Efficient Buffer Design Algorithms for Production Line Profit Maximization

by

Chuan Shi

B.Eng., Tsinghua University, China (2005)
M.Eng., Tsinghua University, China (2007)

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
FEB 16 2012
LIBRARIES

ARCHIVES

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2012

© Massachusetts Institute of Technology 2012. All rights reserved.

Author .....
Department of Mechanical Engineering
October 31, 2011

Certified by .....
Stanley B. Gershwin
Senior Research Scientist, Department of Mechanical Engineering
Thesis Supervisor

Accepted by .....
David E. Hardt
Chairman, Department Committee on Graduate Students

To Mom, Dad, and Grandparents

Nobody trips over mountains. It is the small pebble that causes you to stumble. Pass all the pebbles in your path and you will find you have crossed the mountain.

— Author unknown

Efficient Buffer Design Algorithms for Production Line Profit Maximization

by

Chuan Shi

Submitted to the Department of Mechanical Engineering
on October 31, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mechanical Engineering

Abstract

A production line is a manufacturing system where machines are connected in series and separated by buffers. The inclusion of buffers increases the average production rate of the line by limiting the propagation of disruptions, but at the cost of additional capital investment, floor space of the line, and inventory. Production lines are also a special case of assembly/disassembly systems as well as closed-loop systems. This thesis makes contributions to production system profit maximization.

The profit of a production line is the revenue associated with the production rate minus the buffer space cost and average inventory holding cost. We assume that machines have already been chosen and therefore our only decision variables are the buffer sizes and the loop population. The difficulties of the research come from evaluation and optimization. We improve evaluation of loop systems. The optimization problem is hard since both the objective function and the constraints are nonlinear. Our optimization problem, where we consider the nonlinear production rate constraint and average inventory cost, is new.

We present an accurate, fast, and reliable algorithm for maximizing profits through buffer space optimization for production lines, and extend the algorithm to closed-loop systems and production lines with an additional maximum part waiting time constraint. A nonlinear programming approach is adopted to solve the optimization problem. Two necessary modifications are proposed to improve the accuracy of the existing loop evaluation method before optimization of loops is studied. An analytical formulation of the part waiting time distribution is developed for two-machine one-buffer lines. It is used in the profit maximization for production lines with both the production rate constraint and the maximum part waiting time constraint. Numerical experiments are provided to show the accuracy and efficiency of the proposed algorithms. Finally, a segmentation method and an additive property of production line optimization are studied. They enable us to optimize very long lines rapidly and accurately.

Thesis Supervisor: Stanley B. Gershwin
Title: Senior Research Scientist, Department of Mechanical Engineering

Doctoral Thesis Committee

Stanley B. Gershwin (chair)

Senior Research Scientist, Department of Mechanical Engineering

Stephen C. Graves

Abraham J. Siegel Professor of Management Science,

Professor of Mechanical Engineering and Engineering Systems

Brian W. Anthony

MIT Director, SMA-Manufacturing Systems and Technology Program,

Research Scientist, Department of Mechanical Engineering

Acknowledgments

There are many people to whom I want to express my deepest acknowledgement for their generous help to me, in many forms, along my PhD journey at MIT.

First and foremost, I would like to thank my advisor Dr. Stanley B. Gershwin for being a fantastic mentor and friend. His trust, encouragement, inspiration, and patience are my treasures along the journey. When I was admitted to MIT, I tried to find a perfect combination between mathematics and manufacturing as my research area, because these two areas are where my passion roots deeply. Lucky enough, I found manufacturing systems engineering. Even luckier, I found Stan, without whom I could not have achieved so much. Special thanks to Professor Stephen C. Graves and Dr. Brian W. Anthony who agreed to be my thesis committee members willingly. They always provided me with their valuable suggestions and insightful discussions.

Next, I would like to thank to my family. Thanks to my mom, dad, and grandparents for their endless support and love which makes me feel that they are always around, even given the fact that they are on the other side of the earth. Furthermore, special thanks to my unique Mo Zhou for her affection and support that make me feel that I am never alone.

Thanks also go to my colleagues (both MIT scholars and visiting scholars to MIT), Irvin Schick, Fernando Tubilla, Marcello Colledani, Alain Patchong, Zheng Wang, Elisa Gebennini, Andrea Ratti, Sumant Raykar, Adam Traina, and Rong Yuan for their thoughtful comments. In addition, I found that the theses of James E. Schor, Loren M. Werner, and Zhenyu Zhang were very insightful for my research.

I would express my gratitude to Professor David E. Hardt, Leslie Regan, Karuna Mohindra, and David Rodriguera in the Department of Mechanical Engineering, to Professor John N. Tsitsiklis of Department of Electrical Engineering and Computer Science, and to Danielle Guichard-Ashbrook of the International Student Office. Their help for many different reasons were of tremendous value to make my study and life at MIT a lot easier.

I feel very grateful to be surrounded by numerous fabulous friends, and they make

my time at MIT and Boston area enjoyable and unforgettable. Thanks to Rong Xiao, Meng Luo, Huafei Sun, Kai Liao, Ning Zhai, Maokai Lin, Xiao Shen, Chiwon Kim, Dongfang Bai, Feng Tan, Wener Lv, Aifeng Tao, Anyang Hou, Ming Xiong, Huiping Zhou, and many more. Thanks to all my students at MIT 2.853/4 Classes 2007, 2009, and 2010 for their great support to me as a teaching assistant. I enjoyed sharing with and learning from you all very much.

I want to thank Dr. Yan Pang of IBM for his inspirational discussion with me about my research. I also want to give special thanks to Dr. Steve Honkomp of Procter & Gamble and Dr. Rajesh Jugulum of Citigroup, Inc. for the great summer internship opportunities they offered me. I am very proud of the achievements that we have made.

Finally, this work has been supported financially by the MIT-Portugal Program and the Singapore-MIT Alliance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

A *manufacturing system* is a set of machines, transportation elements, computers, storage buffers, and other items that are used together for manufacturing (Gershwin 1994). A *production line*, or *flow line*, or *transfer line*, is organized with machines connected in series and separated by buffers. Figure 1-1, for instance, shows a six-machine five-buffer line, where squares represent machines (or sequences of machines without buffers) while circles represent buffers. (In the following, we treat a sequence of machines without buffers as a single machine M_i.) Therefore, a production line that has k machines will have $k-1$ buffers, and it is called a k-machine, $k-1$-buffer line, or k-machine line for short. Material flows in the direction of the arrows, from upstream inventory to the first machine for an operation, to the first buffer where it waits for the second machine, to the second machine, etc. There are two quantities associated with each buffer. N_i is the size of Buffer B_i and $\bar{n}_i$ is the average inventory of Buffer B_i.

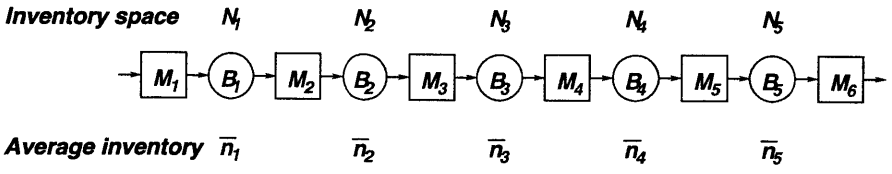


Figure 1-1: A production line example

Production lines are of economic importance as they are used in high volume manufacturing, particularly automobile production. Their capital costs range from hundreds of thousands to tens of millions of dollars. In addition, production lines represent the simplest form of an important phenomenon: manufacturing stages interfering with each other and buffers decoupling them. This is because machines are unreliable and therefore material flow may be disrupted by machine failures. Such failures can cause neighboring machines to be idle, and they, in turn, can create idleness to their neighbors. The inclusion of buffers increases the average production rate of the line by limiting the propagation of disruptions, but at the cost of additional capital investment, floor space of the line, and inventory (Shi and Gershwin 2011b). As indicated in Gershwin (1994) and Dallery (1999), the behavior of such production line systems is complex because of the random nature of machine failures and repairs and their effect on the whole production line due to blocking and starvation. Therefore, the performance analysis that evaluates the production rate, the average inventory level, as well as the profit rate is of high importance in the design and operation of production lines.

The development of analytical methods for performance evaluation of production lines in the past several decades enables people to understand the behavior of such systems. Although simulation can be an alternative to analytical methods in production line evaluation, the speed and accuracy of analytical methods allow production line practitioners to make robust decisions in line design faster, which is especially important for products with short life cycles or for companies in highly competitive market environment. As a result, analytical methods have been wildly used in industry. Burman et al. (1998), for example, apply analytical methods to predict capacity and to determine the sizes and locations of buffers that would increase capacity at the cost of a minor increase in inventory for a Hewlett-Packard (HP) production line. HP's implementation of this work yields incremental revenue of about $280 million. A detailed literature review about production line evaluation as well as optimization by analytical methods is provided in Section 1.2. A summary of the research about design and operation of production lines and general manufacturing systems conducted

at MIT can be found from Gershwin (2003) and Gershwin (2005). The analytical approach is adopted in this thesis.

Koenigsberg (1959) points out that three major problems in the design and operation of production lines are to select the number of machines in the line, the locations of buffers, and the sizes of buffers. These three problems can be categorized into two main phases (Borgh 2009a) in production line design: first, a small number of system alternatives are selected among a wide range of options (Problems 1 and 2), then the characteristics of these systems are more deeply investigated in order to find the most suitable solution toward a specific goal of line design (Problem 3).

Among the three production line design problems of Koenigsberg (1959), we are particularly interested in the third one, as buffer allocation is one of the most important decisions to make in production line design. As indicated previously, buffers decouple machines and therefore increase the production rate of the line, however, at the cost of increasing buffer space and higher *work-in-process inventory* levels. Inventory is a undesirable consequence of buffers for several reasons. First, it costs money to create or store. Second, the average lead time is proportional to the average amount of inventory according to Little's Law (Little 1961). This means that a larger inventory level may lead to a longer lead time. Third, inventory in a factory can be vulnerable to damage, which projects a potential lost of investment. Finally, the space and equipment needed for inventory costs money as well. Given these additional costs in terms of both extra capital investment and longer lead time of products due to inventory, it is highly desirable to find ways to optimize buffer space allocation to make factories most efficient and most profitable.

Once the buffer allocation problem is solved, we can make use of the result to solve problems 1 and 2 of Koenigsberg (1959). That is, for each possible setting of the production line (with a certain number of machines and a certain configuration of buffer locations), we solve the buffer allocation problem. After the buffer allocation problem is solved for all line configuration candidates, we compare them and choose one that best suits the line design objective.

In this thesis, we develop optimal buffer allocation algorithms for production line

profit maximization subject to a production rate constraint. In other words, we want to help factories determine how to achieve the required production rate target (which is related to the demand) at the minimum cost of buffer spaces and work-in-process inventory. In particular, we assume that the manufacturing process and machines have already been chosen. As a result, the decision variables are sizes of buffer spaces N_1, N_2, $\cdots$, N_{k-1}, or $\mathbf{N}$ in the vector form, for a k-machine $k-1$-buffer line. Production line cost comes from buffer space cost and average inventory cost.

The rest of this chapter is organized as follows. A detailed literature review about production line models, production line evaluation, and production line optimization is provided in Section 1.2. The research goal and contributions of this thesis are summarized in Section 1.3. We outline the structure of the thesis in Section 1.4 before moving on to Chapter 2.

1.2 Literature Review

Substantial research has been conducted on production line evaluation and optimization. See reviews by Koenigsberg (1959), Buxey et al. (1973), Buzacott and Hanifin (1978), Dallery and Gershwin (1992), Papadopoulos and Heavey (1996), and Li et al. (2009), as well as books by Buzacott and Shanthikumar (1993), Papadopoulos et al. (1993), Gershwin (1994), and Altiok (1997). Production line evaluation has been done by both the exact solutions for two-machine lines and the approximation approaches for longer lines with more than two machines. In terms of optimization, there are many studies focusing on maximizing the production rate but few studies concentrating on maximizing the profit. In production line optimization, there are two distinct approaches: the simulation-based approach and the numerical evaluation approach. It is desirable to develop numerical methods since they are much faster than simulation. For description of simulation methods, see Smunt and Perkins (1985) and Gershwin and Schor (2000). We describe some literature on non-simulation methods.

In what follows, we introduce commonly used production line models in Section 1.2.1. Then, we provide a review of major results about production line evaluation

in Section 1.2.2. Finally, we comment on some work that deals with the production line optimization in Section 1.2.3.

1.2.1 Production Line Models

Dallery and Gershwin (1992) discuss production line models as well as their features in great detail. In addition, Gershwin (1994) covers three production line models. They are the *deterministic processing time and discrete material model* (or *deterministic model* for short), the *exponential processing time discrete material model* (or *exponential model* for short), and the *continuous processing time and continuous material model* (or *continuous model* for short). Next, we first comment on two key features of any given production line model, and then we will brief explain the three models.

Blocking Mechanisms. Since buffers between two adjacent machines are of finite capacity, it is possible that a buffer gets full due to a failure of the downstream machine and therefore the upstream machine is forced to stop even if it does not fail. This phenomenon is called the blocking of the upstream machine. As indicated in Dallery and Gershwin (1992), different blocking mechanisms are of interest. In particular, they are *blocking-after-service* and *blocking-before-service* (Perros 1990).

Blocking-after-service is also referred to as *type 1 blocking* (Onvural and Perros 1986), *manufacturing blocking*, *production blocking*, *transfer blocking*, and *non-immediate blocking* (Gün and Makowski 1989). On the other hand, blocking-before-service is also referred to as *type 2 blocking* (Onvural and Perros 1986), *communication blocking*, *service blocking*, and *immediate blocking* (Gün and Makowski 1989). The difference between the two blocking mechanisms is whether the upstream machine is allowed to operate when the buffer is full. The former allows operation while the latter one does not. We assume blocking-before-service for the production line models considered in this thesis[1].

Failure Types. Two major types of failures have been considered in the litera-

[1]The effect of the difference between blocking models is no greater than the effect of changing all buffer sizes by 1 (Gershwin 1994).

ture: *operation dependent failures* (ODF) and *time dependent failures* (TDF) (Buzacott and Hanifin 1978). ODFs indicate that machine may only fail when it is operating. However, TDFs are not related to the processing of parts and thus can occur at any time, including when machine is idle. Good examples of ODFs and TDFs are given in Dallery and Gershwin (1992), where it says ODFs are mainly due to mechanical causes (like tool breakage or motor burnout) while TDFs are mainly due to failures of electronic systems, such as controllers. However, ODF is the most important kind of failure in a production line (Buzacott and Hanifin 1978). We consider ODFs in this thesis.

The Deterministic Processing Time, Discrete Material Model. This model is also known informally as the deterministic model. The key feature of this model is that processing times of all machines are equal, deterministic, and constant. Therefore, time is scaled so that operations take one time unit. We further assume that all the machines start their operations at the same instant. Transportation time is negligible compared to the operation time.

Machines are unreliable and are parameterized by probabilities of failure and repair. Each machine is allowed to have have only one failure mode or multiple failure modes, and therefore we have the so called *deterministic single failure mode model* (Buzacott 1967a and Gershwin 1994) and *deterministic multiple failure mode model* (Tolio and Matta 1998). For the single failure mode model, the parameters of Machine M_i are p_i, the probability of a failure during a time unit while the machine is operating; and r_i, the probability of a repair during a time unit while the machine is down. As a consequence, the times to failure and to repair are geometrically distributed. By convention, repairs and failures occur at the beginnings of time units and changes in the buffer levels take place at the ends of time units.

Several most influential early papers of Buzacott have been dedicated to study the behavior of this deterministic processing time and discrete material production line model. Because of the influence of Buzacott's work, this model is usually known as the Buzacott model (Dallery and Gershwin 1992).

