > \mu_3$. Obviously, the utilization of the machines are higher for smaller processing rates. For $\mu_1 < \mu_2 < \mu_3$, we performed experiments to see the effect of changing buffer sizes. Again until the intersection point, performance measures are better for the case where C is constant and B is varying. All the previous discussion related to changes in buffer sizes for constant processing rates are also valid for $\mu_1 < \mu_2 < \mu_3$ and others. It is worth in noting that expected value of WIP is again a linear function of buffer size while the variance of WIP is increasing exponentially with the increasing buffer size. The plots of these experiments are given in figures 3.19. to 3.20.

After these observations, we compare different processing rate combinations for the case where C=6, B is changing. We do not consider B=6, C is changing case because the former performs better as discussed before. The plots for these experiments are given in figures 3.21. to 3.23. In the plots, notation is as follows. a, b, c denotes that $\mu_1 = a$/unit time, $\mu_2 = b$/unit time, $\mu_3 = c$/unit time respectively. For machine utilizations, obviously one get the highest utilization if the processing rate of that machine is kept low. For mean throughput, $\mu_1 = 4$/unit time, $\mu_2 = 3$/unit time, $\mu_3 = 2$/unit time gives the best result and $\mu_1 = 3$/unit time, $\mu_2 = 2$/unit time, $\mu_3 = 4$/unit time gives the second best. Expected value of WIP is the smallest for the case where $\mu_1 = 2$/unit time, $\mu_2 = 3$/unit time, $\mu_3 = 4$/unit time and it is the highest when $\mu_1 = 4$/unit time, $\mu_2 = 3$/unit time, $\mu_3 = 2$/unit time. This may be due to the fact that if the first machine of the transfer line is the fastest, and the last machine is the slowest, although parts pass to successive machines faster, they will wait in the buffers for the completion of processing of parts already residing on the successive machines because those are slower. Consequently, if machine load allocation is considered for a fixed available amount, $\mu_1 < \mu_2 < \mu_3$ gives the best WIP value. $\mu_1 = 3$/unit time, $\mu_2 = 2$/unit time, $\mu_3 = 4$/unit time combination gives the second best value in terms of the expected WIP. Yet, for the variance of WIP, $\mu_1 = 4$/unit time, $\mu_2 = 3$/unit time, $\mu_3 = 2$/unit time combination gives the smallest value whereas $\mu_1 = 2$/unit time, $\mu_2 = 3$/unit time, $\mu_3 = 4$/unit time results in the highest value and $\mu_1 < \mu_2 < \mu_3$ combination leads to the second highest value. Hence, if the concern is to keep the WIP at a small and stationary value one should choose $\mu_1 = 3$/unit time, $\mu_2 = 2$/unit time, $\mu_3 = 4$/unit time combination.

There are many alternative experimental design procedures. Nevertheless, only a few of them are presented because of space considerations. However, with the available code any kind of relationship can be investigated further.

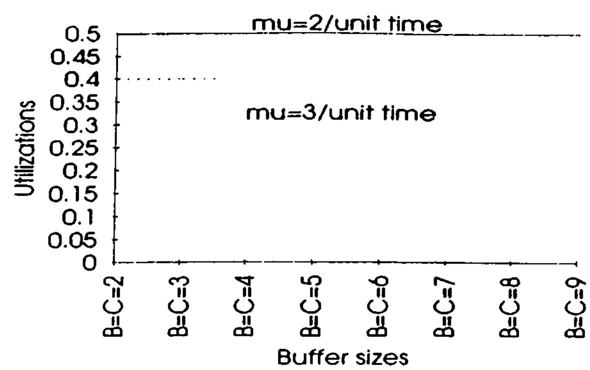

Figure 3.1: Utilization of machines for K=2, $\lambda=1$, and varying buffer sizes and machine processing rates

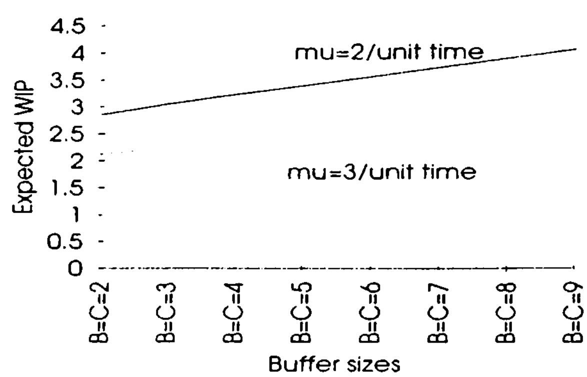

Figure 3.2: Expected WIP for K=2, $\lambda=1$, and varying buffer sizes and machine processing rates

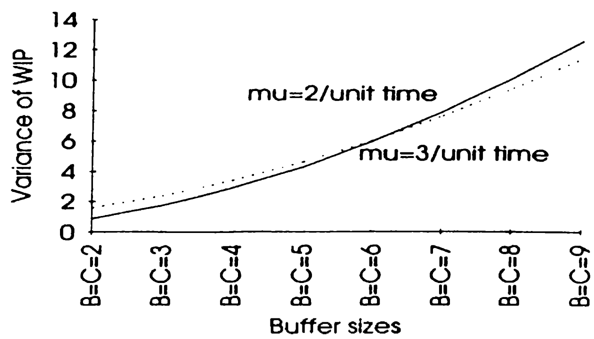

Figure 3.3: Variance of WIP for K=2, λ=1, and varying buffer sizes and machine processing rates

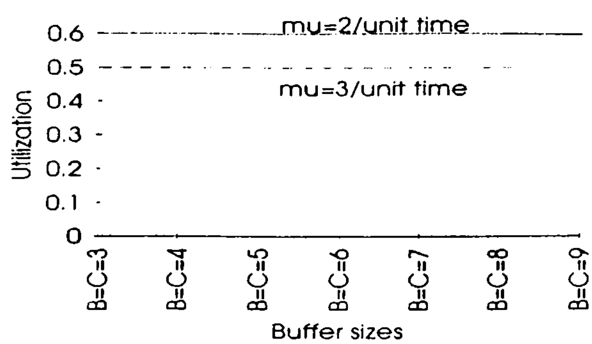

Figure 3.4: Utilization of machines for K=3, λ=1, and varying buffer sizes and machine processing rates

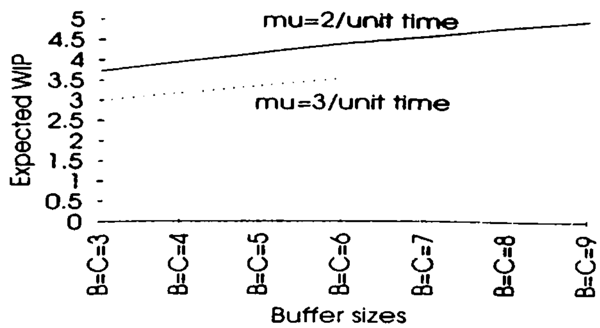

Figure 3.5: Expected WIP for K=3, λ=1, and varying buffer sizes and machine processing rates

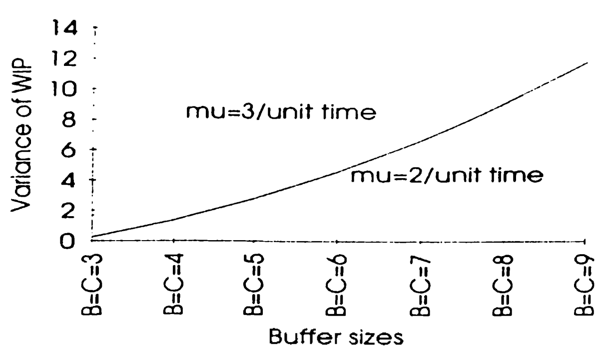

Figure 3.6: Variance of WIP for K=3, λ=1, and varying buffer sizes and machine processing rates

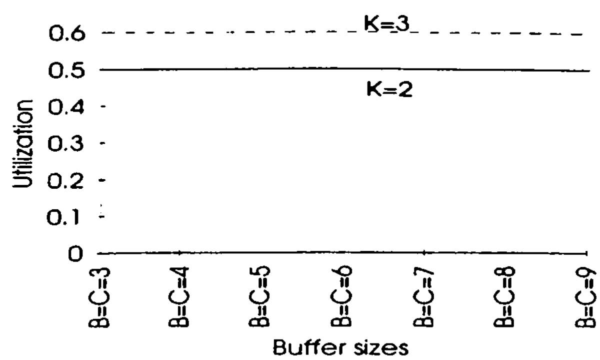

Figure 3.7: Utilization of machines for fixed processing rate 2/unit time, $\lambda=1$, and varying buffer sizes

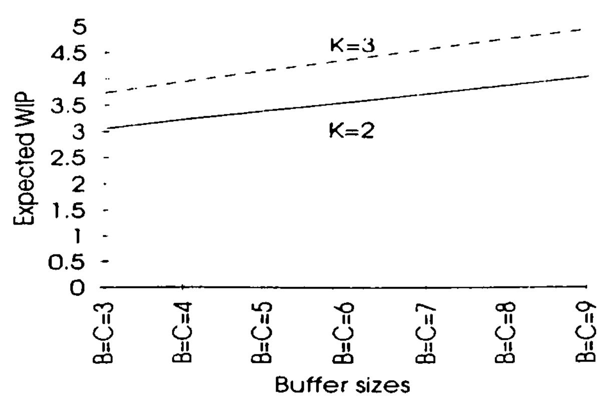

Figure 3.8: Expected WIP for fixed processing rate 2/unit time, $\lambda=1$, and varying buffer sizes