The Exponential Processing Time, Discrete Material Model. This model

is also known as the exponential model. In this model, the behavior of Machine M_i is characterized by three exponentially distributed random variables: the service time (with mean $1/\mu_i$), the time to fail (with mean $1/p_i$ — abbreviated MTTF) and the time to repair (with mean $1/r_i$ — abbreviated MTTR). In other words, the service, failure and repair times for M_i are assumed to be exponential random variables with parameters μ_i, p_i and r_i. This model is more flexible than the previous one because the machines are not all required to operate at the same speed.

In terms of the number of failure modes a machine may have, there are the exponential single failure mode model (Choong and Gershwin 1987) and the exponential multiple failure mode model (Levantesi et al. 1999b).

The Continuous Processing Time, Continuous Material Model. This model is also known as the continuous model. In this model, the material that is processed is treated as though it is a continuous fluid. The assumptions on which this model is based are more general than those of the deterministic model in that the machines can operate at different speeds (Gershwin 1994). In addition, the rate of machine failure is affected by the buffer level: whether it is empty, full, or in between. Again, machine may have single failure mode or multiple failure modes. Therefore, we have the continuous single failure mode model (Burman 1995) and the continuous multiple failure mode model (Levantesi et al. 2003).

The constant μ_i is the speed at which Machine M_i processes material while it is operating and not constrained by the other machine or the buffer. The unreliability of a machine is captured by exponential random variables. However, it is important to indicate that in this model the failure probability of a machine (in a given small time interval) is affected by buffer levels because of the different machine speeds. (See Gershwin 1994 for details.) If Machine M_i is not affected, its failure rate is p_i. The repair rate of Machine M_i is denoted by r_i, and the corresponding repair probability is not affected by other machines or buffer levels.

In this thesis, we study three major research topics for the deterministic model and the continuous model. We outline the connection between the research topics and the production line models in Section 1.3.

1.2.2 Production Line Evaluation

We start the review of production line evaluation by considering two-machine lines, for which analytical solutions exist. For good summary of the analysis of different two-machine line models, see Buzacott and Shanthikumar (1993) and Gershwin (1994).

Buzacott (1968) discusses the evaluation of the efficiency of production systems, including long serial lines, without internal storage buffers. For other early work about two-machine lines without buffers, see Rao (1975), Lau (1986a), and (1986b). Here we focus on two-machine line with buffers. Buzacott (1967a) derives the analytic formula for the production rate for two-machine, one-buffer deterministic processing time lines. Early work on the effect of storage buffers on the production rate of production lines include Buzacott (1967b, 1971, 1972) and Gershwin and Berman (1981). Other than these works, there are some major papers that introduce the analytical solutions of different two-machine line models (i.e., the deterministic model, the exponential model, and the continuous model) with the ODF assumption. They are summarized in Table 1.1. (For a discussion about some analytical work of two-machine lines with TDF, see Li et al. 2006.)

Table 1.1: Summary of analytical solutions for different two-machine line models

Line model	Number of failure modes	Analytical solution
Deterministic	single	Schick and Gershwin (1978) Buzacott and Shanthikumar (1993) Gershwin (1994)
	multiple	Tolio and Gershwin (1996) Tolio et al. (2002)
Exponential	single	Gershwin and Berman (1981)
	multiple	Levantesi et al. (1999c)
Continuous	single	Wijngaard (1979) Gershwin and Schick (1980) Glassey and Hong (1986)
	multiple	Levantesi et al. (1999a)

Recently, analytical solutions of more general two-machine line models have been developed. For example, Gershwin and Fallah-Fini (2007) propose a method to an-

alyze general deterministic processing time, discrete material production lines with single buffer and identical processing rates. In van Vuuren and Adan (2009), they present analytical solutions to analyze two-machine lines where machines are modeled as reliable servers with generally distributed service times. Tan and Gershwin (2009) and (2011) study general continuous Markovian two-machine production line systems. Tolio (2011) analyzes continuous two-machine lines with multiple up and down states, where machines at different up states are allowed to have different processing speeds. In addition, Gebennini et al. (2009) and Gebennini et al. (2011) study two-machine line evaluation with a restart policy for the first machine for the deterministic model and the continuous model, respectively. These work mentioned above enlarge the application scope of two-machine line evaluation.

The invention of a *decomposition* method with unreliable machines and finite buffers (Gershwin 1984, 1987a) and its many extensions and modifications (Gershwin 1987b, Choong and Gershwin 1987, Dallery et al. 1988, 1989, Glassey and Hong 1993, Burman 1995, Gershwin and Burman 2000, Dallery and Le Bihan 1995, 1999, Le Bihan and Dallery 2000, Tolio and Matta 1998, Levantesi et al. 1999b, and Levantesi et al. 2003) enable the numerical evaluation of different models of production lines having more than two machines. We briefly review the roadmap of the decomposition approach as follows:

- Gershwin (1984), (1987a) first developed a decomposition method for the discrete time discrete material long line model (i.e., the deterministic model), where machines have the same processing time. The method is implemented by the corresponding Dallery-David-Xie (*DDX*) algorithm (Dallery et al. 1988).

  For a k-machine $k-1$-buffer line, the decomposition method is based on a representation of the $k-1$-buffer system by $k-1$ single-buffer systems, i.e., $k-1$ two-machine one-buffer building blocks (Gershwin 1994). For each building block that contains Buffer B_i of the original line, two pseudo-machines, denoted by $M^u(i)$ and $M^d(i)$, are constructed to represent the portion upstream of B_i and the portion downstream of B_i of the original line, respectively (see Figure

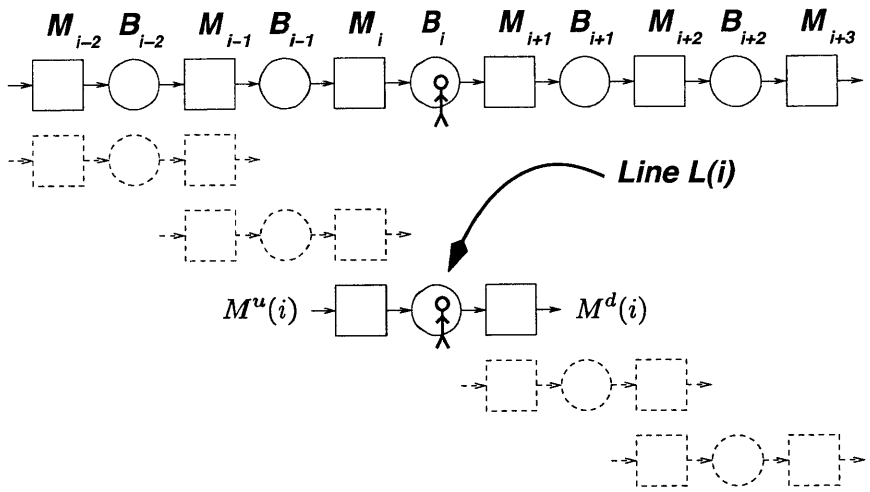


Figure 1-2: The decomposition method

1-2). For each pseudo-machine, there are two parameters that parameterize the failure and repair properties of the machine. As a result, there are $4(k-1)$ unknowns. On the other hand, $4(k-1)$ equations are constructed by considering the conservation of flow, the flow rate-idle time relationship, the resumption of flow, and the boundary conditions. The unique solutions of the $4(k-1)$ unknowns are solved by those $4(k-1)$ equations, after which the production rate of the line as well as the average inventory of each buffer are evaluated. The analytical solution of two-machine line evaluation developed by Schick and Gershwin (1978) and Gershwin and Schick (1983) is adopted to evaluate those $k-1$ building blocks and to solve the decomposition equations. Dallery et al. (1989) extend the decomposition method and the DDX method to a continuous production line model.

- The limit of the model above is that the times that parts spend being processed at machines are equal at all machines (i.e., all machines have the same processing speed). Therefore, Gershwin (1987b) extends the decomposition method to a discrete time discrete material *nonhomogeneous* line model in which machines are allowed to take different lengths of time performing operations on parts.

- Choong and Gershwin (1987) extend the original decomposition method to the

exponential processing time and discrete material production line model. In this model, machines are allowed to have different processing rates. In particular, the processing rate, the failure rate, and the repair rate of each machine are exponentially distributed. This method makes use of the work of Gershwin and Berman (1981) to evaluate the resulting $k-1$ two-machine building blocks. Gershwin (1989) adopts a version of the DDX algorithm to implement this decomposition approach for exponential lines.

- Glassey and Hong (1993) develop a decomposition method based on Gershwin (1987a) for continuous processing time and continuous material production line models. In particular, in their model, machines are allowed to have different deterministic processing rates. In addition, for each machine, the time to failure and the time to repair are exponentially distributed. In the decomposition, the analysis of the two-machine line by Glassey and Hong (1986), which modifies the method of Wijngaard (1979), is used. Burman (1995) claims some disadvantage of the work of Glassey and Hong (1993) and develops a different set of decomposition equations for the continuous line model. The continuous two-machine model of Gershwin and Schick (1980) is used to evaluate each of the building blocks in the decomposition. In addition, Burman (1995) invents an Accelerated DDX (*ADDX*) algorithm to solve those decomposition equations. The ADDX algorithm is demonstrated to be faster and to provide better reliability of convergence than the DDX method.

- Dallery and Le Bihan (1995), (1999) indicate that when the reliability parameters (mean time to failure and mean time to repair) of the different machines have different orders of magnitude, the original decomposition method of Gershwin (1987a) and the DDX algorithm (Dallery et al. 1988) provide less accurate evaluation results. Therefore, they propose an improvement of the original decomposition method that provides accurate results even in the above mentioned situation. The reliability of the decomposition method is further improved by Le Bihan and Dallery (2000).

- Finally, Tolio and Matta (1998), Levantesi et al. (1999b), and Levantesi et al. (2003) develop three decomposition methods, based on Gershwin (1987a), for all three production line models (i.e., the deterministic model, the exponential model, and the continuous model). The most important feature of these methods is that they allow machines to have more than one failure mode because of the development of the analytical solutions of two-machine line with multiple failure mode machines of all those three models (Tolio and Gershwin 1996, Levantesi et al. 1999c, and Levantesi et al. 1999a). In other words, any given machine can fail in different ways. This makes the line models more realistic. In particular, Tolio and Matta (1998) deal with the deterministic model and make use of the work of Tolio and Gershwin (1996) to evaluate two-machine building blocks. Levantesi et al. (1999b) deal with the exponential model and adopt the analysis of Levantesi et al. (1999c) to evaluate two-machine building blocks. Levantesi et al. (2003) deal with the continuous model and apply the work of Levantesi et al. (1999a) for two-machine building block evaluation. We refer to the three decomposition methods above as *Tolio decomposition*, while the original work of Gershwin (1987a) as *Gershwin decomposition*. Then the most important difference between Gershwin decomposition and Tolio decomposition is that: in Gershwin decomposition, a single set of failure parameters (i.e., failure and repair probabilities in the discrete model, and failure and repair rates in the exponential and the continuous models) is determined for each pseudo-machine to approximate the portion of the original line represented by that pseudo-machine; while in Tolio decomposition, multiple failure modes are determined for each pseudo-machine to approximate the portion of the original line it represents. These multiple failure modes of a certain pseudo-machine correspond to the real failures of all machines (of the original line) represented by that pseudo-machine. Because of the advantage of being able to construct pseudo-machines with multiple-failure modes, Tolio decomposition is expected to be more accurate, yet slower, than Gershwin decomposition.

In addition to the works mentioned above, Syrowicz (1999) and Colledani et al.

(2005) extend the decomposition method to study deterministic, multiple-part-type, multiple-failure-mode production lines. Bierbooms et al. (2011) apply the decomposition method to analyze the performance of continuous production lines with finite buffers and machines with generally distributed uptimes and downtimes. Senanayake et al. (2011) develop an analytical method for the performance evaluation of hybrid production lines where both manual and automated operations co-exist based on decomposition. The decomposition approach is also extended to the evaluation for assembly/disassembly systems (Di Mascolo et al. 1991, Gershwin 1991 and Gershwin and Burman 2000) as well as closed-loop systems[2] (Frein et al. 1996, Werner 2001, and Gershwin and Werner 2007) and multiple-loop systems (Zhang 2006).

Other than the decomposition methods, De Koster (1987) also proposes an aggregation approach. However, in that approach, the correlations among the buffers are not taken into account and the aggregation is only proceeded forward (Li et al. 2009). The early verision aggregation method of De Koster (1987) is further improved by De Koster (1988). Also see Terracol and David (1987) for such a method. In addition, Lim et al. (1990) develop an aggregation method. As summarized in Li et al. (2009), the method consists of a backward and a forward aggregation. In the backward aggregation, the last subline $M_{k-1} - B_{k-1} - M_k$ are aggregated into a single machine represented by M_{k-1}^b. Then the subline $M_{k-2} - B_{k-2} - M_{k-1}^b$ is aggregated into Machine M_{k-2}^b, and so on until all machines and buffers are aggregated into M_1^b. In the forward aggregation, the subline $M_1 - B_1 - M_2^b$ is aggregated into M_2^f. Then M_2^f is aggregated with M_3 and B_2 to form M_3^f, and so on until all machines and the intervening buffers are aggregated into M_k^f. The process is repeated until the throughputs of M_1^b and M_k^f converge and are used as an estimate of the throughput of the line.

Methods have been found for the exact numerical analysis of some small lines with more than two machines. For instance, Gershwin and Schick (1983) derive an analytical solution for a three-machine line with unreliable machines and small buffers. There are also numerical methods for exact analysis of lines that are slightly longer

[2]We will further study single loop systems in Chapters 5 and 6.

with small buffers (Tan 2002). However, they are severely limited. In this thesis, we use decomposition for the evaluation and optimization of much larger production line systems.

1.2.3 Production Line Optimization

Next, we review some work that is designated to the optimization of production lines. Park (1993) developes a two-phase heuristic algorithm to solve the total buffer space minimization problem. But his method can not always find the optimal solutions and does not always converge.

Seong et al. (1994) adopt the concept of concept of pseudo gradient and gradient projection to solve the production rate maximization problem and the profit maximization problem for a specified total buffer space for continuous production lines. Seong et al. (1995) use a gradient method to solve the production rate maximization problem for exponential production lines.

Gershwin and Goldis (1995) employ a gradient method to solve the total buffer space minimization problem. Their algorithm is based on the observation that if the production rate is expanded to first order the problem may be formulated as an integer linear program.

Schor (1995), and Gershwin and Schor (2000) present an efficient buffer allocation algorithm that applied a primal-dual approach to minimize the total buffer space under a production rate constraint. In their work, the primal problem is to minimize total buffer space subject to a production rate constraint, while the dual problem is to maximize the production rate of the line subject to total buffer space constraint. They also study the profit maximization of a line through a nonlinear programming method that is fast and accurate, but they do not consider the production rate constraint in the profit maximization problem. (As we will indicate later in Section 4.3.5, their primal problem is a special case of the our profit maximization problem for production lines.)

More recently, Huang et al. (2002) consider a flow-shop-type production system and apply a dynamic programming approach to maximize its production rate or min-

imize its work-in-process under a certain buffer allocation strategy. Diamantidis and Papadopoulos (2004) also present a dynamic programming algorithm for optimizing buffer allocation based on the aggregation method of Lim et al. (1990). Although their dynamic programming methodology brings new approaches to production line design, they do not attempt to maximize the profits of lines.

Chan and Ng (2002) compare four buffer allocation strategies and presente a modified one for production rate maximization. Shi and Men (2003) introduce a hybrid algorithm based on hybrid nested partitions and a Tabu search method (Glover and Laguna 1997) for production line optimization. However, they focus on maximizing the production rate of the line under a total buffer space constraint, rather than the profit of the line. Smith and Cruz (2005) solve the buffer allocation problem for general finite buffer queueing networks in which they minimize buffer space cost under the production rate constraint, but they do not consider the average inventory cost.

One paper that considers both buffer space cost and average inventory cost is Dolgui et al. (2002). Their buffer allocation problem aims at determining buffer capacities considering the production rate of the line, the buffer acquisition and installation cost, and the inventory cost. For that problem, they propose a genetic algorithm where tentative solutions are evaluated with an approximate method based on the Markov-model aggregation approach. However, they do not have the production rate constraint in their problem.

Colledani et al. (2003) minimize the total buffer space subject to a production rate constraint for deterministic single failure mode lines. Their algorithm is baed on an iterative scheme that, starting from the configuration of the line with minimal capacity of each buffer, proceeds by increasing the capacity of buffers until the target production rate is reached. Colledani and Tolio (2005) solve the same problem for deterministic multiple failure mode lines. They use a first order Taylor expansion to linearize the decomposition equations and therefore the production rate of the line, so that to convert the nonlinear production rate constraint into a linear constraint. As a result, their problem becomes a mixed integer linear problem. Tolio et al. (2009) extend their algorithm to continuous production lines with multiple failure modes.

Some practical considerations in optimization of flow production systems are reported in Tempelmeier (2003). In addition, some metaheuristic methods are adopted to deal with the scheduling and balancing problems for production lines or assembly lines (Jin et al. 2006 and Bautista and Pereira 2007).

The optimization problem becomes much harder if the production rate constraint is considered in production line design because the production rate is a nonlinear function of buffer sizes. As it will be indicated in Section 1.3, our optimization problem includes the production rate constraint and aims at maximizing the profit for production lines, where we consider both buffer space cost and inventory holding cost. The average inventory of the line, and consequently the line's cost, are also nonlinear functions of buffer sizes. Hence, we have nonlinear elements in both our objective function and constraints.

1.3 Research Goal and Contributions

The goal of this thesis is to develop efficient buffer design algorithms for production line profit maximization subject to a production rate constraint, and therefore to provide valuable insight about production line design to manufacturing system practitioners.

In this thesis, we define the profit of a k-machine $k-1$-buffer line as

$$\text{Profit} = AP(N_1, \cdots, N_{k-1}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i - Z, \tag{1.1}$$

where $A > 0$ (\$/part) is the revenue coefficient associated with the production rate $P(N_1, \cdots, N_{k-1})$ (or $P(\mathbf{N})$), b_i and c_i (\$/part/time unit) are cost coefficients associated with the buffer space and average inventory for the ith buffer, respectively, and Z stands for all costs other than those due to buffer sizes, average inventory, and raw material. Since Z is independent of $\mathbf{N}$, we simplify the formulation above to

$$J(N_1, \cdots, N_{k-1}) = AP(N_1, \cdots, N_{k-1}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i, \tag{1.2}$$

where we refer to $J(N_1, \cdots, N_{k-1})$ as the profit of the line. In addition, the production rate can be required to satisfy as $P(\mathbf{N}) \geq \hat{P}$, where $\hat{P}$ is the target production rate.

The goal is achieved by investigating three major topics. They are

1. production line profit maximization subject to a production rate constraint,

2. single closed-loop system (i.e., a special production line structure where the last machine and the first machine are connected with a buffer) profit maximization subject to a production rate constraint,

3. and finally, production line profit maximization subject to both a production rate constraint and a maximum waiting time constraint of parts in a given buffer.

In particular, we consider Topic 1 to be the primary focus from the perspective of algorithm development. This is because, as we will show, the algorithm developed for Topic 1 can be extended to Topics 2 and 3. However, Topics 2 and 3 have their unique attributes that are not covered by Topic 1. In order to optimize closed-loop systems, we have to first improve the evaluation algorithm of such systems and therefore two necessary modifications about loop evaluation are provided (see Chapter 5). On the other hand, in order to study the maximum part waiting time constraint in a given buffer, the analytical formulation of the part waiting time distribution is developed for two-machine lines and it is extended for longer lines with decomposition in Chapter 7. For the specific problem formulation and algorithm derivation associated with each topic, refer to individual chapters. The research motivations of Topics 2 and 3 are also provided in those corresponding chapters.

The primary production line model considered in this thesis is the deterministic single failure mode production line model of Gershwin (1994). However, the profit maximization algorithm is also applied to optimize the deterministic multiple failure mode line model of Tolio and Matta (1998) and the continuous multiple failure mode line model of Levantesi et al. (2003) in Chapter 4. In addition, the proposed algorithm also applies to the continuous line model of Burman (1995).

1.4 Thesis Outline

The remaining of the thesis is organized as follows:

- A discussion about the qualitative behavior of the production rate of production lines, where the monotonicity and concavity of $P(\mathbf{N})$ are proved for the deterministic two-machine line model of Gershwin (1994), and that for longer lines are discussed with numerical experiments and a literature review. (Chapter 2.)

- A discussion about the qualitative behavior of average buffer levels, where three-machine two-buffer lines are analyzed since they present the simplest form of inventory level behavior of one buffer caused by varying the size of another buffer. The findings are applied to longer lines. (Chapter 3.)

- An efficient buffer allocation algorithm for production line profit maximization subject to a production rate constraint for production lines. (Chapter 4.)

- Two modifications that enable more accurate evaluation results for closed-loop systems. The evaluation of closed-loop systems is also extended to the case of single open-loop systems. (Chapter 5.)

- The profit maximization algorithm for closed-loop systems. This is an extension of the algorithm developed for production lines. (Chapter 6.)

- The analytical formulation of the part waiting time distribution for the two-machine line model of Gershwin (1994) (which can be easily extended to the multiple failure mode model of Tolio and Gershwin 1996 and Tolio et al. 2002), and an optimization algorithm for production line profit maximization subject to both the production rate constraint and the maximum part waiting time constraint of parts in a given buffer. (Chapter 7.)

- A segmentation method for long line optimization that makes use of the algorithm of Chapter 4 and significantly reduces the computer time of long line optimization while guaranteeing the accuracy. (Chapter 8.)

- The development of an additive property in production line optimization, which together with the segmentation method, provide useful line design insights. (Chapter 9.)
- Summary of the thesis contributions and outline of future research directions. (Chapter 10.)

Chapter 2

Qualitative Behavior of the Production Rate $P(\mathbf{N})$

We indicate in Chapter 1 that the primary goal of this research is to develop an efficient buffer allocation algorithm that maximizes the profit of production lines under a production rate constraint. The difficulty of achieving this goal comes from the nonlinearity of both the production rate as well as the profit of the line, since the profit is a function of the production rate, the buffer sizes, and the average inventory levels of all buffers.

In this chapter and the next chapter (Chapter 3), we study the qualitative behavior of the production rate $P(\mathbf{N})$ (where $\mathbf{N}$ is the vector of buffer sizes) and average buffer levels $\bar{n}_i(\mathbf{N})$, respectively. Understanding them would help us better develop the desired algorithm for production line profit maximization. In particular, the qualitative properties of the production rate $P(\mathbf{N})$ includes its *continuity*, *monotonicity*, and *concavity*. Gershwin and Schor (2000) describe these three properties as follows and we follow their description:

- *continuity*: a small change in a buffer's size would lead to a small change in the system's performance (i.e., the production rate as well as average inventory levels of all buffers).

- *monotonicity*: an increase in a buffer's size (while all the other buffer sizes are

increased or held constant) increases the production rate.

- *concavity*: the increase in production rate due to a unit increase in buffer size decreases as the buffer size increases.

In what follows, we first discuss the continuity of $P(\mathbf{N})$ in Section 2.1. After that, the monotonicity and concavity of $P(N)$ for two-machine lines are proved in Section 2.2. Finally, we provide some literature review and numerical evidence to show the monotonicity and concavity of $P(\mathbf{N})$ for long lines in Section 2.3.

We want to point out clearly that the production rate of long lines is evaluated by an approximate decomposition method (Gershwin 1987a) and therefore we do not have rigorous proofs of the monotonicity and concavity of $P(\mathbf{N})$ for long lines. However, both the literature review and the numerical evidence discussed in Section 2.3 indicate that they are good assumptions of $P(\mathbf{N})$ for long lines. On the other hand, since we have exact analytical solutions for two-machine lines, we prove the monotonicity and concavity of $P(N)$ for two-machine lines.

2.1 Continuity of $P(\mathbf{N})$

In the discrete time discrete material production line model, buffer spaces are discrete and buffer sizes are integers. However, we can take advantage of the analytical form of the two-machine line evaluation, which enables us to treat N as a continuous variable. This is because the formulas for the production rate and the average inventory level of the two-machine line do not require N to be integers[1]. This enables us to evaluate a two-machine line whose buffer size is not an integer. We provide an example of this.

Consider a two-machine line whose parameters are $r_1 = .1$, $p_1 = .01$, $r_2 = .2$, and $p_2 = .01$. The production rate of the line as a function of the buffer size N is illustrated in Figure 2-1. In particular, the curve in Figure 2-1 is generated by varying the buffer size N as a continuous variable with a step size of .01 for drawing the curve; while the discrete dots in Figure 2-1 are generated by restricting N to take integer values.

[1]We include the continuous variable version of the analytical solution of the two-machine line in Appendix A.

The smooth curve of $P(N)$ indicates that when the restriction $N \in Z^+$ is relaxed to $N \in R^+$, the production rate of two-machine lines $P(N)$ becomes a continuous function. There is no unexpected bump or discontinuity when N is considered as a continuous variable.

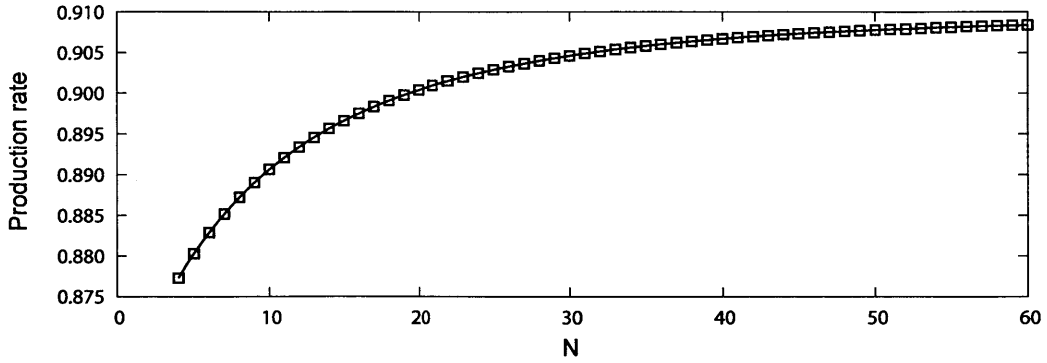


Figure 2-1: $P(N)$ vs. non-integer N, two-machine one-buffer line

On the other hand, the average inventory level of the two-machine line mentioned above is also illustrated in Figure 2-2. Similarly, the curve in Figure 2-2 is generated by varying the buffer size N as a continuous variable; while the discrete dots in Figure 2-2 are generated by restricting N to take integer values. The smooth curve of $\bar{n}(N)$ indicates that when the restriction $N \in Z^+$ is relaxed to $N \in R^+$, the average inventory level $\bar{n}(N)$ also becomes a continuous function. There is no unexpected bump or discontinuity.

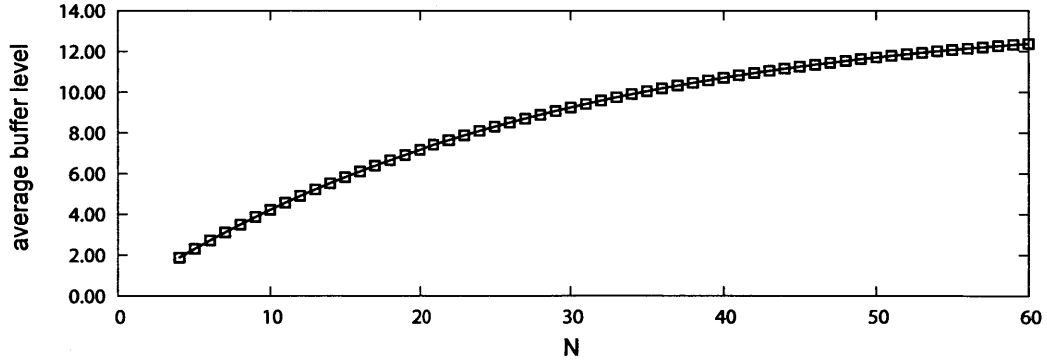


Figure 2-2: $\bar{n}(N)$ vs. non-integer N, two-machine one-buffer line

From these results, it appears that for two-machine lines, the buffer size N can be

considered as a continuous variable, and therefore $P(N)$ and $\bar{n}(N)$ can be considered as continuously differentiable functions. Moreover, since we use the decomposition method (Gershwin 1987a), which makes use of the two-machine line evaluation, to evaluate long lines, all the buffer sizes $N_i, i = 1, \cdots, k-1$ can be treated as continuous variables. As a consequence, for discrete time discrete material long lines, the production rate $P(\mathbf{N})$ and the average inventory levels $\bar{n}_i(\mathbf{N}), i = 1, \cdots, k-1$ can all be considered as continuous differentiable functions of buffer size $\mathbf{N}$. We illustrate this with a numerical example.

Consider a three-machine two-buffer discrete material line whose parameters are $r_1 = .1$, $p_1 = .01$, $r_2 = .2$, $p_2 = .03$, $r_3 = .4$, and $p_3 = .01$. The production rate of the line as N_1 and N_2 vary is illustrated in Figure 2-3. The surface in Figure 2-1 is generated by varying the buffer size N as a continuous variable with a step size of .5 and evaluating the line using the approximate decomposition method. The smooth surface of $P(N_1, N_2)$ indicates that when the restriction $\mathbf{N} \in Z^+$ is relaxed to $\mathbf{N} \in R^+$, the production rate of three-machine two-buffer lines becomes a continuous function[2]. There is no unexpected bump or discontinuity. The continuity of $\bar{n}_1(\mathbf{N})$ and $\bar{n}_1(\mathbf{N})$ is shown in Figure 2-4.

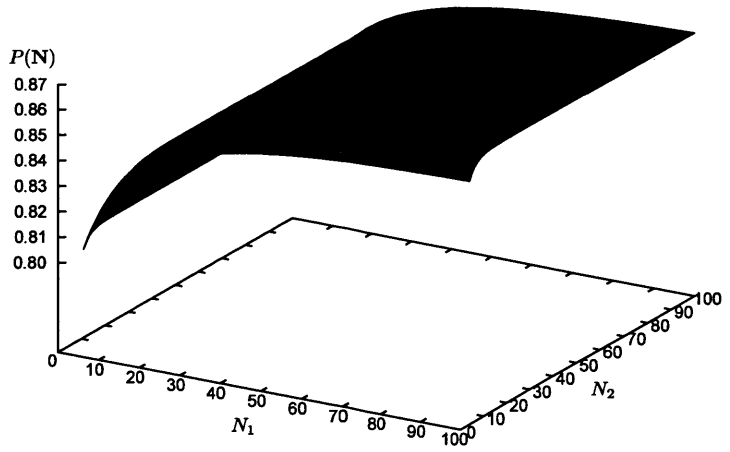


Figure 2-3: $P(\mathbf{N})$ vs. non-integer $\mathbf{N}$, three-machine two-buffer line

[2]Note that the production rate also appears to be a concave function of N_1 and N_2 in Figure 2-1. We will revisit this point when we discuss the concavity of $P(\mathbf{N})$ for long lines in Section 2.3.