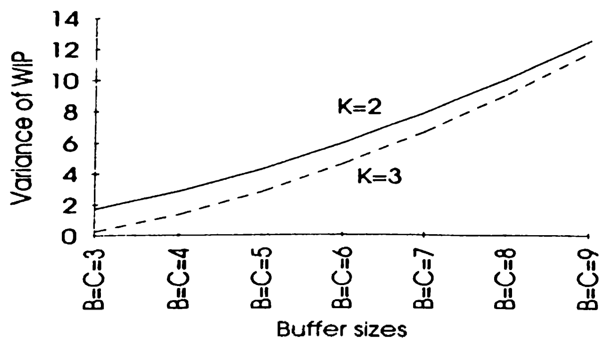

Figure 3.9: Variance of WIP for fixed processing rate 2/unit time, λ=1, and varying buffer sizes

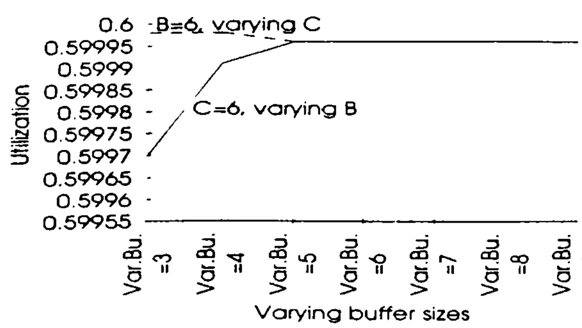

Figure 3.10: Utilization of machines for K=3, λ=1, μ=2/unit time, and varying buffer sizes

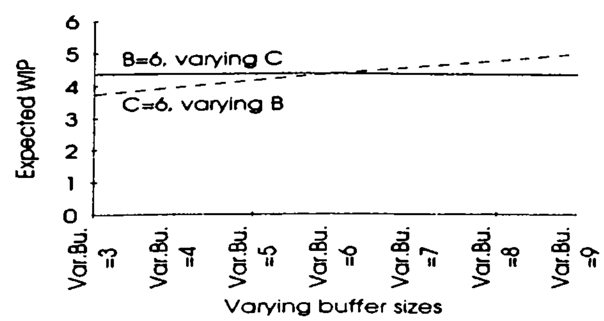

Figure 3.11: Expected WIP for K=3, $\lambda=1$, $\mu=2$/unit time, and varying buffer sizes

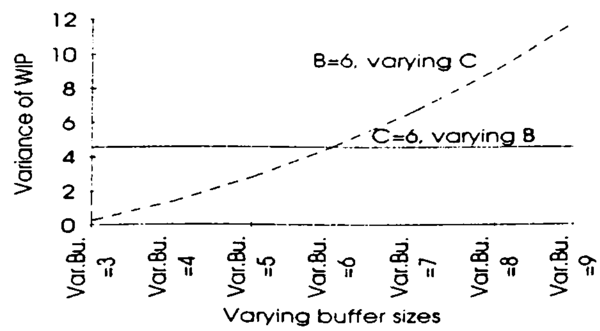

Figure 3.12: Variance of WIP for K=3, $\lambda=1$, $\mu=2$/unit time, and varying buffer sizes

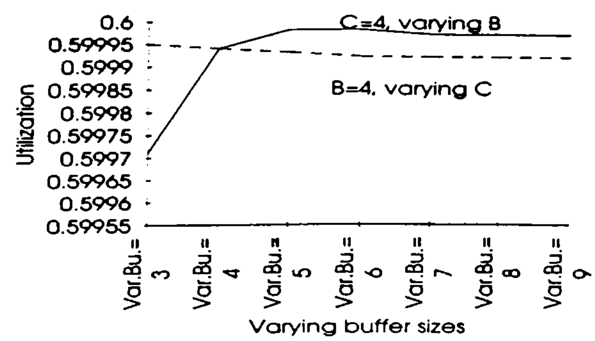

Figure 3.13: Utilization of machines for K=3, λ=1, μ=2/unit time, and varying buffer sizes

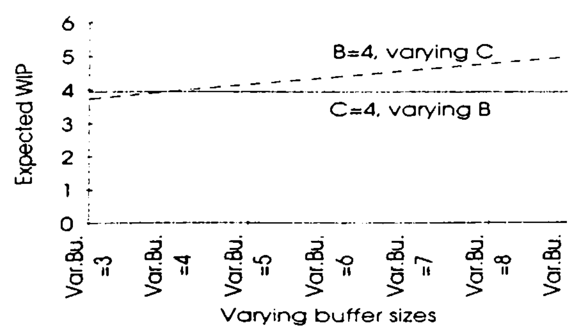

Figure 3.14: Expected WIP for K=3, λ=1, μ=2/unit time, and varying buffer sizes

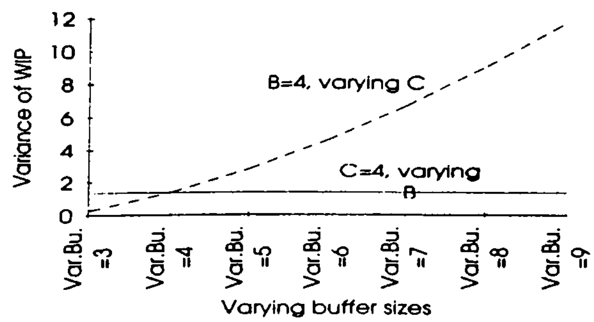

Figure 3.15: Variance of WIP for K=3, λ=1, μ=2/unit time, and varying buffer sizes

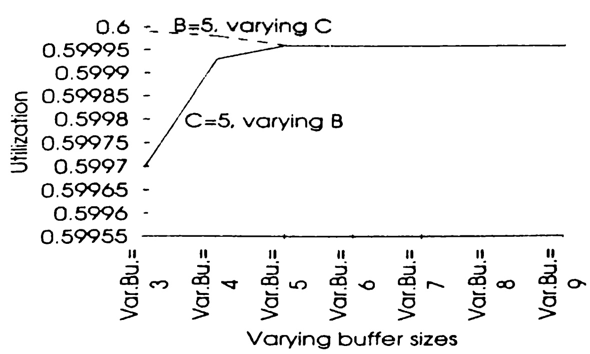

Figure 3.16: Utilization of machines for K=3, λ=1, μ=2/unit time, and varying buffer sizes

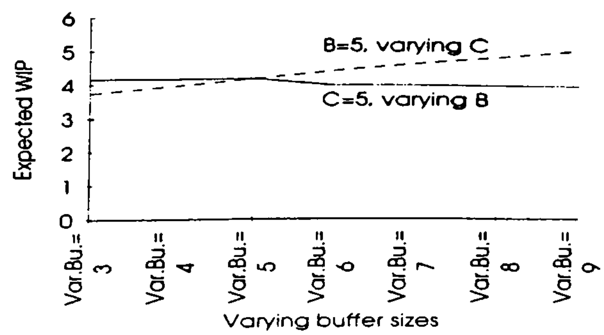

Figure 3.17: Expected WIP for K=3, $\lambda=1$, $\mu=2$/unit time, and varying buffer sizes

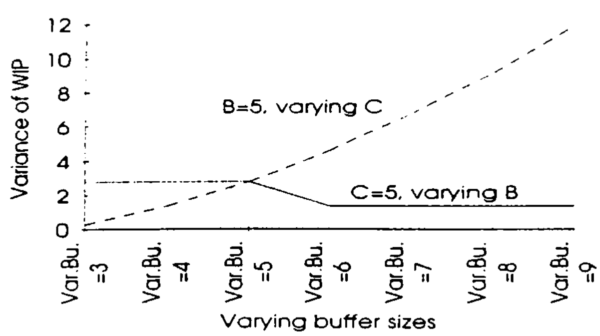

Figure 3.18: Variance of WIP for K=3, $\lambda=1$, $\mu=2$/unit time, and varying buffer sizes

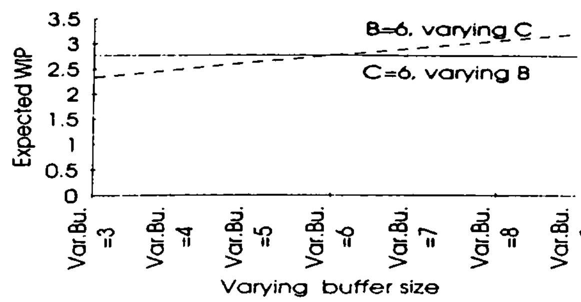

Figure 3.19: Expected WIP for K=3, λ=1, μ_1 = 2/unit time, μ_2 = 3/unit time, μ_3 = 4/unit time, and varying buffer sizes

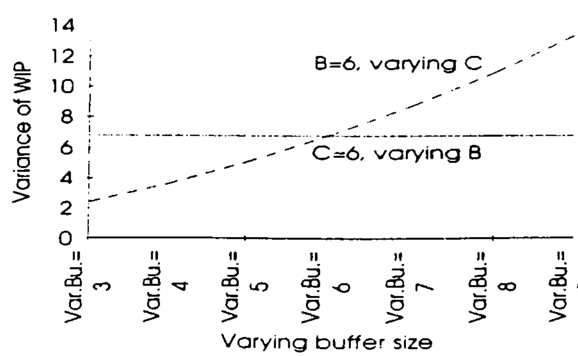

Figure 3.20: Variance of WIP for K=3, λ=1, μ_1 = 2/unit time, μ_2 = 3/unit time, μ_3 = 4/unit time, and varying buffer sizes

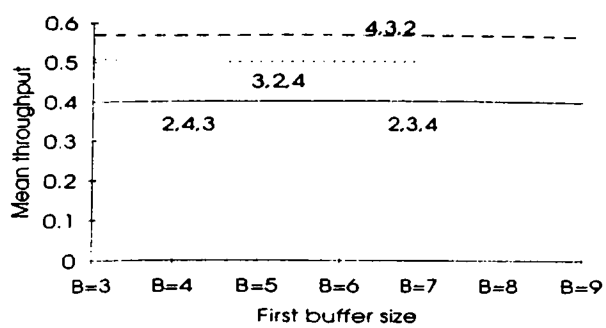

Figure 3.21: Mean throughput for K=3, C=6, λ=1, and varying buffer sizes and machine processing rates