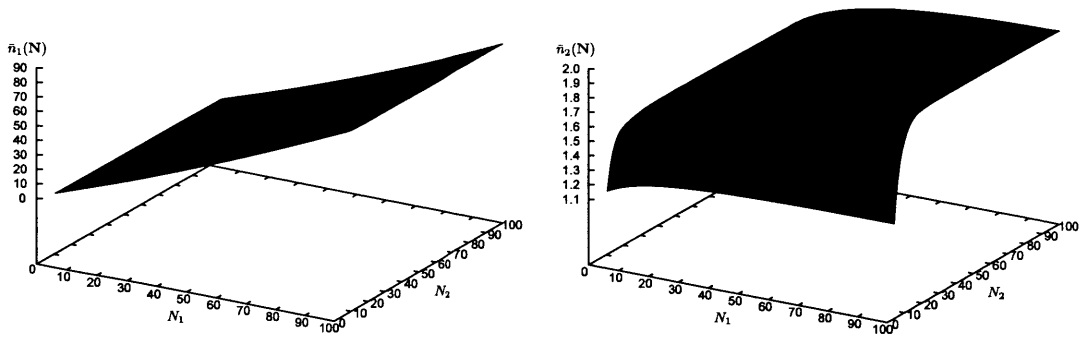


Figure 2-4: $\bar{n}_1(\mathbf{N})$ and $\bar{n}_2(\mathbf{N})$ vs. non-integer $\mathbf{N}$, three-machine two-buffer line

2.2 The Monotonicity and Concavity of $P(N)$ in Two-Machine Lines

Gershwin (1994) develops sets of analytical solutions for different two-machine one-buffer line models. As mentioned in Chapter 1, by following the naming methods there, they are the deterministic model, the exponential model, and the continuous model. In particular, we study the deterministic model (i.e., deterministic processing time and discrete material model) here. The analytical solution of this model is summarized in Appendix A. In what follows, we show the monotonicity and concavity of the production rate of two-machine lines.

For a deterministic processing time and discrete material two-machine line, each Machine M_i is parameterized by the repair probability r_i and the failure probability p_i. The size of the buffer between Machines M_1 and M_2 is N. The production rate of the line is $P(N)$. In addition, according to Gershwin (1994), e_i is defined to be the *isolated production rate* of M_i. It is what the production rate of M_i would be if it were never impeded by the other machine or the buffer. It is given by

$$e_i = \frac{r_i}{r_i + p_i} \tag{2.1}$$

and it represents the fraction of time that M_i is operational. The actual production rate of M_i is less because of blocking or starvation. Since we use P to represent the production rate in the thesis, in what follows, we use P_i instead of e_i to represent the

isolated production rate of Machine M_i.

There are two possible cases for a two-machine line: the two machines have the same isolated production rate and the two machines have different isolated production rates. We analyze the two cases separately.

2.2.1 The Isolated Production Rates of the Two Machines are Different

According to Gershwin (1994), the production rate of a two-machine line can be calculated by

$$\begin{aligned} P(N) &= P_1(1-p_b) \\ &= P_1\left(1 - CX^{N-1}\frac{r_1+r_2-r_1r_2-p_1r_2}{p_1r_2}\right) \\ &= \frac{r_1}{r_1+p_1}\left(1 - CX^{N-1}\frac{r_1+r_2-r_1r_2-p_1r_2}{p_1r_2}\right) \end{aligned} \tag{2.2}$$

where P_1 is the isolated production rate of the upstream machine M_1 assuming no blocking or starvation, p_b is the probability of blocking of the upstream machine, C is a *normalizing constant*, and

$$Y_1 = \frac{r_1+r_2-r_1r_2-r_1p_2}{p_1+p_2-p_1p_2-p_1r_2}, \tag{2.3}$$

$$Y_2 = \frac{r_1+r_2-r_1r_2-p_1r_2}{p_1+p_2-p_1p_2-r_1p_2}, \tag{2.4}$$

$$X = \frac{Y_2}{Y_1}. \tag{2.5}$$

On the other hand, the production rate can also be computed by

$$
\begin{aligned}
P(N) &= P_2(1-p_s) \\
&= P_2\left(1 - CX\frac{r_1+r_2-r_1r_2-r_1p_2}{r_1p_2}\right) \qquad (2.6) \\
&= \frac{r_2}{r_2+p_2}\left(1 - CX\frac{r_1+r_2-r_1r_2-r_1p_2}{r_1p_2}\right)
\end{aligned}
$$

where P_2 is the isolated production rate of the downstream machine M_2 assuming no blocking or starvation, p_s is the probability of starvation of the downstream machine.

Rewriting equation (2.2) gives

$$
P(N)\frac{r_1+p_1}{r_1} + CX^{N-1}\frac{r_1+r_2-r_1r_2-p_1r_2}{p_1r_2} = 1. \qquad (2.7)
$$

Similarly, rewriting equation (2.6) gives

$$
P(N)\frac{r_2+p_2}{r_2} + CX\frac{r_1+r_2-r_1r_2-r_1p_2}{r_1p_2} = 1. \qquad (2.8)
$$

Note that when $P_1 = P_2$, then $X = 1$ according to Equations (2.3), (2.4), and (2.5). Therefore, Equations (2.7) and (2.8) are identical and we cannot find the expressions for the production rate $P(N)$ and the normalizing constant C by solving them[3]. However, when $P_1 \neq P_2$, $X \neq 1$ and Equations (2.7) and (2.8) are different. Therefore, we are able to solve them together to find $P(N)$ and C because there are two equations and two unknowns. Thus, solving them together yields an analytical expression for the production rate $P(N)$ of the two-machine line,

$$
P(N) = \frac{X^{N-1}\dfrac{r_1+r_2-r_1r_2-p_1r_2}{p_1r_2} - X\dfrac{r_1+r_2-r_1r_2-r_1p_2}{r_1p_2}}{\dfrac{r_2+p_2}{r_2}X^{N-1}\dfrac{r_1+r_2-r_1r_2-p_1r_2}{p_1r_2} - \dfrac{r_1+p_1}{r_1}X\dfrac{r_1+r_2-r_1r_2-r_1p_2}{r_1p_2}}, \qquad (2.9)
$$

[3]We study this case with a different approach in Section 2.2.2.

and an analytical expression for the normalizing constant C,

$$C = \frac{\frac{r_2+p_2}{r_2} - \frac{r_1+p_1}{r_1}}{\frac{r_2+p_2}{r_2}X^{N-1}\frac{r_1+r_2-r_1r_2-p_1r_2}{p_1r_2} - \frac{r_1+p_1}{r_1}X\frac{r_1+r_2-r_1r_2-r_1p_2}{r_1p_2}}. \quad (2.10)$$

We would like to emphasize that the expressions for $P(N)$ and C derived above are **not** meaningful when $P_1 = P_2$ since in that case Equations (2.7) and (2.8) are identical and we cannot solve two unknowns from one equation.

To show the monotonicity and concavity of $P(N)$, we first derive the first order and the second order derivatives of $P(N)$ with respect to N. If we can show that $\mathrm{d}P/\mathrm{d}N > 0$ and $\mathrm{d}^2P/\mathrm{d}N^2 < 0$, then the monotonicity and concavity of $P(N)$ follow. Next, we will show the desired properties. First, we need to derive $\mathrm{d}P/\mathrm{d}N > 0$. It can be computed from (2.9). We realize that both the numerator and the denominator in (2.9) have the decision variable N. Then $\mathrm{d}P/\mathrm{d}N$ can be computed according to the quotient rule of calculus (Larson et al. 2005) and the derivative of non natural base exponential function (Berresford and Rockett 2008),

$$\frac{\mathrm{d}P}{\mathrm{d}N} = \frac{AB\ln X\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^N}{\left(\frac{A}{P_2}X^{N-1}-\frac{B}{P_1}X\right)^2} \quad (2.11)$$

where

$$A = \frac{r_1+r_2-r_1r_2-p_1r_2}{p_1r_2},$$

and

$$B = \frac{r_1+r_2-r_1r_2-r_1p_2}{r_1p_2}.$$

It is easy to see that the denominator of (2.11) is positive. In addition, with the assumption that $0 < p_i < 1$ and $0 < r_i < 1$, A, B, and X are positive. Moreover, if $P_1 > P_2$, then $X > 1$; while if $P_1 < P_2$, then $0 < X < 1$. Therefore, it can be seen that $(1/P_2 - 1/P_1)$ and $\ln X$ always have the same sign. Therefore, the numerator of (2.11) is also positive. Therefore, $\mathrm{d}P/\mathrm{d}N > 0$. This shows that $P(N)$ is a monotonically

increasing function of N.

Next, we consider $\mathrm{d}^2P/\mathrm{d}N^2$, which can be computed from $\mathrm{d}P/\mathrm{d}N$ according to the quotient rule again. Therefore, we have

$$\frac{\mathrm{d}^2P}{\mathrm{d}N^2} = \frac{-\frac{A^3B(\ln X)^2}{P_2^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^{3N-2} + \frac{AB^3(\ln X)^2}{P_1^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^{N+2}}{\left(\frac{A}{P_2}X^{N-1}-\frac{B}{P_1}X\right)^4}. \quad (2.12)$$

It is easy to see that the denominator of (2.12) is positive. To study the sign of the numerator of (2.12), we consider the two cases ($P_1 > P_2$ and $P_1 < P_2$) separately.

First, assume that $P_1 > P_2$. In this case, $X > 1$. In addition, it easy to see that $P_1 > P_2$ implies that $p_1r_2 < p_2r_1$ and therefore $A > B(> 0)$. In addition, according to the convention of the deterministic processing time and discrete material model of Gershwin (1994), we have $N \geq 4$. As a result $3N-2 > N+2$ and $X^{3N-2} > X^{N+2} > 0$. (Note that $3N - 2 > N + 2$ does not require $N \geq 4$. In fact, it requires $N > 2$. The analytical solution in Gershwin 1994 does not apply to the case where $N = 2$ or 3, although it appears that our version of the analytical solution (see Appendix A) works when $N = 2$ and 3. We choose to follow the practice of Gershwin 1994 and let $N \geq 4$.) Since $A > B$ and $P_1 > P_2$, we know that

$$\frac{A^3B(\ln X)^2}{P_2^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right) > \frac{AB^3(\ln X)^2}{P_1^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right) > 0,$$

and therefore the numerator of (2.12) satisfies

$$
\begin{aligned}
& -\frac{A^3B(\ln X)^2}{P_2^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^{3N-2}+\frac{AB^3(\ln X)^2}{P_1^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^{N+2} \\
< & -\frac{AB^3(\ln X)^2}{P_1^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^{3N-2}+\frac{AB^3(\ln X)^2}{P_1^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^{N+2} \\
< & -\frac{AB^3(\ln X)^2}{P_1^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^{N+2}+\frac{AB^3(\ln X)^2}{P_1^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^{N+2} \\
= & \ 0.
\end{aligned}
$$

Consequently, from the derivation above, we see that the numerator of (2.12) is negative when $P_1 > P_2$. Recall that the denominator of (2.12) is positive. Therefore, $\mathrm{d}^2P/\mathrm{d}N^2 < 0$.

On the other hand, assume that $P_1 < P_2$. In this case, $0 < X < 1$. It easy to see that $P_1 < P_2$ implies that $p_1r_2 > p_2r_1$ and therefore $B > A(> 0)$. Again, since $N \geq 4$, $3N-2 > N+2$ and $0 < X^{3N-2} < X^{N+2}$. Since $A < B$ and $P_1 < P_2$, we know that

$$
0 > \frac{A^3B(\ln X)^2}{P_2^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right) > \frac{AB^3(\ln X)^2}{P_1^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right),
$$

and therefore the numerator of (2.12) satisfies

$$
\begin{aligned}
& -\frac{A^3B(\ln X)^2}{P_2^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^{3N-2}+\frac{AB^3(\ln X)^2}{P_1^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^{N+2} \\
< & -\frac{A^3B(\ln X)^2}{P_2^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^{3N-2}+\frac{A^3B(\ln X)^2}{P_2^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^{N+2} \\
< & -\frac{A^3B(\ln X)^2}{P_2^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^{3N-2}+\frac{A^3B(\ln X)^2}{P_2^2}\left(\frac{1}{P_2}-\frac{1}{P_1}\right)X^{3N-2} \\
= & \ 0.
\end{aligned}
$$

Consequently, from the derivation above, we see that the numerator of (2.12) is also negative when $P_1 < P_2$. Recall that the denominator of (2.12) is positive. Therefore, $\mathrm{d}^2P/\mathrm{d}N^2 < 0$.

According to the analysis above, we see that no matter if $P_1 > P_2$ or $P_1 < P_2$, $\mathrm{d}^2P/\mathrm{d}N^2$ is always negative, which indicates the concavity of $P(N)$ when $P_1 \neq P_2$.

2.2.2 The Isolated Production Rates of the Two Machines are the Same

Let us consider the case where $P_1 = P_2$ now. In this case, we cannot solve for the production rate $P(N)$ and the normalizing constant C by (2.7) and (2.8). Therefore, in this case, we find C by realizing that all steady state probabilities must sum to 1,

$$CA_eX + CX + CXY_2 + CB_eX + CX^{N-1} + CX^{N-1}Y_1 + CD_eX^{N-1} + CE_eX^{N-1} + C(N-3)(1+Y_1)(1+Y_2) = 1 \tag{2.13}$$

where

$$A_e = \frac{r_1 + r_2 - r_1r_2 - r_1p_2}{r_1p_2},$$

$$B_e = \frac{r_1 + r_2 - r_1r_2 - r_1p_2}{p_1 + p_2 - p_1p_2 - r_1p_2}\frac{1}{p_2},$$

$$D_e = \frac{r_1 + r_2 - r_1r_2 - p_1r_2}{p_1 + p_2 - p_1p_2 - p_1r_2}\frac{1}{p_1},$$

$$E_e = \frac{r_1 + r_2 - r_1r_2 - p_1r_2}{p_1r_2}.$$

We modify (2.13) and get

$$C\left[(A_e+1+Y_2+B_e)X+(1+Y_1+D_e+E_e)X^{N-1}+(N-3)(1+Y_1)(1+Y_2)\right]=1. \tag{2.14}$$

For simplicity, let

$$M_e = 1+Y_2+A_e+B_e>0, \tag{2.15}$$

$$R_e = 1+Y_1+D_e+E_e>0, \tag{2.16}$$

$$Q_e = (1+Y_1)(1+Y_2)>0, \tag{2.17}$$

then Equation (2.14) becomes

$$C\left[M_eX+R_eX^{N-1}+Q_e(N-3)\right]=1, \tag{2.18}$$

or

$$\begin{aligned} C &= \frac{1}{M_eX+R_eX^{N-1}+Q_e(N-3)} \\ &= \frac{1}{M_e+R_e+Q_e(N-3)} \end{aligned} \tag{2.19}$$

because $X=1$. Therefore, the production rate $P(N)$ can be computed by

$$\begin{aligned} P(N) &= P_2(1-p_s) \\ &= P_2-\frac{P_2A_e}{M_e+R_e+Q_e(N-3)}. \end{aligned} \tag{2.20}$$

The first order and the second order derivatives of $P(N)$ with respect to N are

$$\frac{\mathrm{d}P}{\mathrm{d}N}=\frac{P_2A_eQ_e}{\Big(M_e+R_e+Q_e(N-3)\Big)^2}>0, \tag{2.21}$$

and

$$\frac{\mathrm{d}^2P}{\mathrm{d}N^2}=\frac{-2P_2A_eQ_e^2}{\Big(M_e+R_e+Q_e(N-3)\Big)^3}<0. \tag{2.22}$$

According to Equations (2.21) and (2.22), we conclude that $P(N)$ is also monotonically increasing and concave in this case. Sections 2.2.1 and 2.2.2 indicate the monotonicity and the concavity of $P(N)$ for two-machine lines.

2.3 The Monotonicity and Concavity of $P(\mathbf{N})$ in Longer Lines

2.3.1 Literature Review

A common intuition in the line design field is the concavity (as well as the monotonicity) of $P(\mathbf{N})$, though there is no analytical result that conclusively shows that the lines (with more than two machines) we study in this thesis exhibit the concavity property. However, some research in similar systems indicates that this is a reasonable assumption.

Okamura and Yamashina (1977) demonstrate the monotonicity and concavity of the throughput of two-machine one-buffer transfer lines with geometric machines and finite buffers. They classify a two-machine line into three types according to the magnitudes of the failure and repair probabilities of machines. However, no matter to which type a given line belongs, its production rate is a monotonically increasing and concave function of the size of the buffer between the two machines.

Shanthikumar and Yao (1989a) point out the monotonicity and concavity properties in cyclic queueing networks with finite buffers and exponential servers. They use the evolution equations of a sample path approach (Muth 1979, Dallery and Gershwin 1992) to prove the results. Dallery and Gershwin (1992) point out that although the monotonicity property was established in the context of closed systems in Shanthikumar and Yao (1989a), it is readily applicable to the case of transfer lines.

Shanthikumar and Yao (1989b) establish the monotonicity and concavity of the throughput in a multicell system. Each cell processes a given part family. According to the flow pattern of jobs, the cells are categorized into two types. A Type 1 cell is modeled as a Jackson network; a Type 2 cell is modeled as an ordered-entry system

with heterogeneous servers. Both models have finite waiting room due to the buffer capacity allocated to the cells. They show that the production rate of each cell of either type is an increasing and concave function of its buffer allocation.

Anantharam and Tsoucas (1990) prove the stochastic concavity of throughput in a series of $\cdot/M/1/B$ queues. The notation means that the ith queue has on server with independent and identically distributed (i.i.d.) exponential service times and a waiting room of size B_i.

Meester and Shanthikumar (1990) study the throughput in tandem queueing systems with m stages and finite intermediate buffer storage spaces. Each stage has a single server and the service times are independent and exponentially. They show that for this system the number of customers departing from each of the k stages during time interval $[0, t]$ for any $t \geq 0$ is strongly stochastically increasing and concave in the buffer storage capacities. Consequently, the through put of this tandem queueing system is an increasing and concave function of the buffer storage capacities. Hillier and So (1995) make use of the concavity of throughput result of Meester and Shanthikumar (1990) for optimal design of tandem queuing systems with finite buffers. Dallery et al. (1994) generalize the work of Meester and Shanthikumar (1990) and present the concavity properties in Fork/Join queueing networks with blocking (FJQN/B). The FJQN/B is first introduced by Ammar and Gershwin (1989).

Glasserman and Yao (1996) show the monotonicity and concavity of the throughput as a function of of buffer parameters in serial lines with general blocking and synchronized service. In this research, a production line is modeled as a generalized semi-Markov process. Schor (1995) explains that even though Glasserman and Yao (1996) assume reliable servers, this result is applicable to our system where machines are unreliable. This is because any system with unreliable machines may be transformed into a system of reliable machines by changing the distribution of the service time. As an example, Altiok and Stidham (1983) show that a machine with exponential service, failure and repair times can be represented as a reliable machine with a coxian server (Cox 1955). Therefore, the differences between flow lines with reliable machines and flow lines with unreliable machines has more to do with how the system

is described than how the system performs (Schor 1995).

Rajan and Agrawal (1998) demonstrate the concavity of the throughput of a large class of queueing systems with i.i.d. *new-better-than-used* (Marshall and Shaked 1986) service times. Xie (2002) show the concavity of the throughput of 2-stage continuous transfer lines subject to time-dependent failures. Kwon (2006) studies the optimal buffer allocation problem of a flexible manufacturing system of Sung and Kwon (1994) and indicates that in both the first-level and the second-level queue-alone subsystems, the throughputs are monotonically increasing and concave functions of their buffer sizes, respectively.

In addition, some work based on the concavity of the production rate for the same or similar systems has been published. Seong et al. (1995) develop two heuristic algorithms for buffer allocation in a production line with unreliable machines with the concavity assumption of the throughput. In particular, they study how to maximize the production rate of the line given fixed total buffer space. Park (1993) assumes the concavity of the production rate over both a buffer and a vector of buffers in his study of buffer size optimization. Gershwin and Schor (2000) establish a primal-dual algorithm for buffer space allocation in production lines basing on the assumption of concave $P(\mathbf{N})$. Levantesi et al. (2001) presents another algorithm for buffer allocation in production lines with the same assumption. Jeong and Kim (2000) applies that property to assembly systems. So (1997) also mentions the concavity of the production rate in his study on optimal buffer allocation strategy for unpaced production lines.

Moreover, Colledani and Tolio (2005) develop a buffer allocation algorithm that minimizes the total buffer space subject to a production rate constraint for production lines with finite buffers and machines that are allowed to have multiple geometric failure modes. In their approach, they make use of the monotonicity and concavity of the production rate to assure the convergence of the proposed gradient methods. Colledani et al. (2003) and Tolio et al. (2009) also assume the monotonicity and concavity of the production rate in their algorithms for minimization of total buffer space.

Other than the literature mentioned above, Schor (1995) provides a detailed survey on the monotonicity property of the production rate. We make use of the monotonicity and concavity of $P(\mathbf{N})$ assumption when we derive the production line profit maximization algorithm in Chapter 4.

2.3.2 Numerical Evidence

The production rate of a given three-machine two-buffer line as a function of buffer sizes (N_1, N_2) is plotted in Figure 2-3 when we discuss the continuity of $P(\mathbf{N})$ in Section 2.1. The figure also illustrates that $P(\mathbf{N})$ appears to be a monotonically increasing and concave function of of $\mathbf{N}$. Numerical experiments reveal that the shape of the $P(\mathbf{N})$ surface for three machine lines remain the same qualitatively regardless of the machine parameters (see Figure 2-5).

Table 2.1: Machine parameters of the five experiments

case	Line 1	Line 2	Line 3	Line 4	Line 5
r_1	.1	.8	.7	.1	.12
p_1	.1	.096	.01	.01	.009
r_2	.09	.1	.12	.1	.15
p_2	.01	.01	.008	.01	.009
r_3	.11	.1	.12	.8	.07
p_3	.01	.01	.008	.096	.01

In Figure 2-5, we plot the production rate $P(\mathbf{N})$ for five three-machine two-buffer lines with different parameters. The iso-production rate curves are also projected on the $N_1 - N_2$ plane. Machine parameters of these five lines are listed in Table 2.1. As it will be discussed in Chapter 3, the five lines under consideration can be classified into five different types of three-machine two-buffer lines according to the classification method (with respect to the relative speeds of different parts of the line divided by one of the two buffers) presented in Section 3.2.1. (Any given three-machine line will belong to one of the five types.) Both the shape of the $P(\mathbf{N})$ surfaces and the concave iso-throughput curves in Figure 2-5 suggest that the production rates of the five lines are all concave functions no matter which type a line is. These numerical experiments

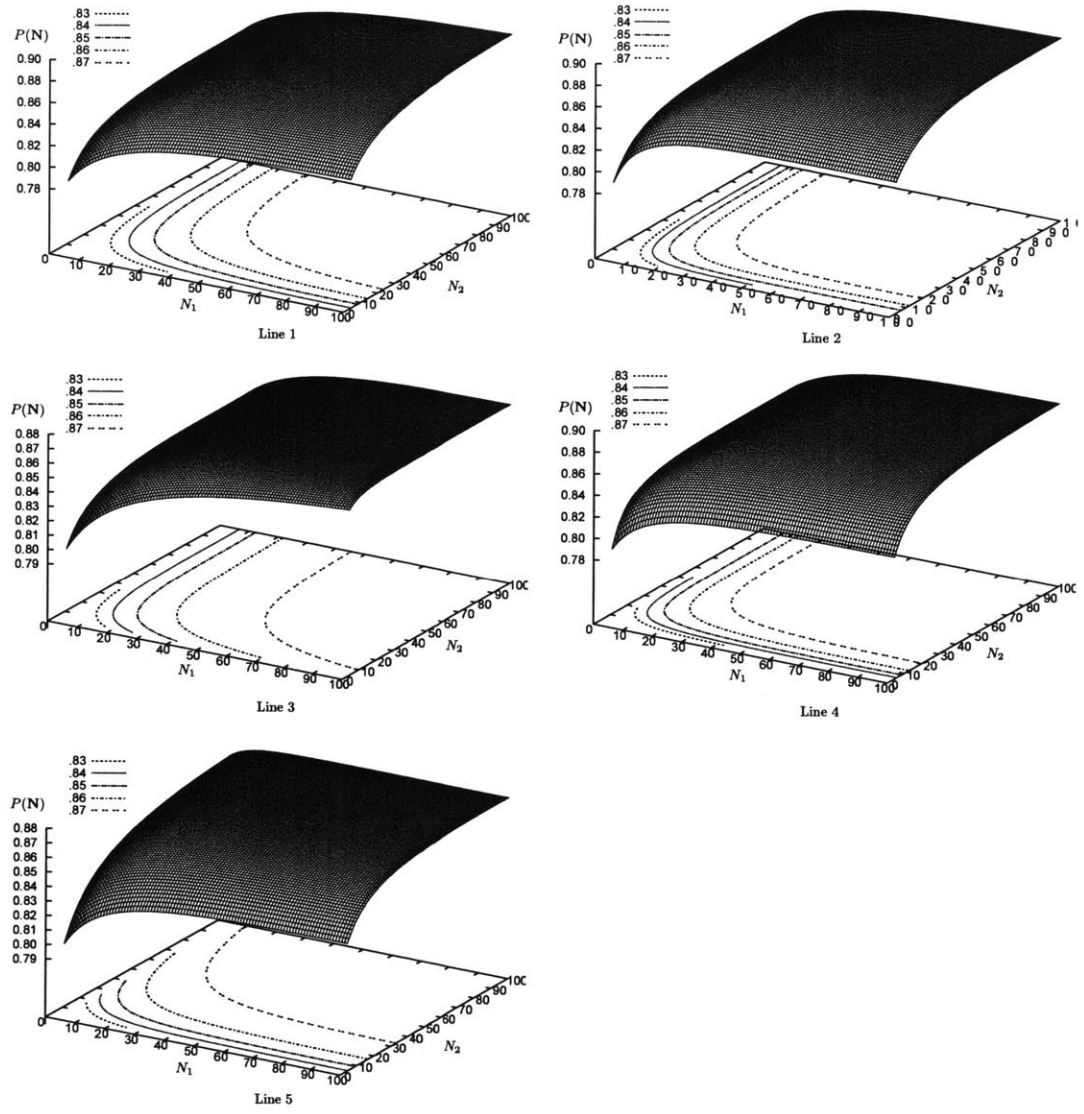


Figure 2-5: $P(\mathbf{N})$ vs. $\mathbf{N}$, five experiments

support the conjecture that $P(\mathbf{N})$ is a monotonically increasing and concave function of of $\mathbf{N}$.

2.4 Summary

In this chapter, some qualitative properties of the production rate $P(\mathbf{N})$, including the continuity, the monotonicity, and the concavity, are studied. The continuity enables

us to treat $P(\mathbf{N})$ as a continuous function of $\mathbf{N}$ even when the deterministic line model is used. This facilitates us to apply a gradient method as part of the proposed profit maximization algorithm in Chapter 4. In addition, the monotonicity and concavity assumption of $P(\mathbf{N})$ are also used in deriving the optimization algorithm.

Chapter 3

Qualitative Behavior of Average Buffer Levels

3.1 Motivation

We indicate in Chapter 1 that the profit of a production line is a function of buffer sizes, the average inventory levels of all buffers, and the production rate of the line. In addition, some qualitative properties, including continuity, monotonicity, and concavity, of the production rate as a function of buffer sizes are discussed in Chapter 2. Therefore, in this chapter, we study the qualitative behavior of average inventory levels as functions of buffer sizes based on observations made from numerical experiments, and extend the scope of the study to the profit of production lines. In particular, we study three-machine two-buffer lines because

1. They represent the simplest example of interaction between buffers. Therefore, it enables us to understand how the average inventory of a buffer changes as we vary the size of the buffer as well as the other buffer.

2. Understanding three-machine two-buffer lines gives insight into longer lines. This is because for any two buffers B_i and B_j in a k-machine $k-1$-buffer line, they divide the line into three segments: $M_1 - B_1 - \cdots - B_{i-1} - M_i$, $M_{i+1} - B_{i+1} - \cdots - B_{j-1} - M_j$, and $M_{j+1} - B_{j+1} - \cdots - B_{k-1} - M_k$ (see Figure

3-1). Therefore, the original k-machine $k-1$-buffer line can be viewed as a three-machine two-buffer line. We show this extension in Section 3.5.

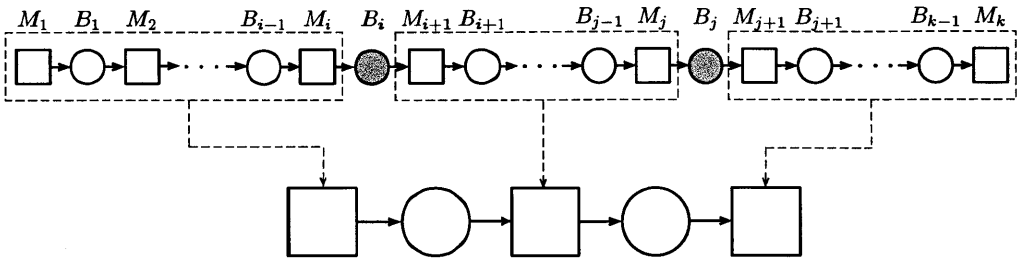


Figure 3-1: A three-machine two-buffer line representative of a k-machine $k-1$-buffer line

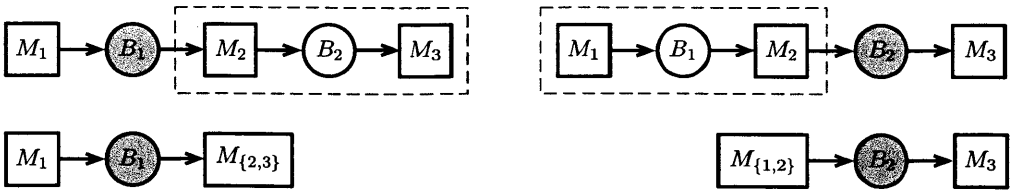
Visualizing these quantities of a three-machine two-buffer line as a function of N_1 and/or N_2 can help us gain some important insights about the behavior of long lines. Moreover, we make further use of these insights to derive the optimal buffer allocation algorithm for production line profit maximization in Chapter 4.

3.2 Three-Machine Two-Buffer Line Classification

3.2.1 Motivation of Classification

A three-machine two-buffer line has buffers B_1 and B_2. The average inventory of a given buffer not only varies as the size of that buffer changes, but also varies with the size of the other. In other words, $\bar{n}_1$ is a function of both N_1 and N_2, and the same is true for $\bar{n}_2$. If we want to study $\bar{n}_1$ (as a function of N_1 and N_2) in a three-machine two-buffer line, we can view it as a two-machine one-buffer line, with Machine M_1 being the upstream machine, Buffer B_1 as the buffer, and a downstream pseudo-machine $M_{\{2,3\}}$ that represents M_2, B_2, and M_3 as a whole in the original line (see Figure 3-2(a)). (Note that we are not using decomposition here to approximate $M_2 - B_2 - M_3$ and therefore $M_{\{2,3\}}$ is exactly $M_2 - B_2 - M_3$.)

In Figure 3-2(a), in order to study the qualitative behavior of $\bar{n}_1$, we consider the original three-machine two-buffer line as a two-machine one-buffer line with respect



(a) Two-machine one-buffer line w.r.t. B_1

(b) Two-machine one-buffer line w.r.t. B_2

Figure 3-2: Two-machine line representations of the original three-machine line

to Buffer B_1, or $M_1 - B_1 - M_{\{2,3\}}$. Given certain machine parameters of M_1, M_2 and M_3, there are three cases:

1. No matter what value N_2 takes, M_1 is always faster than $M_{\{2,3\}}$. This is to say that the isolated production rate of M_1 is always higher than that of $M_{\{2,3\}}$. For convenience, let P_1 denote the production rate of M_1, while $P_{\{2,3\}}(N_2)$ denotes the production rate of $M_{\{2,3\}}$ as a function of N_2. Then we have $P_1 > P_{\{2,3\}}(N_2), \forall N_2 > 0$, or equivalently, $P_1 \geq P_{\{2,3\}}(\infty)$.