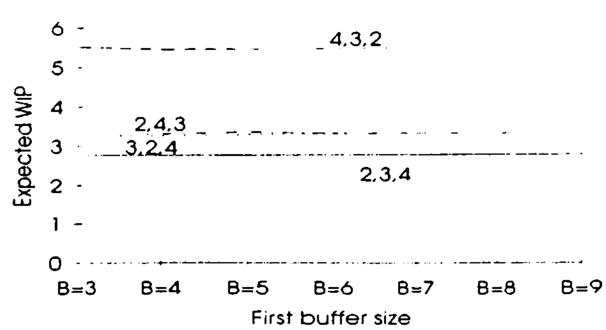

Figure 3.22: Expected WIP for K=3, C=6, λ=1, and varying buffer sizes and machine processing rates

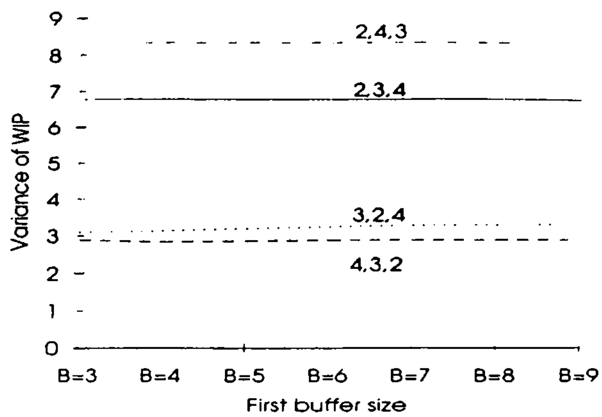

Figure 3.23: Variance of WIP for K=3, C=6, λ=1, and varying buffer sizes and machine processing rates

$K = 2$, $\lambda = 1/unit\ time$, $\mu_1, \mu_2, \mu_3 = 2/unit\ time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 3 **C = 3**	m/c 1: 0.49973 m/c 2: 0.49974 m/c 3: 0.49972	3.05119	1.71888	0.49972
B = 4 **C = 4**	m/c 1: 0.49995 m/c 2: 0.49994 m/c 3: 0.49994	3.22636	2.85004	0.49994
B = 5 **C = 5**	m/c 1: 0.49998 m/c 2: 0.49998 m/c 3: 0.49998	3.39548	4.25388	0.49998
B = 6 **C = 6**	m/c 1: 0.49999 m/c 2: 0.49997 m/c 3: 0.49997	3.56303	5.93383	0.49997
B = 7 **C = 7**	m/c 1: 0.49998 m/c 2: 0.49997 m/c 3: 0.49997	3.73032	7.89185	0.49997
B = 8 **C = 8**	m/c 1: 0.49999 m/c 2: 0.49997 m/c 3: 0.49997	3.89759	10.12822	0.49997
B = 9 **C = 9**	m/c 1: 0.49999 m/c 2: 0.49999 m/c 3: 0.49999	4.06483	12.64311	0.49997

Table 3.1: Performance measures for K=2, μ = 2 /unit time, varying but identical buffer sizes

$K = 2$, $\lambda = 1/unit\ time$, $\mu_1, \mu_2, \mu_3 = 3/unit\ time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 3 **C = 3**	m/c 1: 0.39996 m/c 2: 0.39996 m/c 3: 0.39995	2.24963	2.35495	0.59992
B = 4 **C = 4**	m/c 1: 0.40000 m/c 2: 0.39999 m/c 3: 0.39999	2.37015	3.35739	0.59992
B = 5 **C = 5**	m/c 1: 0.39999 m/c 2: 0.39999 m/c 3: 0.39999	2.48963	4.56783	0.59998
B = 6 **C = 6**	m/c 1: 0.39999 m/c 2: 0.40000 m/c 3: 0.39999	2.60897	5.98791	0.59998
B = 7 **C = 7**	m/c 1: 0.39999 m/c 2: 0.40000 m/c 3: 0.39999	2.72826	7.61813	0.59998
B = 8 **C = 8**	m/c 1: 0.39999 m/c 2: 0.40000 m/c 3: 0.39999	2.84760	9.45851	0.59998
B = 9 **C = 9**	m/c 1: 0.40000 m/c 2: 0.40000 m/c 3: 0.39999	2.96695	11.50923	0.59998

Table 3.2: Performance measures for K=2, μ = 3 /unit time, varying but identical buffer sizes

$K = 3,\ \lambda = 1/unit\,time,\ \mu_1, \mu_2, \mu_3 = 2/unit\,time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 4 **C = 4**	m/c 1: 0.59994	3.95232	1.35631	0.39995
	m/c 2: 0.59993			
	m/c 3: 0.59993			
B = 5 **C = 5**	m/c 1: 0.59996	4.16361	2.79834	0.39997
	m/c 2: 0.59996			
	m/c 3: 0.59995			
B = 6 **C = 6**	m/c 1: 0.59996	4.37328	4.57078	0.39996
	m/c 2: 0.59995			
	m/c 3: 0.59994			
B = 7 **C = 7**	m/c 1: 0.59996	4.58278	6.67388	0.39997
	m/c 2: 0.59995			
	m/c 3: 0.59995			
B = 8 **C = 8**	m/c 1: 0.59996	4.79211	9.10815	0.39996
	m/c 2: 0.59995			
	m/c 3: 0.59994			
B = 9 **C = 9**	m/c 1: 0.59837	4.97557	11.87326	0.39991
	m/c 2: 0.59994			
	m/c 3: 0.59834			

Table 3.3: Performance measures for K=3, μ = 2 /unit time, varying but identical buffer sizes

$K = 3,\ \lambda = 1/unit\,time,\ \mu_1, \mu_2, \mu_3 = 3/unit\,time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughpu
B = 4 **C = 4**	m/c 1: 0.49998	3.17167	2.85485	0.49997
	m/c 2: 0.49998			
	m/c 3: 0.49997			
B = 5 **C = 5**	m/c 1: 0.49998	3.34598	4.33281	0.49997
	m/c 2: 0.49997			
	m/c 3: 0.49997			
B = 6 **C = 6**	m/c 1: 0.49998	3.52020	6.09833	0.49997
	m/c 2: 0.49997			
	m/c 3: 0.49997			
B = 7 **C = 7**	m/c 1: 0.49999	3.69443	8.15162	0.49997
	m/c 2: 0.49997			
	m/c 3: 0.49997			
B = 8 **C = 8**	m/c 1: 0.49998	3.86866	10.49251	0.49997
	m/c 2: 0.49998			
	m/c 3: 0.49997			
B = 9 **C = 9**	m/c 1: 0.49998	4.04285	13.12134	0.49997
	m/c 2: 0.49998			
	m/c 3: 0.49998			

Table 3.4: Performance measures for K=3, μ = 3 /unit time, varying but identical buffer sizes

$K = 3,\ \lambda = 1/unit\,time,\ \mu_1, \mu_2, \mu_3 = 2/unit\,time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 3 **C = 4**	m/c 1: 0.59971 m/c 2: 0.59972 m/c 3: 0.59972	3.94940	1.35150	0.39981
B = 4 **C = 3**	m/c 1: 0.59995 m/c 2: 0.59991 m/c 3: 0.59998	3.73677	0.25311	0.39992
B = 3 **C = 5**	m/c 1: 0.59970 m/c 2: 0.59971 m/c 3: 0.59969	4.15993	2.79363	0.39979
B = 5 **C = 3**	m/c 1: 0.59999 m/c 2: 0.59995 m/c 3: 0.59991	3.73735	0.25530	0.39994
B = 3 **C = 6**	m/c 1: 0.59970 m/c 2: 0.59971 m/c 3: 0.59969	4.36964	4.56815	0.39979
B = 6 **C = 3**	m/c 1: 0.59998 m/c 2: 0.59996 m/c 3: 0.59991	3.73744	0.25576	0.39994
B = 3 **C = 7**	m/c 1: 0.59970 m/c 2: 0.59970 m/c 3: 0.59969	4.57923	6.67416	0.39979
B = 7 **C = 3**	m/c 1: 0.59998 m/c 2: 0.59994 m/c 3: 0.59990	3.73739	0.25590	0.39993
B = 3 **C = 8**	m/c 1: 0.59970 m/c 2: 0.59970 m/c 3: 0.59969	4.78870	9.11130	0.39979
B = 8 **C = 3**	m/c 1: 0.59998 m/c 2: 0.59995 m/c 3: 0.59991	3.73741	0.25579	0.39994
B = 3 **C = 9**	m/c 1: 0.59970 m/c 2: 0.59969 m/c 3: 0.59969	4.99832	11.87986	0.39979
B = 9 **C = 3**	m/c 1: 0.59998 m/c 2: 0.59995 m/c 3: 0.59991	3.73741	0.25579	0.39994