2. No matter what value N_2 takes, M_1 is always slower than $M_{\{2,3\}}$. This is to say that the production rate of M_1 is always smaller than that of $M_{\{2,3\}}$. With the same notation, we have $P_1 < P_{\{2,3\}}(N_2), \forall N_2 > 0$, or equivalently, $P_1 \leq P_{\{2,3\}}(0)$.

3. For some values of N_2, M_1 is faster than $M_{\{2,3\}}$, while for other values of N_2, M_1 is slower than $M_{\{2,3\}}$. In other words, the production rate of M_1 is smaller for some values of N_2 but higher for other values of N_2 than that of $M_{\{2,3\}}$. Therefore, $P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}(\infty)$.

In these three cases, $\bar{n}_1$ exhibits different behaviors. Studying them separately would help us better understand the qualitative behavior of the average inventory of Buffer B_1. Similarly, there are three cases for $\bar{n}_2$. Therefore, there are totally $3 \times 3 =$ nine possible types of behavior for a three-machine two-buffer line.

3.2.2 Nine Types

To sum up, the nine types are listed in Table 3.1. However, as we show shortly, not all of these nine types are feasible, because the case for $\bar{n}_1$ and the case for $\bar{n}_2$ are not independent. We will study the nine types individually and summarize their feasibilities accordingly.

Table 3.1: Feasibility of nine types, to be determined

	$P_1 \geq P_{\{2,3\}}(\infty)$	$P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}(\infty)$	$P_1 \leq P_{\{2,3\}}(0)$
$P_3 \geq P_{\{1,2\}}(\infty)$			
$P_{\{1,2\}}(0) < P_3 < P_{\{1,2\}}(\infty)$			
$P_3 \leq P_{\{1,2\}}(0)$			

3.3 Feasibility Analysis of Nine Types

3.3.1 Feasibility Analysis

In this section, we analyze the feasibility of each of those nine types. For the convenience in the analysis below, it is helpful to point out that

$$P_1 = \frac{r_1}{r_1 + p_1},$$

$$P_2 = \frac{r_2}{r_2 + p_2},$$

$$P_3 = \frac{r_3}{r_3 + p_3},$$

$$P_{\{1,2\}}(0) = \frac{1}{1 + \frac{p_1}{r_1} + \frac{p_2}{r_2}} = \frac{1}{\frac{1}{P_1} + \frac{1}{P_2} - 1},$$

$$P_{\{1,2\}}(\infty) = \min\left(\frac{r_1}{r_1 + p_1}, \frac{r_2}{r_2 + p_2}\right) = \min(P_1, P_2),$$

$$P_{\{2,3\}}(0) = \frac{1}{1+\frac{p_2}{r_2}+\frac{p_3}{r_3}} = \frac{1}{\frac{1}{P_2}+\frac{1}{P_3}-1},$$

$$P_{\{2,3\}}(\infty) = \min\left(\frac{r_2}{r_2+p_2}, \frac{r_3}{r_3+p_3}\right) = \min(P_2, P_3).$$

Type 1: $P_3 \geq P_{\{1,2\}}(\infty)$ **and** $P_1 \geq P_{\{2,3\}}(\infty)$

It is easy to see that $P_3 \geq P_{\{1,2\}}(\infty)$ requires

$$\frac{r_3}{r_3+p_3} \geq \min\left(\frac{r_1}{r_1+p_1}, \frac{r_2}{r_2+p_2}\right),$$

while $P_1 \geq P_{\{2,3\}}(\infty)$ requires

$$\frac{r_1}{r_1+p_1} \geq \min\left(\frac{r_2}{r_2+p_2}, \frac{r_3}{r_3+p_3}\right).$$

Combining these two conditions, we know that Type 1 is feasible if and only if,

$$\frac{r_3}{r_3+p_3} \geq \min\left(\frac{r_1}{r_1+p_1}, \frac{r_2}{r_2+p_2}\right) \quad \text{and} \quad \frac{r_1}{r_1+p_1} \geq \min\left(\frac{r_2}{r_2+p_2}, \frac{r_3}{r_3+p_3}\right).$$

An example of Type 1 is:

$$\begin{array}{lll} r_1 = .1 & p_1 = .01 & P_1 = .909 \\ r_2 = .09 & p_2 = .01 & P_2 = .9 \\ r_3 = .11 & p_3 = .01 & P_3 = .917 \end{array}$$

Type 2: $P_3 \geq P_{\{1,2\}}(\infty)$ **and** $P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}(\infty)$

$P_3 \geq P_{\{1,2\}}(\infty)$ means that

$$\frac{r_3}{r_3+p_3} \geq \min\left(\frac{r_1}{r_1+p_1}, \frac{r_2}{r_2+p_2}\right),$$

while $P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}(\infty)$ indicates that

$$\frac{1}{1+\frac{p_2}{r_2}+\frac{p_3}{r_3}} < \frac{r_1}{r_1+p_1} < \min\left(\frac{r_2}{r_2+p_2}, \frac{r_3}{r_3+p_3}\right).$$

Combining these two conditions reveals that Machine M_1 has to be the slowest one of the line and if $P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}$ is satisfied, then $P_3 \geq P_{\{1,2\}}(\infty)$ is satisfied automatically. Therefore, Type 2 is feasible. An example of Type 2 is:

$$\begin{aligned} r_1 &= .8 \quad p_1 = .096 \quad P_1 = .893 \\ r_2 &= .1 \quad p_2 = .01 \quad P_2 = .909 \\ r_3 &= .1 \quad p_3 = .01 \quad P_3 = .909 \end{aligned}$$

$$\frac{1}{1+\frac{p_2}{r_2}+\frac{p_3}{r_3}} = \frac{1}{1+\frac{.01}{.11}+\frac{.01}{.11}} = .833$$

Type 3: $P_3 \geq P_{\{1,2\}}(\infty)$ **and** $P_1 \leq P_{\{2,3\}}(0)$

$P_3 \geq P_{\{1,2\}}(\infty)$ requires

$$\frac{r_3}{r_3+p_3} \geq \min\left(\frac{r_1}{r_1+p_1}, \frac{r_2}{r_2+p_2}\right),$$

while $P_1 \leq P_{\{2,3\}}(0)$ requires

$$\frac{r_1}{r_1+p_1} \leq \frac{1}{1+\frac{p_2}{r_2}+\frac{p_3}{r_3}}.$$

Therefore, Type 3 is feasible if and only if

$$\frac{r_1}{r_1+p_1} \leq \frac{1}{1+\frac{p_2}{r_2}+\frac{p_3}{r_3}},$$

which implies $P_3 \geq P_{\{1,2\}}(\infty)$. Thus, Type 3 is feasible. An example of Type 3 is:

$$\begin{aligned} r_1 &= .07 \quad p_1 = .01 \quad P_1 = .875 \\ r_2 &= .12 \quad p_2 = .008 \quad P_2 = .938 \\ r_3 &= .12 \quad p_3 = .008 \quad P_3 = .938 \end{aligned}$$

$$\frac{1}{1+\frac{p_2}{r_2}+\frac{p_3}{r_3}} = \frac{1}{1+\frac{.008}{.12}+\frac{.008}{.12}} = .882$$

Type 4: $P_{\{1,2\}}(0) < P_3 < P_{\{1,2\}}(\infty)$ **and** $P_1 \geq P_{\{2,3\}}(\infty)$

$P_{\{1,2\}}(0) < P_3 < P_{\{1,2\}}(\infty)$ requires

$$\frac{1}{1+\frac{p_1}{r_1}+\frac{p_2}{r_2}} < \frac{r_3}{r_3+p_3} < \min\left(\frac{r_1}{r_1+p_1}, \frac{r_2}{r_2+p_2}\right),$$

while $P_1 \geq P_{\{2,3\}}(\infty)$ requires

$$\frac{r_1}{r_1+p_1} \geq \min\left(\frac{r_2}{r_2+p_2}, \frac{r_3}{r_3+p_3}\right).$$

According to these two conditions, we know that Machine M_3 has to be the slowest one and if $P_{\{1,2\}}(0) < P_3 < P_{\{1,2\}}(\infty)$ is satisfied, then $P_1 \geq P_{\{2,3\}}(\infty)$ is satisfied automatically. Thus, Type 4 is feasible. In addition, it can be seen that Type 4 lines are reverses of Type 2 lines. An example of Type 4 is:

$$\begin{aligned} r_1 &= .1 \quad p_1 = .01 \quad P_1 = .909 \\ r_2 &= .1 \quad p_2 = .01 \quad P_2 = .909 \\ r_3 &= .8 \quad p_3 = .096 \quad P_3 = .893 \end{aligned}$$

$$\frac{1}{1+\frac{p_1}{r_1}+\frac{p_2}{r_2}} = \frac{1}{1+\frac{.01}{.1}+\frac{.01}{.1}} = .833$$

Type 5: $P_3 \leq P_{\{1,2\}}(0)$ **and** $P_1 \geq P_{\{2,3\}}(\infty)$

$P_3 \leq P_{\{1,2\}}(0)$ requires

$$\frac{r_3}{r_3+p_3} \leq \frac{1}{1+\frac{p_1}{r_1}+\frac{p_2}{r_2}},$$

while $P_1 \geq P_{\{2,3\}}(\infty)$ requires

$$\frac{r_1}{r_1+p_1} \geq \min\left(\frac{r_2}{r_2+p_2}, \frac{r_3}{r_3+p_3}\right).$$

Therefore, Type 3 is feasible if and only if

$$\frac{r_3}{r_3+p_3} \leq \frac{1}{1+\frac{p_1}{r_1}+\frac{p_2}{r_2}},$$

which implies $P_1 \geq P_{\{2,3\}}(\infty)$. Thus, Type 5 is feasible. In addition, Type 5 lines are reverses of Type 3 lines. An example of Type 5 is:

$$\begin{aligned} r_1 &= .12 \quad p_1 = .009 \quad P_1 = .930 \\ r_2 &= .15 \quad p_2 = .009 \quad P_2 = .943 \\ r_3 &= .07 \quad p_3 = .01 \quad P_3 = .875 \end{aligned}$$

$$\frac{1}{1+\frac{p_1}{r_1}+\frac{p_2}{r_2}} = \frac{1}{1+\frac{.009}{.12}+\frac{.009}{.15}} = .881$$

Type 6: $P_{\{1,2\}}(0) < P_3 < P_{\{1,2\}}(\infty)$ **and** $P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}(\infty)$

$P_{\{1,2\}}(0) < P_3 < P_{\{1,2\}}(\infty)$ requires

$$\frac{1}{1+\frac{p_1}{r_1}+\frac{p_2}{r_2}} < \frac{r_3}{r_3+p_3} < \min\left(\frac{r_1}{r_1+p_1}, \frac{r_2}{r_2+p_2}\right),$$

while $P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}(\infty)$ requires

$$\frac{1}{1+\frac{p_2}{r_2}+\frac{p_3}{r_3}} < \frac{r_1}{r_1+p_1} < \min\left(\frac{r_2}{r_2+p_2}, \frac{r_3}{r_3+p_3}\right).$$

Condition 1 implies that M_3 should be the slowest machine. However, condition 2 indicates that M_1 should be the slowest one. Since they contradict to each other, Type 6 is infeasible.

Type 7: $P_{\{1,2\}}(0) < P_3 < P_{\{1,2\}}(\infty)$ **and** $P_1 \leq P_{\{2,3\}}(0)$

$P_{\{1,2\}}(0) < P_3 < P_{\{1,2\}}(\infty)$ requires

$$\frac{1}{1+\frac{p_1}{r_1}+\frac{p_2}{r_2}} < \frac{r_3}{r_3+p_3} < \min\left(\frac{r_1}{r_1+p_1}, \frac{r_2}{r_2+p_2}\right),$$

while $P_1 \leq P_{\{2,3\}}(0)$ requires

$$\frac{r_1}{r_1+p_1} \leq \frac{1}{1+\frac{p_2}{r_2}+\frac{p_3}{r_3}}.$$

Condition 1 implies that Machine M_3 should be the slowest one, while condition 2 implies that Machine M_1 should be the slowest one. Therefore, Type 7 is infeasible.

Type 8: $P_3 \leq P_{\{1,2\}}(0)$ **and** $P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}(\infty)$

$P_3 \leq P_{\{1,2\}}(0)$ requires

$$\frac{r_3}{r_1+p_3} \leq \frac{1}{1+\frac{p_1}{r_1}+\frac{p_2}{r_2}},$$

while $P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}(\infty)$ requires

$$\frac{1}{1+\frac{p_2}{r_2}+\frac{p_3}{r_3}} < \frac{r_1}{r_1+p_1} < \min\left(\frac{r_2}{r_2+p_2}, \frac{r_3}{r_3+p_3}\right).$$

Condition 1 implies that M_3 should be the slowest one, while condition 2 implies that M_1 should be the slowest one. Therefore, Type 8 is infeasible.

Type 9: $P_3 \leq P_{\{1,2\}}(0)$ **and** $P_1 \leq P_{\{2,3\}}(0)$

$P_3 \leq P_{\{1,2\}}(0)$ requires

$$\frac{r_3}{r_1 + p_3} \leq \frac{1}{1 + \frac{p_1}{r_1} + \frac{p_2}{r_2}},$$

while $P_1 \leq P_{\{2,3\}}(0)$ requires

$$\frac{r_1}{r_1 + p_1} \leq \frac{1}{1 + \frac{p_2}{r_2} + \frac{p_3}{r_3}}.$$

Condition 1 implies that Machine M_3 should be the slowest one, while condition 2 indicates that Machine M_1 should be the slowest one. Therefore, Type 9 is infeasible.

3.3.2 Feasibility Summary

According to the analysis above, we see that Types 1, 2, 3, 4, and 5 are feasible, while other four types are infeasible. The feasibilities of all types are summarized in Table 3.2. Moreover, the five feasible types are summarized in Table 3.3.