Table 3.5: Performance measures for K=3, μ = 2 /unit time, varying buffer sizes

$K=3,\ \lambda = 10/unit\,time,\ \mu_1, \mu_2, \mu_3 = 2/unit\,time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 3 **C = 4**	m/c 1: 0.88492 m/c 2: 0.88734 m/c 3: 0.87700	5.84431	0.68977	0.58467
B = 4 **C = 3**	m/c 1: 0.90190 m/c 2: 0.88315 m/c 3: 0.86019	5.65212	1.26821	0.57346
B = 3 **C = 5**	m/c 1: 0.88491 m/c 2: 0.89315 m/c 3: 0.88383	6.17855	0.89020	0.58922
B = 5 **C = 3**	m/c 1: 0.91264 m/c 2: 0.88528 m/c 3: 0.85847	5.85120	1.94654	0.57231
B = 3 **C = 6**	m/c 1: 0.88648 m/c 2: 0.89643 m/c 3: 0.88488	6.38906	1.16364	0.58992
B = 6 **C = 3**	m/c 1: 0.92000 m/c 2: 0.87571 m/c 3: 0.87435	6.11289	2.44452	0.58290
B = 3 **C = 7**	m/c 1: 0.88469 m/c 2: 0.89801 m/c 3: 0.90460	6.93156	2.05035	0.60306
B = 7 **C = 3**	m/c 1: 0.92437 m/c 2: 0.89280 m/c 3: 0.87536	6.27304	3.46149	0.58357
B = 3 **C = 8**	m/c 1: 0.88895 m/c 2: 0.89379 m/c 3: 0.90155	7.07223	3.16398	0.60103
B = 8 **C = 3**	m/c 1: 0.92450 m/c 2: 0.89180 m/c 3: 0.87540	6.43605	4.30758	0.58360
B = 3 **C = 9**	m/c 1: 0.88998 m/c 2: 0.89456 m/c 3: 0.90009	7.22336	4.10163	0.60006
B = 9 **C = 3**	m/c 1: 0.92450 m/c 2: 0.89186 m/c 3: 0.87560	6.61222	5.50126	0.58373

Table 3.6: Performance measures for K=3, λ = 10/unit time, μ = 2 /unit time, varying buffer sizes

$K = 3$, $\lambda = 1/unit\ time$, $\mu_1, \mu_2, \mu_3 = 2/unit\ time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 4 C = 5	m/c 1: 0.59993 m/c 2: 0.59992 m/c 3: 0.59991	4.16297	2.79084	0.39994
B = 5 C = 4	m/c 1: 0.59998 m/c 2: 0.59998 m/c 3: 0.59996	3.95294	1.35831	0.39997
B = 4 C = 6	m/c 1: 0.59992 m/c 2: 0.59991 m/c 3: 0.59991	4.37260	4.56930	0.39994
B = 6 C = 4	m/c 1: 0.59998 m/c 2: 0.59997 m/c 3: 0.59996	3.95301	1.35902	0.39997
B = 4 C = 7	m/c 1: 0.59992 m/c 2: 0.59993 m/c 3: 0.59991	4.58203	6.67262	0.39994
B = 7 C = 4	m/c 1: 0.59997 m/c 2: 0.59998 m/c 3: 0.59997	3.95305	1.35943	0.39998
B = 4 C = 8	m/c 1: 0.59992 m/c 2: 0.59992 m/c 3: 0.59991	4.79143	9.10769	0.39994
B = 8 C = 4	m/c 1: 0.59997 m/c 2: 0.59996 m/c 3: 0.59995	3.95295	1.35911	0.39997
B = 4 C = 9	m/c 1: 0.59992 m/c 2: 0.59992 m/c 3: 0.59991	5.00092	11.87354	0.39994
B = 9 C = 4	m/c 1: 0.59997 m/c 2: 0.59996 m/c 3: 0.59995	3.95296	1.35911	0.39997

Table 3.7: Performance measures for K=3, μ = 2 /unit time, varying buffer sizes

$K = 3$, $\lambda = 1/unit\ time$, $\mu_1, \mu_2, \mu_3 = 2/unit\ time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 5 C = 6	m/c 1: 0.59996 m/c 2: 0.59995 m/c 3: 0.59994	4.37325	4.57039	0.39996
B = 6 C = 5	m/c 1: 0.59996 m/c 2: 0.59996 m/c 3: 0.59995	4.16365	2.79879	0.39997
B = 5 C = 7	m/c 1: 0.59996 m/c 2: 0.59995 m/c 3: 0.59994	4.58266	6.67355	0.39996
B = 7 C = 5	m/c 1: 0.59996 m/c 2: 0.59995 m/c 3: 0.59994	4.16362	2.79884	0.39996
B = 5 C = 8	m/c 1: 0.59996 m/c 2: 0.59994 m/c 3: 0.59994	4.79207	9.10809	0.39996
B = 8 C = 5	m/c 1: 0.59996 m/c 2: 0.59995 m/c 3: 0.59994	4.16364	2.79885	0.39996
B = 5 C = 9	m/c 1: 0.59996 m/c 2: 0.59995 m/c 3: 0.59994	5.00143	11.87347	0.39996
B = 9 C = 5	m/c 1: 0.59996 m/c 2: 0.59996 m/c 3: 0.59994	4.16364	2.79888	0.39996

Table 3.8: Performance measures for K=3, μ = 2 /unit time, varying buffer sizes

$K = 3$, $\lambda = 1/unit\,time$, $\mu_1, \mu_2, \mu_3 = 2/unit\,time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 6 **C = 7**	m/c 1: 0.59996 m/c 2: 0.59995 m/c 3: 0.59995	4.58282	6.67385	0.39997
B = 7 **C = 6**	m/c 1: 0.59996 m/c 2: 0.59995 m/c 3: 0.59994	4.37326	4.57079	0.39996
B = 6 **C = 8**	m/c 1: 0.59996 m/c 2: 0.59996 m/c 3: 0.59995	4.79222	9.10805	0.39997
B = 8 **C = 6**	m/c 1: 0.59996 m/c 2: 0.59995 m/c 3: 0.59994	4.37328	4.57074	0.39996
B = 6 **C = 9**	m/c 1: 0.59996 m/c 2: 0.59995 m/c 3: 0.59995	5.00159	11.87337	0.39997
B = 9 **C = 6**	m/c 1: 0.59996 m/c 2: 0.59995 m/c 3: 0.59994	4.37326	4.57079	0.39996

Table 3.9: Performance measures for K=3, μ = 2 /unit time, varying buffer sizes

$K = 3$, $\lambda = 1/unit\,time$, $\mu_1, \mu_2, \mu_3 = 2/unit\,time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 7 **C = 8**	m/c 1: 0.59996 m/c 2: 0.59995 m/c 3: 0.59994	4.79206	9.10824	0.39996
B = 8 **C = 7**	m/c 1: 0.59996 m/c 2: 0.59994 m/c 3: 0.59994	5.48267	6.67401	0.39996
B = 7 **C = 9**	m/c 1: 0.59996 m/c 2: 0.59995 m/c 3: 0.59994	5.00151	11.87352	0.39996
B = 9 **C = 7**	m/c 1: 0.59996 m/c 2: 0.59995 m/c 3: 0.59995	4.58276	6.67391	0.39997

Table 3.10: Performance measures for K=3, μ = 2 /unit time, varying buffer sizes

$K = 3$, $\lambda = 1/unit\,time$, $\mu_1, \mu_2, \mu_3 = 2/unit\,time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 8 **C = 9**	m/c 1: 0.59996 m/c 2: 0.59994 m/c 3: 0.59994	5.00147	11.87363	0.39996
B = 9 **C = 8**	m/c 1: 0.59996 m/c 2: 0.59995 m/c 3: 0.59993	4.79201	9.10819	0.39995

Table 3.11: Performance measures for K=3, μ = 2 /unit time, varying buffer sizes

$K = 3$, $\lambda = 1/unit\ time$, $\mu_1 = 2/unit\ time$, $\mu_2 = 3/unit\ time$, $\mu_3 = 4/unit\ time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 3 **C = 6**	m/c 1: 0.59999 m/c 2: 0.39998 m/c 3: 0.29999	2.77134	6.78317	0.39997
B = 6 **C = 3**	m/c 1: 0.59999 m/c 2: 0.39998 m/c 3: 0.29999	2.32942	2.40859	0.39997
B = 4 **C = 6**	m/c 1: 0.59999 m/c 2: 0.39998 m/c 3: 0.29999	2.77133	6.78318	0.39997
B = 6 **C = 4**	m/c 1: 0.59999 m/c 2: 0.39998 m/c 3: 0.29999	2.47672	3.61557	0.39997
B = 5 **C = 6**	m/c 1: 0.59999 m/c 2: 0.39998 m/c 3: 0.29998	2.77133	6.78318	0.39997
B = 6 **C = 5**	m/c 1: 0.59999 m/c 2: 0.39998 m/c 3: 0.29999	2.62402	5.07375	0.39997
B = 6 **C = 6**	m/c 1: 0.59999 m/c 2: 0.39999 m/c 3: 0.29999	2.77133	6.78318	0.39997
B = 7 **C = 6**	m/c 1: 0.59999 m/c 2: 0.39998 m/c 3: 0.29999	2.77133	6.78318	0.39997
B = 6 **C = 7**	m/c 1: 0.59999 m/c 2: 0.39998 m/c 3: 0.29999	2.91863	8.74380	0.39997
B = 8 **C = 6**	m/c 1: 0.59999 m/c 2: 0.39998 m/c 3: 0.29999	2.77132	6.78315	0.39997
B = 6 **C = 8**	m/c 1: 0.59999 m/c 2: 0.39998 m/c 3: 0.29999	3.06594	10.95563	0.39997
B = 9 **C = 6**	m/c 1: 0.59999 m/c 2: 0.39998 m/c 3: 0.29999	2.77132	6.78315	0.39997
B = 6 **C = 9**	m/c 1: 0.59999 m/c 2: 0.39998 m/c 3: 0.29999	3.21324	13.41860	0.39997

Table 3.12: Performance measures for K=3, $\lambda = 1$ /unit time, $\mu_1 < \mu_2 < \mu_3$, and varying buffer sizes