Table 3.2: Feasibility of nine types

	$P_1 > P_{\{2,3\}}(\infty)$	$P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}(\infty)$	$P_1 < P_{\{2,3\}}(0)$
$P_3 > P_{\{1,2\}}(\infty)$	**feasible**	**feasible**	**feasible**
$P_{\{1,2\}}(0) < P_3 < P_{\{1,2\}}(\infty)$	**feasible**	infeasible	infeasible
$P_3 < P_{\{1,2\}}(0)$	**feasible**	infeasible	infeasible

3.4 Qualitative Behavior of Five Feasible Types

In this section, we further study those five feasible types. In particular, for each type, we will study the behavior of the production rate ($P(N_1, N_2)$), average inventory of Buffer B_1 ($\bar{n}_1$), average inventory of Buffer B_2 ($\bar{n}_2$), and the profit of the three-machine two-buffer line as functions of N_1 and N_2 respectively. In other words, we will fix N_1 or N_2 and vary the other one. In addition, when N_1 is used as the decision

Table 3.3: Five feasible types

	properties	parameter conditions
Type 1	$P_3 \geq P_{\{1,2\}}(\infty)$ and $P_1 \geq P_{\{2,3\}}(\infty)$	$\frac{r_3}{r_3+p_3} \geq \min\left(\frac{r_1}{r_1+p_1}, \frac{r_2}{r_2+p_2}\right)$ and $\frac{r_1}{r_1+p_1} \geq \min\left(\frac{r_2}{r_2+p_2}, \frac{r_3}{r_3+p_3}\right)$
Type 2	$P_3 \geq P_{\{1,2\}}(\infty)$ and $P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}(\infty)$	$\frac{1}{1+\frac{p_2}{r_2}+\frac{p_3}{r_3}} < \frac{r_1}{r_1+p_1} < \min\left(\frac{r_2}{r_2+p_2}, \frac{r_3}{r_3+p_3}\right)$
Type 3	$P_3 \geq P_{\{1,2\}}(\infty)$ and $P_1 \leq P_{\{2,3\}}(0)$	$\frac{r_1}{r_1+p_1} \leq \frac{1}{1+\frac{p_2}{r_2}+\frac{p_3}{r_3}}$
Type 4	$P_{\{1,2\}}(0) < P_3 < P_{\{1,2\}}(\infty)$ and $P_1 \geq P_{\{2,3\}}(\infty)$	$\frac{1}{1+\frac{p_1}{r_1}+\frac{p_2}{r_2}} < \frac{r_3}{r_3+p_3} < \min\left(\frac{r_1}{r_1+p_1}, \frac{r_2}{r_2+p_2}\right)$
Type 5	$P_3 \leq P_{\{1,2\}}(0)$ and $P_1 \geq P_{\{2,3\}}(\infty)$	$\frac{r_3}{r_3+p_3} \leq \frac{1}{1+\frac{p_1}{r_1}+\frac{p_2}{r_2}}$

variable, we will consider three fixed values of N_2. Similarly, when N_2 is used as the decision variable, we will consider three fixed values of N_1. The profit $J(N_1, N_2)$ ($ per time unit) of a three-machine two-buffer line is computed as

$$J(N_1, N_2) = 1000P(N_1, N_2) - N_1 - N_2 - \bar{n}_1 - \bar{n}_2$$

In the analysis below, we will make frequent use of two-machine one-buffer line representations of the original three-machine two-buffer line. This facilitates our analysis of the qualitative behaviors of the four quantities under consideration. In particular, for B_1 in the original three-machine two-buffer line, we can view Machine M_1 as its upstream machine, but consider a downstream pseudo-machine $M_{\{2,3\}}$ that represents M_2, B_2, and M_3. Similarly, for B_2, we can view Machine M_3 as its downstream machine, but consider an upstream pseudo-machine $M_{\{1,2\}}$ that represents M_1, B_1, and M_2. To summarize, the two two-machine one-buffer lines that we will study frequently are $M_1 - B_1 - M_{\{2,3\}}$ and $M_{\{1,2\}} - B_2 - M_3$. As introduced in Section 3.2.1, we use P_1 and P_3 to represent the production rates of Machines M_1 and M_3, respectively. The isolated production rate of $M_{\{2,3\}}$ as a function of N_2 is denoted by $P_{\{2,3\}}(N_2)$. The isolated production rate of $M_{\{1,2\}}$ as a function of N_1 is denoted

by $P_{\{1,2\}}(N_1)$. Note that, we use the notation P to denote the production rate of the entire three-machine two-buffer line, but E to denote the isolated production rate of a portion of the line.

3.4.1 Type 1

Recall that Type 1 means that $P_3 \geq P_{\{1,2\}}(\infty)$ and $P_1 \geq P_{\{2,3\}}(\infty)$. Consider the example shown in Table 3.4. We first vary N_1. The three values of N_2 we consider are 30, 100, and 500. The four quantities being considered are shown in Figure 3-3. We explain them as follows.

Table 3.4: An example of Type 1

machine	M_1	M_2	M_3
r_i	.1	.09	.11
p_i	.01	.01	.01
P_i	.909	.9	.917

- Figure 3-3(a) shows the production rate $P(N_1)$, which appears to be a concave function of N_1. This is consistent with our assumption and argument about the concavity of the production rate in Chapter 2. Since in this type $P_1 \geq P_{\{2,3\}}(\infty)$, the production rate of the entire line is upper bounded by the isolated production rate of $M_2 - B_2 - M_3$, or $P_{\{2,3\}}(N_2)$, when N_1 is large enough. Therefore, as N_1 increases, the production rate of the line approaches to $P_{\{2,3\}}(N_2)$. The production rate asymptotes for these three cases (in terms of the value of N_2) are $P_{\{2,3\}}(30)$, $P_{\{2,3\}}(100)$, and $P_{\{2,3\}}(500)$ respectively, and $P_{\{2,3\}}(30) < P_{\{2,3\}}(100) < P_{\{2,3\}}(500)$.

- Figure 3-3(b) shows $\bar{n}_1(N_1)$. Because $P_1 \geq P_{\{2,3\}}(\infty)\,(> P_{\{2,3\}}(N_2))$, in the two-machine one-buffer line $M_1 - B_1 - M_{\{2,3\}}$, M_1 is always faster than $M_{\{2,3\}}$. Therefore, as N_1 increases, the average inventory $\bar{n}_1$ increases without a limit. $\bar{n}_1$ appears to be a convex function of N_1 in Type 1.

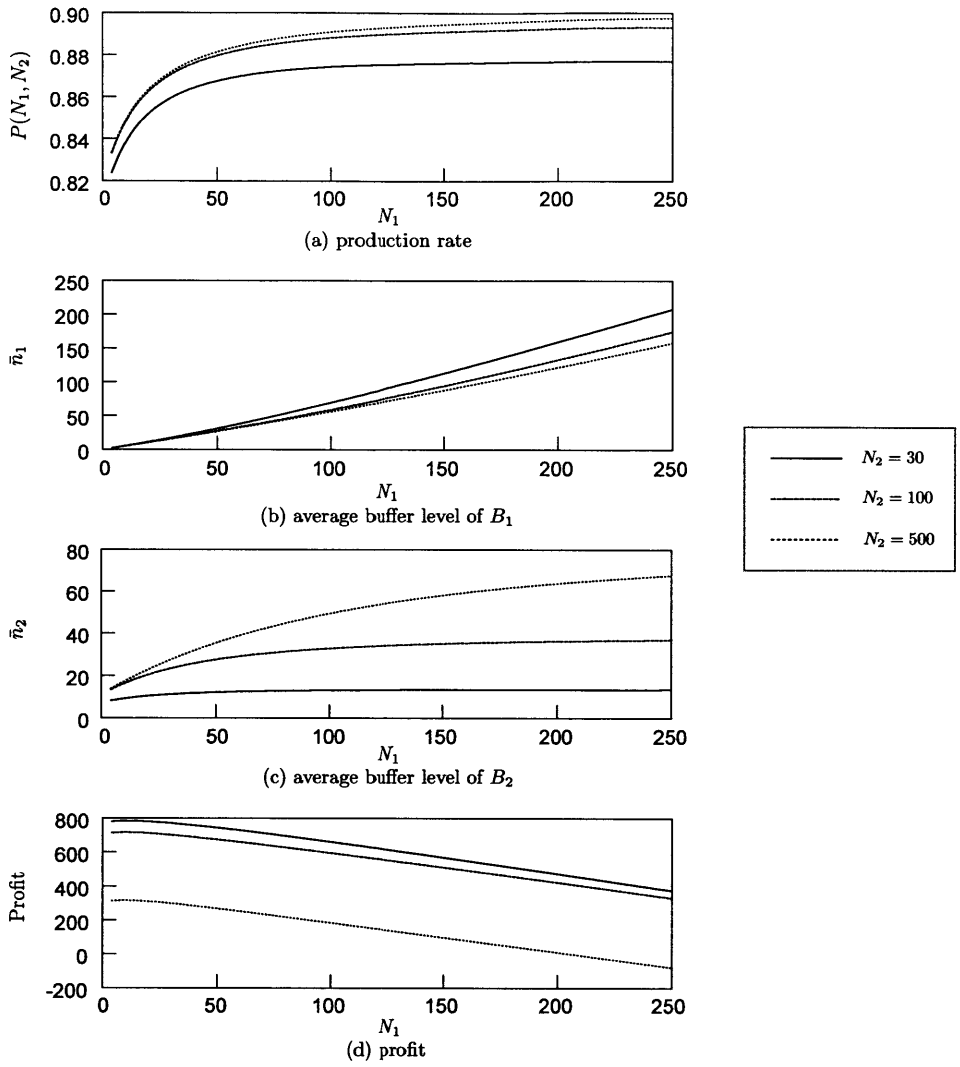


Figure 3-3: Four quantities vs. N_1, Type 1

- Figure 3-3(c) shows $\bar{n}_2(N_1)$. Because $P_3 \geq P_{\{1,2\}}(\infty)$ $\left(> P_{\{1,2\}}(N_1)\right)$, in the two-machine one-buffer line $M_{\{1,2\}} - B_2 - M_3$, M_3 is always faster than the upstream $M_{\{1,2\}}$. As N_1 increases, the $P_{\{1,2\}}(N_1)$ increases (but it is always less than P_3). So, the average inventory of B_2 increases up to an asymptote. $\bar{n}_2$ appears to be a concave function of N_1 in Type 1.

- Figure 3-3(d) shows the profit $J(N_1)$, which appears to be a concave function of N_1. To further study this observation, we compute the second order derivative

of the profit with respect to N_1. Recall that we indicate in Chapter 2 that buffer sizes $\mathbf{N}$ can be treated as continuous variables and the profit of the line $J(\mathbf{N})$ can be considered as a continuously differentiable function. However, since we use the decomposition method of Gershwin (1987a) to evaluate long lines, we do not have analytical solutions of the profit of the line. Therefore, we compute $\mathrm{d}^2J/\mathrm{d}N_1^2 < 0$ according to a forward difference formula,

$$\mathrm{d}^2J/\mathrm{d}N_1^2 = \frac{\mathrm{d}J(N_1+\delta N_1)/\mathrm{d}N_1 - \mathrm{d}J(N_1)/\mathrm{d}N_1}{\delta N_1} \tag{3.1}$$

where $\delta N_1 = .01$ is the step size, while $\mathrm{d}J(N_1+\delta N_1)/\mathrm{d}N_1$ and $\mathrm{d}J(N_1)/\mathrm{d}N_1$ are the first order derivatives which are also computed by the forward difference method. We observe that $\mathrm{d}^2J/\mathrm{d}N_1^2 < 0$. This is consistent with the observation of the concavity of $J(N_1)$. For all three values of N_2, there is a unique optimal value of N_1 between 0 and 50 that maximizes the profit of the three-machine two-buffer line.

Next, we vary N_2 and consider three values of N_1. They are 30, 100, and 500. The four quantities being considered are shown in Figure 3-4.

- Figure 3-4(a) shows $P(N_2)$, which appears to be a concave function of N_2. Since in this type $P_3 \geq P_{\{1,2\}}(\infty)$, the production rate of the entire line is upper bounded by $P_{\{1,2\}}(N_1)$, when N_2 is large enough. Therefore, as N_2 increases, the production rate of the line approaches to $P_{\{1,2\}}(N_1)$. The production rate asymptotes for these three cases (in terms of the value of N_1) are $P_{\{1,2\}}(30)$, $P_{\{1,2\}}(100)$, and $P_{\{1,2\}}(500)$ respectively.

- Figure 3-4(b) shows $\bar{n}_1(N_2)$. Because $P_1 \geq P_{\{2,3\}}(\infty)\,(> P_{\{2,3\}}(N_2))$, in the two-machine one-buffer line $M_1 - B_1 - M_{\{2,3\}}$, M_1 is always faster than $M_{\{2,3\}}$. When N_2 is small, $P_{\{2,3\}}(N_2)$ is small and B_1 tends to be full. As N_2 increases, $P_{\{2,3\}}(N_2)$ increases but it is always less than P_1. The average inventory level $\bar{n}_1$ becomes smaller and finally reaches an asymptote. $\bar{n}_1$ appears to be a convex function of N_2.

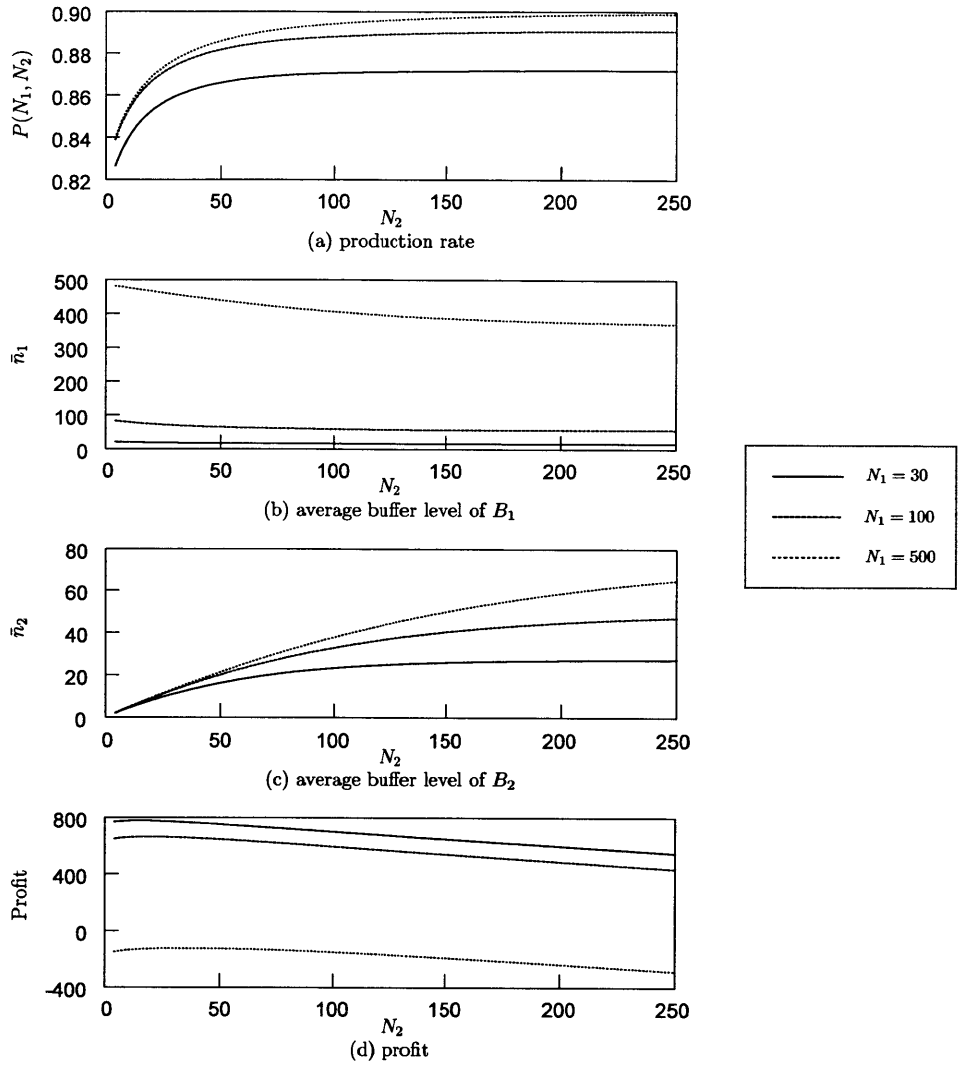


Figure 3-4: Four quantities vs. N_2, Type 1

- Figure 3-4(c) shows $\bar{n}_2(N_2)$. Because $P_3 \geq P_{\{1,2\}}(\infty)$ $(> P_{\{1,2\}}(N_1))$, in the two-machine one-buffer line $M_{\{1,2\}} - B_2 - M_3$, M_3 is always faster than the upstream $M_{\{1,2\}}$. As N_2 increases, $\bar{n}_2$ increases up to an asymptote. It appears that $\bar{n}_2$ is a concave function of N_2.

- Figure 3-4(d) shows the profit $J(N_2)$, which appears to be a concave function of N_2. However, a checking of $\mathrm{d}^2J/\mathrm{d}N_2^2$ indicates that $\mathrm{d}^2J/\mathrm{d}N_2^2$ is negative for some values of N_2 while positive for others, in all three cases (see Figure

3-5). Note from Figure 3-5 that the magnitude of positive $\mathrm{d}^2J/\mathrm{d}N_2^2$ is very small. Therefore, the non-concavity of $J(N_2)$ is hard to be observe in Figure 3-4(d). In addition, since we evaluate long lines by means of decomposition, it is not clear if the tiny positive $\mathrm{d}^2J/\mathrm{d}N_2^2$ is a property of $J(N_2)$ or is due to the approximation made in the decomposition method. If it is indeed a property of $J(N_2)$, then the profit of the line is neither a concave nor a convex function of N_2. For all three values of N_1, there is a unique optimal value of N_2 around 50 that maximizes the profit of the three-machine two-buffer line.