$K = 3$, $\lambda = 1/unit\ time$, $\mu_1 = 4/unit\ time$, $\mu_2 = 3/unit\ time$, $\mu_3 = 2/unit\ time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 3 **C = 6**	m/c 1: 0.42820 m/c 2: 0.57114 m/c 3: 0.85414	5.51679	2.88643	0.56943
B = 6 **C = 3**	m/c 1: 0.42895 m/c 2: 0.56417 m/c 3: 0.83583	4.36783	0.34618	0.55722
B = 4 **C = 6**	m/c 1: 0.42860 m/c 2: 0.56973 m/c 3: 0.85007	5.45188	2.83933	0.56671
B = 6 **C = 4**	m/c 1: 0.42879 m/c 2: 0.56694 m/c 3: 0.84230	4.77095	0.83785	0.56153
B = 5 **C = 6**	m/c 1: 0.42863 m/c 2: 0.56962 m/c 3: 0.85005	5.45609	2.87377	0.56670
B = 6 **C = 5**	m/c 1: 0.42716 m/c 2: 0.57663 m/c 3: 0.86888	5.20027	1.70465	0.57925
B = 6 **C = 6**	m/c 1: 0.42862 m/c 2: 0.56962 m/c 3: 0.84988	5.45632	2.89207	0.56659
B = 7 **C = 6**	m/c 1: 0.42862 m/c 2: 0.56962 m/c 3: 0.84990	5.45705	2.89819	0.56660
B = 6 **C = 7**	m/c 1: 0.42860 m/c 2: 0.57026 m/c 3: 0.85207	5.75337	4.43026	0.56805
B = 8 **C = 6**	m/c 1: 0.42862 m/c 2: 0.56962 m/c 3: 0.84990	5.45726	2.90046	0.56660
B = 6 **C = 8**	m/c 1: 0.42855 m/c 2: 0.57143 m/c 3: 0.85564	6.06768	6.28685	0.57043
B = 9 **C = 6**	m/c 1: 0.42861 m/c 2: 0.56961 m/c 3: 0.84990	5.45728	2.90131	0.56660
B = 6 **C = 9**	m/c 1: 0.42852 m/c 2: 0.57108 m/c 3: 0.85509	6.28824	8.46648	0.57006

Table 3.13: Performance measures for K=3, $\lambda = 1$ /unit time, $\mu_1 > \mu_2 > \mu_3$, and varying buffer sizes

$K = 3$, $\lambda = 1/unit\,time$, $\mu_1 = 3/unit\,time$, $\mu_2 = 2/unit\,time$, $\mu_3 = 4/unit\,time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 3 **C = 6**	m/c 1: 0.49632 m/c 2: 0.75360 m/c 3: 0.38135	3.27201	3.09792	0.50847
B = 6 **C = 3**	m/c 1: 0.49973 m/c 2: 0.74960 m/c 3: 0.37478	3.08069	1.33024	0.49971
B = 4 **C = 6**	m/c 1: 0.49849 m/c 2: 0.74774 m/c 3: 0.37386	3.26294	3.13643	0.49848
B = 6 **C = 4**	m/c 1: 0.49973 m/c 2: 0.74559 m/c 3: 0.37478	3.15526	1.83488	0.49971
B = 5 **C = 6**	m/c 1: 0.49937 m/c 2: 0.74906 m/c 3: 0.37452	3.29061	3.20734	0.49936
B = 6 **C = 5**	m/c 1: 0.49973 m/c 2: 0.74559 m/c 3: 0.37478	3.22983	2.47756	0.49971
B = 6 **C = 6**	m/c 1: 0.49973 m/c 2: 0.74959 m/c 3: 0.37478	3.30450	3.25867	0.49971
B = 7 **C = 6**	m/c 1: 0.49986 m/c 2: 0.74981 m/c 3: 0.37487	3.31118	3.29107	0.49983
B = 6 **C = 7**	m/c 1: 0.49973 m/c 2: 0.74959 m/c 3: 0.37478	3.37912	4.17770	0.49971
B = 8 **C = 6**	m/c 1: 0.49993 m/c 2: 0.74990 m/c 3: 0.37493	3.31445	3.31009	0.49991
B = 6 **C = 8**	m/c 1: 0.49973 m/c 2: 0.74959 m/c 3: 0.37478	3.45374	5.23485	0.49971
B = 9 **C = 6**	m/c 1: 0.49989 m/c 2: 0.74994 m/c 3: 0.37489	3.31518	3.31962	0.49985
B = 6 **C = 9**	m/c 1: 0.49973 m/c 2: 0.74959 m/c 3: 0.37478	3.52836	6.43011	0.49971

Table 3.14: Performance measures for K=3, $\lambda = 1$ /unit time, $\mu_1 > \mu_2 < \mu_3$, and varying buffer sizes

$K = 3$, $\lambda = 1/unit\ time$, $\mu_1 = 2/unit\ time$, $\mu_2 = 4/unit\ time$, $\mu_3 = 3/unit\ time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 3 **C = 6**	m/c 1: 0.59999 m/c 2: 0.29999 m/c 3: 0.39999	3.31498	8.33663	0.39999
B = 6 **C = 3**	m/c 1: 0.59999 m/c 2: 0.29999 m/c 3: 0.39999	2.69320	2.74580	0.39999
B = 4 **C = 6**	m/c 1: 0.59999 m/c 2: 0.29998 m/c 3: 0.39998	3.31495	8.33661	0.39999
B = 6 **C = 4**	m/c 1: 0.59999 m/c 2: 0.29998 m/c 3: 0.39998	2.90045	4.28083	0.39998
B = 5 **C = 6**	m/c 1: 0.59999 m/c 2: 0.29998 m/c 3: 0.39998	3.31497	8.33662	0.39998
B = 6 **C = 5**	m/c 1: 0.59999 m/c 2: 0.29998 m/c 3: 0.39998	3.10772	6.14441	0.39998
B = 6 **C = 6**	m/c 1: 0.59999 m/c 2: 0.29998 m/c 3: 0.39998	3.31497	8.33662	0.39998
B = 7 **C = 6**	m/c 1: 0.59999 m/c 2: 0.29998 m/c 3: 0.39998	3.31496	8.33661	0.39998
B = 6 **C = 7**	m/c 1: 0.59999 m/c 2: 0.29998 m/c 3: 0.39998	3.55221	10.85740	0.39998
B = 8 **C = 6**	m/c 1: 0.59999 m/c 2: 0.29998 m/c 3: 0.39998	3.31496	8.33661	0.39998
B = 6 **C = 8**	m/c 1: 0.59999 m/c 2: 0.29998 m/c 3: 0.39998	3.72947	13.70679	0.39998
B = 9 **C = 6**	m/c 1: 0.59999 m/c 2: 0.29998 m/c 3: 0.39998	3.31496	8.33661	0.39998
B = 6 **C = 9**	m/c 1: 0.59999 m/c 2: 0.29998 m/c 3: 0.39998	3.93674	16.88482	0.39998

Table 3.15: Performance measures for $K = 3$, $\lambda = 1$ /unit time, $\mu_1 < \mu_2 > \mu_3$, and varying buffer sizes

$K = 2,\ \lambda = 1/unit\ time,\ \mu_1 = \mu_2 = \mu_3 = 2/unit\ time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 2 **C = 3**	m/c 1: 0.49863 m/c 2: 0.49863 m/c 3: 0.49862	3.02789	1.72124	0.49862
B = 3 **C = 2**	m/c 1: 0.49972 m/c 2: 0.49974 m/c 3: 0.49970	2.86469	0.88537	0.49970
B = 2 **C = 4**	m/c 1: 0.49863 m/c 2: 0.49864 m/c 3: 0.49865	3.21166	2.83146	0.49865
B = 4 **C = 2**	m/c 1: 0.49995 m/c 2: 0.49987 m/c 3: 0.49981	2.86671	0.89309	0.49981
B = 2 **C = 5**	m/c 1: 0.49863 m/c 2: 0.49866 m/c 3: 0.49868	3.38036	4.24052	0.49868
B = 5 **C = 2**	m/c 1: 0.49998 m/c 2: 0.50007 m/c 3: 0.50004	2.86920	0.90359	0.50004
B = 2 **C = 6**	m/c 1: 0.49863 m/c 2: 0.49864 m/c 3: 0.49861	3.54752	5.92748	0.49861
B = 6 **C = 2**	m/c 1: 0.49999 m/c 2: 0.50011 m/c 3: 0.50008	2.87010	0.90932	0.50008
B = 2 **C = 7**	m/c 1: 0.49863 m/c 2: 0.49863 m/c 3: 0.49863	3.71538	7.89460	0.49863
B = 7 **C = 2**	m/c 1: 0.49999 m/c 2: 0.49991 m/c 3: 0.49985	2.86754	0.89742	0.49985
B = 2 **C = 8**	m/c 1: 0.49863 m/c 2: 0.49866 m/c 3: 0.49863	3.88268	10.13916	0.49863
B = 8 **C = 2**	m/c 1: 0.50000 m/c 2: 0.49989 m/c 3: 0.49983	2.86752	0.89752	0.49983
B = 2 **C = 9**	m/c 1: 0.49863 m/c 2: 0.49865 m/c 3: 0.49863	4.05032	12.66367	0.49863
B = 9 **C = 2**	m/c 1: 0.50000 m/c 2: 0.49994 m/c 3: 0.49998	2.86770	0.89732	0.49988

Table 3.16: Performance measures for K=2, μ = 2 /unit time, varying buffer sizes

$K=2,\ \lambda = 1/unit\,time,\ \mu_1 = \mu_2 = \mu_3 = 2/unit\,time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 3 **C = 4**	m/c 1: 0.49973 m/c 2: 0.49973 m/c 3: 0.49955	3.22123	2.84249	0.49955
B = 4 **C = 3**	m/c 1: 0.49995 m/c 2: 0.49995 m/c 3: 0.49994	3.05403	1.72760	0.49994
B = 3 **C = 5**	m/c 1: 0.49973 m/c 2: 0.49973 m/c 3: 0.49973	3.39201	4.24637	0.49973
B = 5 **C = 3**	m/c 1: 0.49999 m/c 2: 0.50001 m/c 3: 0.49999	3.05477	1.73078	0.49999
B = 3 **C = 6**	m/c 1: 0.49973 m/c 2: 0.49973 m/c 3: 0.49972	3.55949	5.92769	0.49972
B = 6 **C = 3**	m/c 1: 0.49999 m/c 2: 0.49997 m/c 3: 0.49996	3.05453	1.73009	0.49996
B = 3 **C = 7**	m/c 1: 0.49973 m/c 2: 0.49973 m/c 3: 0.49972	3.72681	7.88744	0.49972
B = 7 **C = 3**	m/c 1: 0.49998 m/c 2: 0.50002 m/c 3: 0.50001	3.05520	1.73414	0.50001
B = 3 **C = 8**	m/c 1: 0.49973 m/c 2: 0.49973 m/c 3: 0.49972	3.89412	10.12577	0.49972
B = 8 **C = 3**	m/c 1: 0.49999 m/c 2: 0.49997 m/c 3: 0.49995	3.05450	1.73014	0.49995
B = 3 **C = 9**	m/c 1: 0.49973 m/c 2: 0.49973 m/c 3: 0.49972	4.06142	12.64271	0.49972
B = 9 **C = 3**	m/c 1: 0.49999 m/c 2: 0.49997 m/c 3: 0.49997	3.05456	1.73023	0.49997

Table 3.17: Performance measures for K=2, $\mu = 2$ /unit time, varying buffer sizes

$K = 4$, $\lambda = 1/unit\,time$, $\mu_1 = \mu_2 = \mu_3 = 2/unit\,time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
B = 5 C = 5	m/c 1: 0.66662 m/c 2: 0.66659 m/c 3: 0.66658	4.53626	1.55442	0.33329
B = 6 C = 6	m/c 1: 0.64402 m/c 2: 0.66657 m/c 3: 0.64398	4.47787	3.74985	0.32199
B = 7 C = 7	m/c 1: 0.66661 m/c 2: 0.66656 m/c 3: 0.66656	4.98003	5.31851	0.33328
B = 8 C = 8	m/c 1: 0.66660 m/c 2: 0.66657 m/c 3: 0.66656	5.20184	7.71824	0.33328
B = 9 C = 9	m/c 1: 0.66661 m/c 2: 0.66656 m/c 3: 0.66656	5.42370	10.46333	0.33328

Table 3.18: Performance measures for K=4, μ = 2 /unit time, varying buffer sizes

$K = 4$, $\lambda = 1/unit\,time$, $\mu_1 = \mu_2 = \mu_3 = 3/unit\,time$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput (parts/unit time)
B = 5 C = 5	m/c 1: 0.57136 m/c 2: 0.57134 m/c 3: 0.57134	3.90583	3.39936	0.42851
B = 6 C = 6	m/c 1: 0.57136 m/c 2: 0.57134 m/c 3: 0.57134	4.11332	5.26289	0.42851
B = 7 C = 7	m/c 1: 0.57136 m/c 2: 0.57134 m/c 3: 0.57134	4.32083	7.45526	0.42851
B = 8 C = 8	m/c 1: 0.57136 m/c 2: 0.57134 m/c 3: 0.57134	4.52833	9.97651	0.42851
B = 9 C = 9	m/c 1: 0.57136 m/c 2: 0.57134 m/c 3: 0.57134	4.73582	12.82676	0.42851

Table 3.19: Performance measures for K=4, μ = 3 /unit time, varying buffer size

$K = 2,\ \lambda = 1/unit\,time,\ B = 2,\ C = 2$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
$\mu_1 = \mu_2 = \mu_3 = 2$ (per unit time)	m/c 1: 0.49860 m/c 2: 0.49862 m/c 3: 0.49854	2.85295	0.85920	0.49854
$\mu_1 = \mu_2 = \mu_3 = 3$ (per unit time)	m/c 1: 0.39969 m/c 2: 0.39969 m/c 3: 0.39967	2.12295	1.55347	0.49951
$\mu_1 = \mu_2 = \mu_3 = 4$ (per unit time)	m/c 1: 0.33324 m/c 2: 0.33323 m/c 3: 0.33322	1.65611	1.63215	0.66644
$\mu_1 = \mu_2 = \mu_3 = 5$ (per unit time)	m/c 1: 0.28567 m/c 2: 0.28567 m/c 3: 0.28566	1.34399	1.52199	0.71415
$\mu_1 = \mu_2 = \mu_3 = 6$ (per unit time)	m/c 1: 0.24998 m/c 2: 0.24997 m/c 3: 0.24997	1.12444	1.37025	0.74988
$\mu_1 = \mu_2 = \mu_3 = 7$ (per unit time)	m/c 1: 0.22222 m/c 2: 0.22221 m/c 3: 0.22221	0.96316	1.22256	0.77774
$\mu_1 = \mu_2 = \mu_3 = 8$ (per unit time)	m/c 1: 0.20000 m/c 2: 0.19998 m/c 3: 0.19998	0.84031	1.09123	0.79992
$\mu_1 = \mu_2 = \mu_3 = 9$ (per unit time)	m/c 1: 0.18182 m/c 2: 0.18180 m/c 3: 0.18181	0.74412	0.97837	0.81814
$\mu_1 = \mu_2 = \mu_3 = 10$ (per unit time)	m/c 1: 0.16667 m/c 2: 0.16666 m/c 3: 0.16666	0.66690	0.88201	0.83330
$\mu_1 = \mu_2 = \mu_3 = 15$ (per unit time)	m/c 1: 0.11765 m/c 2: 0.11765 m/c 3: 0.11764	0.43593	0.57092	0.88230
$\mu_1 = \mu_2 = \mu_3 = 20$ (per unit time)	m/c 1: 0.09091 m/c 2: 0.09090 m/c 3: 0.09090	0.32214	0.41109	0.90900
$\mu_1 = \mu_2 = \mu_3 = 30$ (per unit time)	m/c 1: 0.06250 m/c 2: 0.06250 m/c 3: 0.06249	0.21077	0.25673	0.93735
$\mu_1 = \mu_2 = \mu_3 = 40$ (per unit time)	m/c 1: 0.04762 m/c 2: 0.04762 m/c 3: 0.04762	0.15635	0.18416	0.95240
$\mu_1 = \mu_2 = \mu_3 = 50$ (per unit time)	m/c 1: 0.03846 m/c 2: 0.03846 m/c 3: 0.03846	0.12419	0.14283	0.96150
$\mu_1 = \mu_2 = \mu_3 = 100$ (per unit time)	m/c 1: 0.01961 m/c 2: 0.01960 m/c 3: 0.01960	0.06109	0.06613	0.98000
$\mu_1 = \mu_2 = \mu_3 = 500$ (per unit time)	m/c 1: 0.00398 m/c 2: 0.00398 m/c 3: 0.00398	0.01204	0.01225	0.99500
$\mu_1 = \mu_2 = \mu_3 = 1000$ (per unit time)	m/c 1: 0.00200 m/c 2: 0.00200 m/c 3: 0.00199	0.00601	0.00607	0.99700
$\mu_1 = \mu_2 = \mu_3 = 5000$ (per unit time)	m/c 1: 0.00040 m/c 2: 0.00040 m/c 3: 0.00040	0.00120	0.00120	0.99999
$\mu_1 = \mu_2 = \mu_3 = 10000$ (per unit time)	m/c 1: 0.00020 m/c 2: 0.00020 m/c 3: 0.00020	0.00060	0.00060	0.99999
$\mu_1 = \mu_2 = \mu_3 = 100000$ (per unit time)	m/c 1: 0.00002 m/c 2: 0.00002 m/c 3: 0.00002	0.00006	0.00006	0.99999

Table 3.20: Performance measures for K=2/unit time, λ=1/unit time, and varying processing rates

$K = 3,\ \lambda = 1/unit\ time,\ B = 3,\ C = 3$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
$\mu_1 = \mu_2 = \mu_3 = 2$ (per unit time)	m/c 1: 0.59971 m/c 2: 0.59970 m/c 3: 0.59968	3.73388	0.24604	0.39979
$\mu_1 = \mu_2 = \mu_3 = 3$ (per unit time)	m/c 1: 0.49995 m/c 2: 0.49994 m/c 3: 0.49994	2.99620	1.66350	0.49994
$\mu_1 = \mu_2 = \mu_3 = 4$ (per unit time)	m/c 1: 0.42856 m/c 2: 0.42855 m/c 3: 0.42852	2.44400	2.44453	0.57136
$\mu_1 = \mu_2 = \mu_3 = 5$ (per unit time)	m/c 1: 0.37497 m/c 2: 0.37495 m/c 3: 0.37494	2.03699	2.37338	0.62490
$\mu_1 = \mu_2 = \mu_3 = 10$ (per unit time)	m/c 1: 0.23076 m/c 2: 0.23074 m/c 3: 0.23074	1.05201	1.67399	0.76913
$\mu_1 = \mu_2 = \mu_3 = 15$ (per unit time)	m/c 1: 0.16667 m/c 2: 0.16666 m/c 3: 0.16665	0.68877	1.11557	0.83325
$\mu_1 = \mu_2 = \mu_3 = 20$ (per unit time)	m/c 1: 0.13043 m/c 2: 0.13042 m/c 3: 0.13041	0.50700	0.80068	0.86940
$\mu_1 = \mu_2 = \mu_3 = 30$ (per unit time)	m/c 1: 0.09090 m/c 2: 0.09090 m/c 3: 0.09090	0.32889	0.48729	0.90900
$\mu_1 = \mu_2 = \mu_3 = 40$ (per unit time)	m/c 1: 0.06977 m/c 2: 0.06976 m/c 3: 0.06976	0.24234	0.34022	0.93013
$\mu_1 = \mu_2 = \mu_3 = 50$ (per unit time)	m/c 1: 0.05661 m/c 2: 0.05660 m/c 3: 0.05660	0.19158	0.25790	0.94333
$\mu_1 = \mu_2 = \mu_3 = 100$ (per unit time)	m/c 1: 0.02912 m/c 2: 0.02912 m/c 3: 0.02912	0.09308	0.11146	0.97067
$\mu_1 = \mu_2 = \mu_3 = 500$ (per unit time)	m/c 1: 0.00596 m/c 2: 0.00596 m/c 3: 0.00596	0.01813	0.01896	0.99333
$\mu_1 = \mu_2 = \mu_3 = 1000$ (per unit time)	m/c 1: 0.00299 m/c 2: 0.00299 m/c 3: 0.00299	0.00901	0.00920	0.99667
$\mu_1 = \mu_2 = \mu_3 = 5000$ (per unit time)	m/c 1: 0.00060 m/c 2: 0.00060 m/c 3: 0.00060	0.00180	0.00180	0.99990
$\mu_1 = \mu_2 = \mu_3 = 10000$ (per unit time)	m/c 1: 0.00030 m/c 2: 0.00030 m/c 3: 0.00030	0.00090	0.00090	0.99999
$\mu_1 = \mu_2 = \mu_3 = 100000$ (per unit time)	m/c 1: 0.00003 m/c 2: 0.00003 m/c 3: 0.00003	0.00009	0.00009	0.99999

Table 3.21: Performance measures for K=3/unit time, λ=1/unit time, and varying processing rates

$K = 4,\ \lambda = 1/unit\,time,\ B = 4,\ C = 4$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
$\mu_1 = \mu_2 = \mu_3 = 2$ (per unit time)	m/c 1: 0.66659 m/c 2: 0.66659 m/c 3: 0.66658	4.31269	0.19036	0.33329
$\mu_1 = \mu_2 = \mu_3 = 3$ (per unit time)	m/c 1: 0.57137 m/c 2: 0.57135 m/c 3: 0.57135	3.69833	1.86464	0.42851
$\mu_1 = \mu_2 = \mu_3 = 4$ (per unit time)	m/c 1: 0.49998 m/c 2: 0.49995 m/c 3: 0.49995	3.15307	2.88267	0.49995
$\mu_1 = \mu_2 = \mu_3 = 5$ (per unit time)	m/c 1: 0.44441 m/c 2: 0.44437 m/c 3: 0.44437	2.70466	3.32845	0.55546
$\mu_1 = \mu_2 = \mu_3 = 6$ (per unit time)	m/c 1: 0.39997 m/c 2: 0.39992 m/c 3: 0.39992	2.34481	3.43236	0.59988
$\mu_1 = \mu_2 = \mu_3 = 7$ (per unit time)	m/c 1: 0.36362 m/c 2: 0.36359 m/c 3: 0.36359	2.05601	3.35728	0.63628
$\mu_1 = \mu_2 = \mu_3 = 8$ (per unit time)	m/c 1: 0.33332 m/c 2: 0.33328 m/c 3: 0.33329	1.82195	3.19702	0.66658
$\mu_1 = \mu_2 = \mu_3 = 9$ (per unit time)	m/c 1: 0.30768 m/c 2: 0.30765 m/c 3: 0.30764	1.63011	3.00233	0.69219
$\mu_1 = \mu_2 = \mu_3 = 10$ (per unit time)	m/c 1: 0.28570 m/c 2: 0.28568 m/c 3: 0.28567	1.47104	2.79983	0.71418
$\mu_1 = \mu_2 = \mu_3 = 15$ (per unit time)	m/c 1: 0.21051 m/c 2: 0.21049 m/c 3: 0.21049	0.97029	1.94735	0.78934
$\mu_1 = \mu_2 = \mu_3 = 20$ (per unit time)	m/c 1: 0.16665 m/c 2: 0.16663 m/c 3: 0.16663	0.71341	1.41081	0.83315
$\mu_1 = \mu_2 = \mu_3 = 30$ (per unit time)	m/c 1: 0.11763 m/c 2: 0.11762 m/c 3: 0.11762	0.45952	0.84983	0.88215
$\mu_1 = \mu_2 = \mu_3 = 40$ (per unit time)	m/c 1: 0.09090 m/c 2: 0.09090 m/c 3: 0.09090	0.33640	0.58216	0.90900
$\mu_1 = \mu_2 = \mu_3 = 50$ (per unit time)	m/c 1: 0.07407 m/c 2: 0.07406 m/c 3: 0.07407	0.26450	0.43277	0.92588
$\mu_1 = \mu_2 = \mu_3 = 100$ (per unit time)	m/c 1: 0.03846 m/c 2: 0.03845 m/c 3: 0.03844	0.12672	0.17463	0.96100
$\mu_1 = \mu_2 = \mu_3 = 500$ (per unit time)	m/c 1: 0.00794 m/c 2: 0.00793 m/c 3: 0.00794	0.02430	0.02650	0.99250
$\mu_1 = \mu_2 = \mu_3 = 1000$ (per unit time)	m/c 1: 0.00398 m/c 2: 0.00398 m/c 3: 0.00399	0.01209	0.01270	0.99750
$\mu_1 = \mu_2 = \mu_3 = 5000$ (per unit time)	m/c 1: 0.00080 m/c 2: 0.00080 m/c 3: 0.00080	0.00240	0.00239	0.99990
$\mu_1 = \mu_2 = \mu_3 = 10000$ (per unit time)	m/c 1: 0.00040 m/c 2: 0.00040 m/c 3: 0.00040	0.00120	0.00120	0.99999
$\mu_1 = \mu_2 = \mu_3 = 100000$ (per unit time)	m/c 1: 0.00004 m/c 2: 0.00004 m/c 3: 0.00004	0.00012	0.00012	0.99999

Table 3.22: Performance measures for K=4/unit time, λ=1/unit time, and varying processing rates

$K = 5$, $\lambda = 1/unit\,time$, $B = 5$, $C = 5$				
	Utilization	Expected WIP	Variance of WIP	Mean throughput
$\mu_1 = \mu_2 = \mu_3 = 2$ (per unit time)	m/c 1: 0.71414 m/c 2: 0.71413 m/c 3: 0.71411	4.69638	0.78985	0.28564
$\mu_1 = \mu_2 = \mu_3 = 3$ (per unit time)	m/c 1: 0.62491 m/c 2: 0.62490 m/c 3: 0.62489	4.25242	2.43856	0.37493
$\mu_1 = \mu_2 = \mu_3 = 4$ (per unit time)	m/c 1: 0.55549 m/c 2: 0.55544 m/c 3: 0.55542	3.77273	3.72344	0.44434
$\mu_1 = \mu_2 = \mu_3 = 5$ (per unit time)	m/c 1: 0.49996 m/c 2: 0.49992 m/c 3: 0.49991	3.32833	4.46277	0.49991
$\mu_1 = \mu_2 = \mu_3 = 6$ (per unit time)	m/c 1: 0.45451 m/c 2: 0.45447 m/c 3: 0.45446	2.94301	4.78032	0.54535
$\mu_1 = \mu_2 = \mu_3 = 7$ (per unit time)	m/c 1: 0.41663 m/c 2: 0.41659 m/c 3: 0.41659	2.61719	4.82753	0.58323
$\mu_1 = \mu_2 = \mu_3 = 8$ (per unit time)	m/c 1: 0.38457 m/c 2: 0.38453 m/c 3: 0.38452	2.34325	4.71781	0.61523
$\mu_1 = \mu_2 = \mu_3 = 9$ (per unit time)	m/c 1: 0.35712 m/c 2: 0.35709 m/c 3: 0.35708	2.11267	4.52385	0.64274
$\mu_1 = \mu_2 = \mu_3 = 10$ (per unit time)	m/c 1: 0.33331 m/c 2: 0.33329 m/c 3: 0.33327	1.91728	4.29054	0.66654
$\mu_1 = \mu_2 = \mu_3 = 15$ (per unit time)	m/c 1: 0.24997 m/c 2: 0.24996 m/c 3: 0.24994	1.28059	3.12938	0.74982
$\mu_1 = \mu_2 = \mu_3 = 20$ (per unit time)	m/c 1: 0.19997 m/c 2: 0.19995 m/c 3: 0.19994	0.94290	2.30588	0.79976
$\mu_1 = \mu_2 = \mu_3 = 30$ (per unit time)	m/c 1: 0.14284 m/c 2: 0.14283 m/c 3: 0.14283	0.60465	1.39420	0.85698
$\mu_1 = \mu_2 = \mu_3 = 40$ (per unit time)	m/c 1: 0.11111 m/c 2: 0.11110 m/c 3: 0.11109	0.44007	0.94625	0.88872
$\mu_1 = \mu_2 = \mu_3 = 50$ (per unit time)	m/c 1: 0.09091 m/c 2: 0.09091 m/c 3: 0.09090	0.34422	0.69430	0.90900
$\mu_1 = \mu_2 = \mu_3 = 100$ (per unit time)	m/c 1: 0.04760 m/c 2: 0.04759 m/c 3: 0.04758	0.16231	0.26389	0.95160
$\mu_1 = \mu_2 = \mu_3 = 500$ (per unit time)	m/c 1: 0.00990 m/c 2: 0.00990 m/c 3: 0.00990	0.03057	0.03527	0.99000
$\mu_1 = \mu_2 = \mu_3 = 1000$ (per unit time)	m/c 1: 0.00498 m/c 2: 0.00498 m/c 3: 0.00497	0.01516	0.01651	0.99400
$\mu_1 = \mu_2 = \mu_3 = 5000$ (per unit time)	m/c 1: 0.00100 m/c 2: 0.00100 m/c 3: 0.00100	0.00300	0.00299	0.99990
$\mu_1 = \mu_2 = \mu_3 = 10000$ (per unit time)	m/c 1: 0.00050 m/c 2: 0.00050 m/c 3: 0.00050	0.00150	0.00150	0.99999
$\mu_1 = \mu_2 = \mu_3 = 100000$ (per unit time)	m/c 1: 0.00005 m/c 2: 0.00005 m/c 3: 0.00005	0.00015	0.00015	0.99999

Table 3.23: Performance measures for K=4/unit time, λ=1/unit time, and varying processing rates

Chapter 4

CONCLUSION

In this thesis, we develop a Markov model to calculate some important performance measures of a transfer line with three machines and two finite buffers where processing times of the machines are k-Erlang distributed. The literature is mostly devoted to transfer lines consisting of machines whose processing times are exponentially distributed. We also calculate the variance of WIP as well as the commonly used performance measures.

We perform several experiments to see the effect of important system parameters on the performance measures. Consequently, we arrive conclusions regarding the transfer line design with respect to changing parameters. In the experiments, we varied K; the stage of the Erlang distribution, from 2 to 6, and B and C; buffer sizes, from 2 to 9. For future research, many experiments can be performed to better see and evaluate the relationships between parameters by extending the ranges of these values. Moreover, transfer lines with n machines whose processing times are k-Erlang distributed, and $(n-1)$ buffers can be analyzed. To do this, our solution for three machines can be extended by implementing available decomposition techniques. Moreover, performance measures of such a transfer line can be investigated under the assumption that the processing times of the machines are phase-type distributed. Finally, studies on locating and sizing of buffers can be extended both under the assumptions of our system and the phase-type distributed processing times assumption.

Moreover, we solve the model by generating the balance equations by a computer program coded in C, and then solving these equations using CPLEX. However, one can also try to observe the special properties of the stochastic matrix of the Markov model and exploit them.

We calculate the mean throughput, machine utilizations, expected value of WIP level, and also the variance of WIP. Another important performance measure is the variance of throughput. Hence, as a next step this measure can be calculated for transfer lines under a wide variety of assumptions on processing time distributions and number of machines in the line.

Bibliography

[1] T. Altıok. Approximate analysis of exponential tandem queues with blocking. *Operations Research*, 11, 1982.

[2] T. Altıok and R. Ranjan. Analysis of production lines with general service times and finite buffers: A two-node decomposition approach. Technical report, Department of Industrial and Systems Engineering, Rutgers University, 1987.

[3] M. Berg, M.J.M. Posner, and H. Zhao. Production-inventory systems with unreliable machines. *Operations Research*, 65(1), 1994.

[4] O. Berman. Efficiency and production rate of a transfer line with two machines and a finite storage buffer. *European Journal of Operational Research*, 9:295–308, 1982.

[5] D.E. Blumenfeld. A simple formula for estimating throughput of serial production lines with variable processing times and limited buffer capacity. *International Journal of Production Research*, 28:1163–1182, 1990.

[6] J. Buzacott and J.G. Shantikumar. *Stochastic models of manufacturing systems*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[7] J.A. Buzacott. Prediction of the efficiency of production systems without internal storage. *International Journal of Production Research*, 6:173–188, 1968.

[8] J.A. Buzacott. The effect of station breakdowns and random processing times on the capacity of flow lines. *AIIE Transactions*, 4:308–312, 1972.

[9] J.A. Buzacott and L.E. Hanifin. Transfer line design and analysis - an overview. *IIE Conferences*, 35(6):645–674, July 1978.

[10] J.A. Buzacott and D. Kostelski. Matrix-geometric and recursive algorithm solution of a two-stage unreliable flow line. *IIE Transactions*, 19:429–438, 1987.

[11] Y.F. Choong and S.B. Gershwin. A decomposition method for the approximate evaluation of capacitated transfer lines with unreliable machines and random processing times. *IIE Transactions*, 19:150–159, 1987.

[12] C. Commault and Y. Dallery. Production rate of transfer lines with no intermediate storage. *IIE Transactions*, 22:315–329, 1990.

[13] Y. Dallery, R. David, and X.L. Xie. An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers. *IIE Transactions*, 20:280–283, 1988.

[14] Y. Dallery and S.B. Gershwin. Manufacturing flow line systems: A review of models and analytic results. *Queueing Systems*, 12:3–94, 1992.

[15] S.B. Gershwin. An efficient decomposition algorithm for the approximate evaluation of tandem queues with finite storage space and blocking. *Operations Research*, 35:291–305, 1987.

[16] S.B. Gershwin and O. Berman. Analysis of transfer lines consisting of two unreliable machines with random processing times and finite storage buffers. *AIIE Transactions*, 13, 1981.

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

[18] C.R. Glassey and Y. Hong. Analysis of behaviour of an unreliable $n-$stage transfer line with $(n-1)$ inter-stage storage buffers. *International Journal of Production Research*, 31(3):519–530, 1993.

[19] L. Gun. *Tandem queueing systems subject to blocking with phase-type servers: Analytic solutions and approximations.* PhD thesis. Department of Electronics Engineering, University of Maryland, College Park, 1987.

[20] L. Gun and A.M. Makowski. Matrix-geometric solution for finite capacity queues with phase-type distributions. *Performance '87*, 1987.

[21] L. Gun and A.M Makowski. *Queueing Networks with Blocking*, chapter An approximation method for general tandem queueing systems subject to blocking. pages 147–174. North-Holland, Amsterdam, 1989.

[22] D.K. Hildebrand. On the capacity of tandem server, finite queue, service systems. *Operations Research*, 16:72–82, 1968.

[23] F.S. Hillier and R.W. Boling. Finite queues in series with exponential or erlang service times - a numerical approach. *Operations Research*, 16:286–303, 1967.

[24] F.S. Hillier and K.C. So. The effect of machine breakdowns and interstage storage on the performance of production line systems. *International Journal of Production Research*, 29(10):2043–2055, 1991.

[25] G.C. Hunt. Sequential arrays of waiting lines. *Operations Research*, 4:674–683, 1956.

[26] P.A. Jensen, R. Pakath, and J.R. Wilson. Optimal buffer inventories for multistage production systems with failures. *European Journal of Operational Research*, 51(3):313–326, 1991.

[27] A.D. Knott. The inefficiency of a series of work stations - a simple formula. *International Journal of Production Research*, 8:109–119, 1970.

[28] H. Lau. On balancing variances of station processing times in unpaced lines. *European Journal of Operational Research*, 61(3):345–356, 1992.

[29] H. S. Lau. The production rate of a two-stage system with stochastic processing times. *International Journal of Production Research*, 24:401–412, 1986.

[30] S.S. Lavenberg. The steady-state queueing time distribution for the $m/g/1$ finite capacity queue. *Management Science*, 21:501–506, 1975.

[31] M. Di Mascolo, Y. Frein, and Y. Dallery. An analytical method for performance evaluation of kanban controlled production systems. *Operations Research*, 44(1), 1996.

[32] G.J. Miltenburg. Variance of the number of units produced on a transfer line with buffer inventories during a period of length t. *Naval Research Logistics*, 34:811–822, 1987.

[33] E.J. Muth. Stochastic processes and their network representations associated with a production line queueing model. *Operations Research*, 15:63–83, 1984.

[34] E.J. Muth. An update on analytic models of serial transfer lines. Technical Report Research report 87-15, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, 1987.

[35] E.J. Muth and A. Alkaff. The througput rate of three-station production lines: A unifying solution. *International Journal of Production Research*, 25:1405–1413, 1987.

[36] M.F. Neuts. *Matrix-geometric solutions in stochastic models*. Johns Hopkins University Press, 1981.

[37] H.T. Papadopoulos. The throughput of multistation production lines with no intermediate buffers. *Operations Research*, 43(4), 1995.

[38] H.T. Papadopoulos and M.E.J. O'Kelly. Exact analysis of production lines with no intermediate buffers. *European Journal of Operational Research*, 65(1):118–137, 1993.

[39] T. Park. A two-phase heuristic algorithm for determining buffer sizes of production lines. *International Journal of Production Research*, 31(3):519–530, 1993.

[40] H.G. Perros and T. Altıok. Approximate analysis of open networks of queues with blocking: Tandem configurations. *IEEE Transactions Software Engineering*, SE-12:450–461, 1986.

[41] R. Pike and G.E. Martin. The bowl phenomenon in unpaced lines. *International Journal of Production Research*, 32(3):483–499, 1994.

[42] S.M. Pollock, J.R. Birge, and J.M. Alden. Approximation analysis for open tandem queues with blocking. Technical Report IE report 85-30, University of Michigan, 1985.

[43] N.P. Rao. On the mean production of a two-stage production system of the tandem type. *International Journal of Production Research*, 13:207–217, 1975.

[44] N.P. Rao. Two-stage production systems with intermediate storage. *AIIE Transactions*, 7:414–421, 1975.

[45] N.P. Rao. A viable alternative to the method of stages solution of series production systems with erlang service times. *International Journal of Production Research*, 14:699–702, 1976.

[46] T.L. Smunt and W.C. Perkins. Stochastic unpaced line design: Review and further experimental results. *Journal of Operations Management*, 5:351–373, 1985.

[47] M.C. Springer. A decomposition approximation for finite-buffered flow lines of exponential queues. *European Journal of Operational Research*, 74(1):95–110, 1994.

[48] Y. Takahashi, H. Miyahara, and T. Hasegawa. An approximation method for open restricted queueing networks. *Operations Research*, 1980.

[49] A.P Vladzievski. Losses of working time and the division of automatic lines into sections. *Stanki i Instrument*, 24, 1953. (in Russian) (English Translation : DSIIR CTS 632).