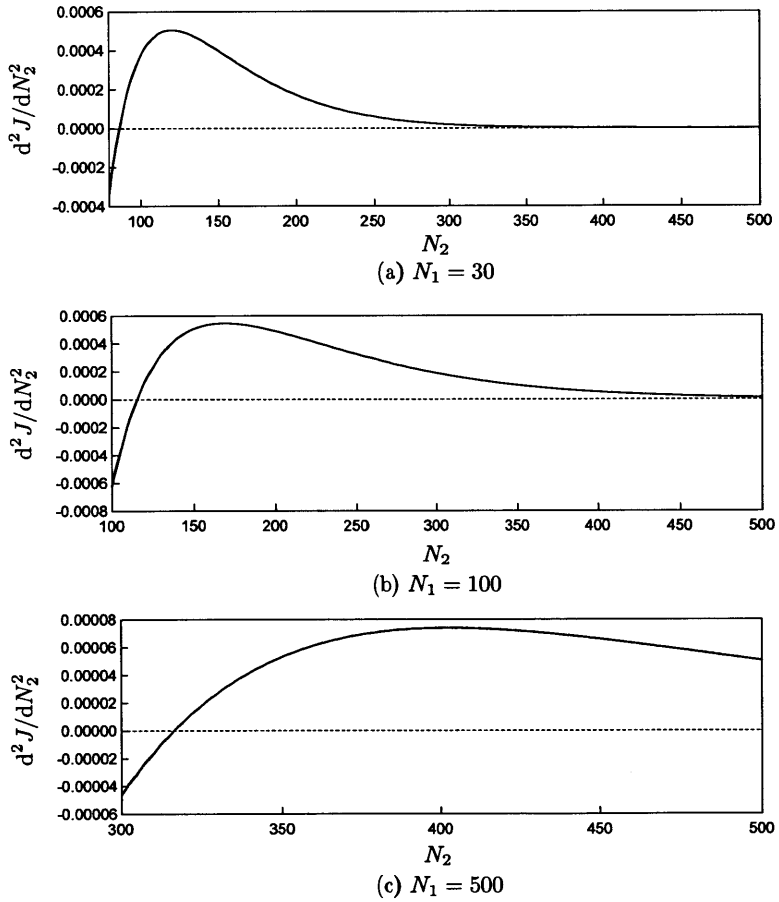


Figure 3-5: $\mathrm{d}^2J/\mathrm{d}N_2^2$ vs. N_2, Type 1

3.4.2 Type 2

Recall that Type 2 means that $P_3 \geq P_{\{1,2\}}(\infty)$ and $P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}(\infty)$. These require

$$\frac{1}{1+\frac{p_2}{r_2}+\frac{p_3}{r_3}} < \frac{r_1}{r_1+p_1} < \min\left(\frac{r_2}{r_2+p_2}, \frac{r_3}{r_3+p_3}\right).$$

Consider the example shown in Table 3.5. We first vary N_1. The three values of N_2 we consider are 30, 100, and 500. The four quantities being considered are shown in Figures 3-6.

Table 3.5: An example of Type 2

machine	M_1	M_2	M_3
r_i	.8	.1	.1
p_i	.096	.01	.01
P_i	.893	.909	.909

- Figure 3-6(a) shows the production rate $P(N_1)$. In this type $P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}(\infty)$. Thus, for small N_2, $P_1 > P_{\{2,3\}}(N_2)$ and the production rate of the line is upper bounded by $P_{\{2,3\}}(N_2)$. However, for large N_2, $P_1 < P_{\{2,3\}}(N_2)$ and the production rate is upper bounded by P_1. For instance, when $N_2 = 500$, the production rate is upper bounded by $P_1 = .893$. However, when $N_2 = 30$, the production rate is upper bounded by $P_{\{2,3\}}(30)$ that is less than .893. The production rate appears to be a concave function of N_1.

- Figure 3-6(b) shows $\bar{n}_1(N_1)$. For small N_2, $P_1 > P_{\{2,3\}}(N_2)$ therefore M_1 is faster than $M_{\{2,3\}}$. In this case, as N_1 increases, $\bar{n}_1$ increases without a limit. However, for large N_2, $P_1 < P_{\{2,3\}}(N_2)$ therefore M_1 is slower than $M_{\{2,3\}}$. In this case, as N_1 increases, $\bar{n}_1$ increases up to an asymptote. Therefore $\bar{n}_1$ appears to be either a concave or a convex function of N_1, depending on N_2.

- Figure 3-6(c) illustrates $\bar{n}_2(N_1)$. Since $P_3 \geq P_{\{1,2\}}(\infty)$ $\left(> P_{\{1,2\}}(N_1)\right)$, in the two-machine one-buffer line $M_{\{1,2\}} - B_2 - M_3$, M_3 is always faster than the

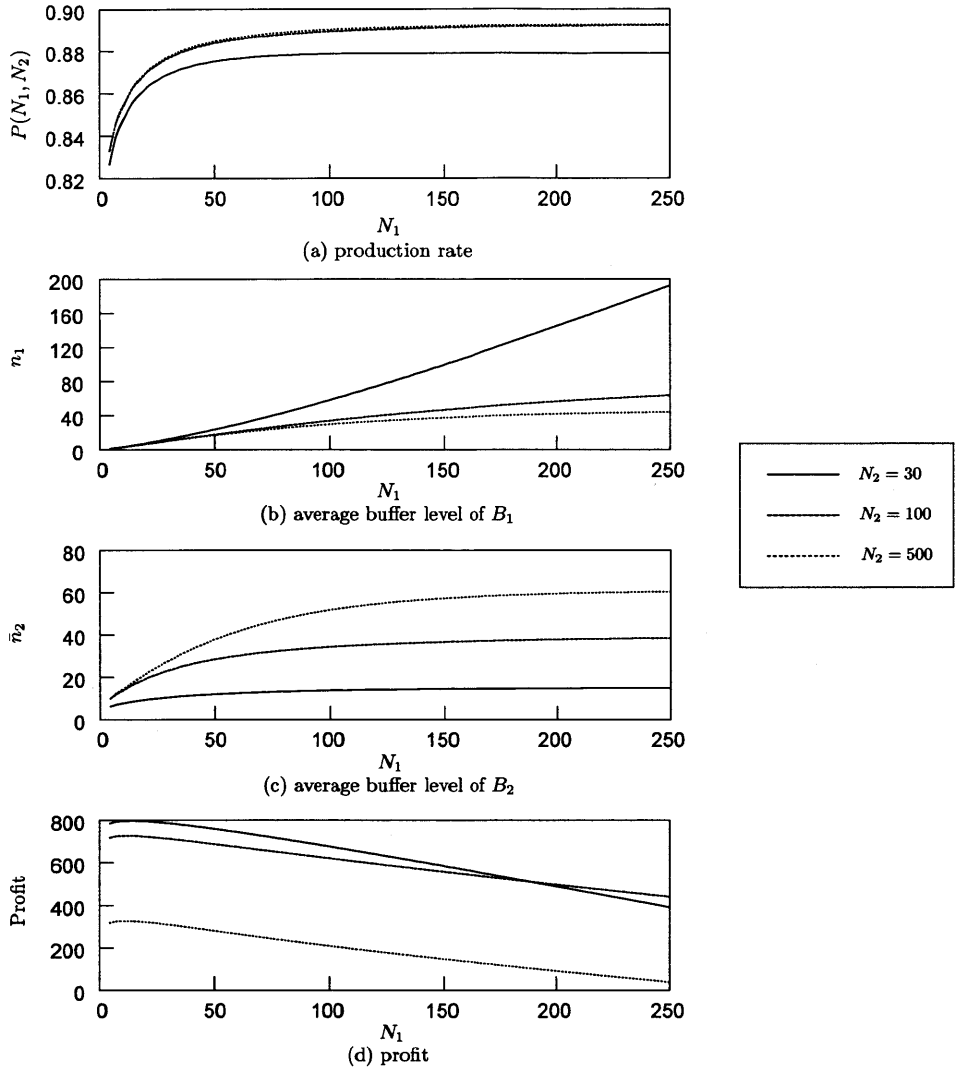
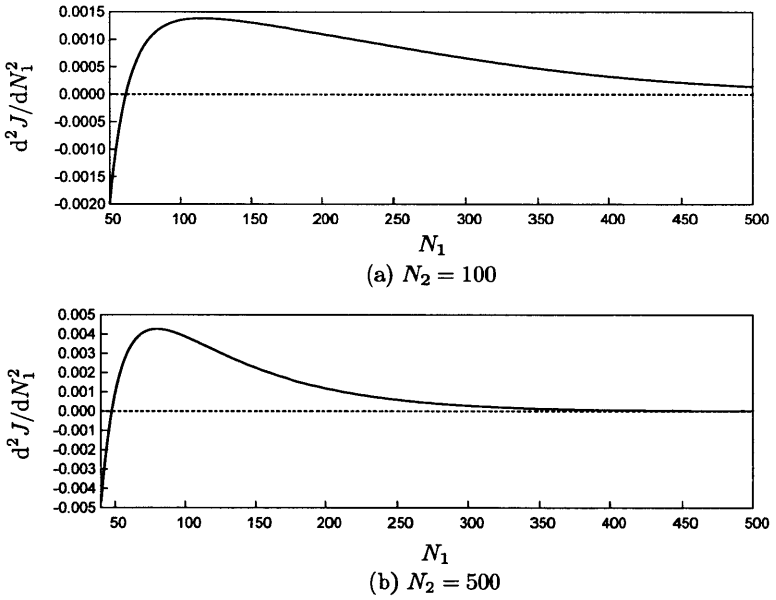


Figure 3-6: Four quantities vs. N_1, Type 2

upstream $M_{\{1,2\}}$. As N_1 increases, $P_{\{1,2\}}(N_1)$ increases (but it is always less than P_3). So, $\bar{n}_2$ increases up to an asymptote. It appears that $\bar{n}_2$ to be a concave function of N_1.

- Figure 3-6(d) shows the profit $J(N_1)$. $J(N_1)$ appears to be a concave function of N_1 when N_2 is small, while it is neither a concave nor a convex function of N_1 when N_2 is large (although this is hard to see in the figure). We further confirm these observations by studying $\mathrm{d}^2J/\mathrm{d}N_1^2$ and find that $\mathrm{d}^2J/\mathrm{d}N_1^2 < 0$

when $N_2 = 30$, while when $N_2 = 100$ and 500, $\mathrm{d}^2J/\mathrm{d}N_1^2$ is positive for some values of N_1 but negative for others (see Figure 3-7). Because of the large inventory cost of B_1 when N_1 is large but N_2 is small, the solid curve (i.e., the $N_2 = 30$ case) eventually crosses the other two curves. For all three values of N_2, there is a unique optimal value of N_1 that maximizes the profit.



(a) $N_2 = 100$

(b) $N_2 = 500$

Figure 3-7: $\mathrm{d}^2J/\mathrm{d}N_1^2$ vs. N_1, Type 2

Next, we vary N_2 and consider three values of N_1. They are 100, 500, and 1000. The four quantities being considered are shown in Figure 3-8.

- Figure 3-8(a) shows $P(N_2)$, which appears to be a concave function of N_2. In this type $P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}(\infty)$. For N_1 that is large enough (e.g., $N_1 = 500$ or 1000), the production rate of the line is approximately $P_{\{2,3\}}(N_2)$ when N_2 is small and increases as N_2 increases. However, when N_2 is large, the production rate is bounded by P_1. This explains why when N_2 is larger than a certain value (say N_2^F), the production rate turns to a constant value (P_1) for all $N_2 \geq N_2^F$ instead of keeping increasing up to an asymptote.

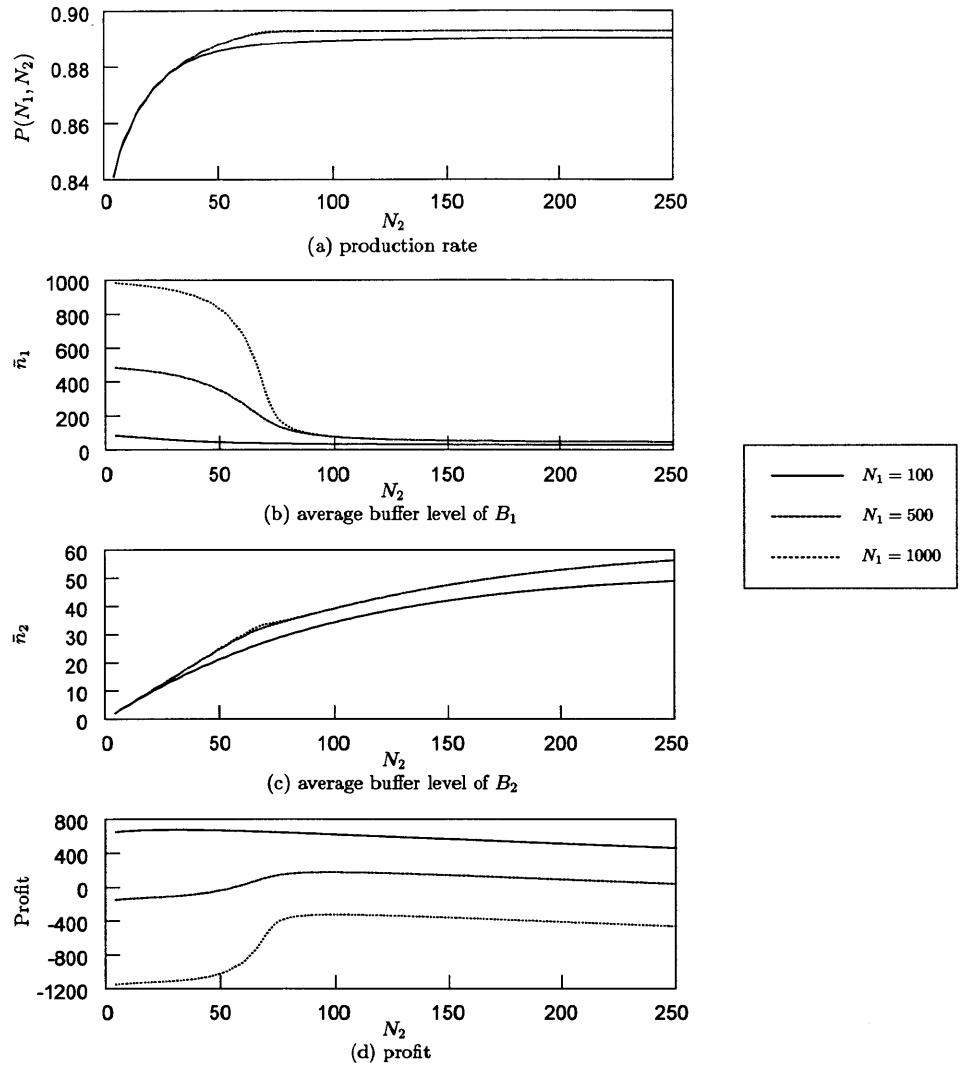


Figure 3-8: Four quantities vs. N_2, Type 2

- Figure 3-8(b) shows $\bar{n}_1(N_2)$. Let us consider the case where $N_1 = 1000$. It shows the most dramatic behavior of $\bar{n}_1$. Recall that $P_{\{2,3\}}(0) < P_1 < P_{\{2,3\}}(\infty)$. Thus, when N_2 is small, $P_1 > P_{\{2,3\}}(N_2)$. In other words, in $M_1 - B_1 - M_{\{2,3\}}$, M_1 is faster than $M_{\{2,3\}}$ and Buffer B_1 tends to be full. So, $\bar{n}_1$ is close to $N_1 = 1000$ when N_2 is small. However, when N_2 is large, $P_1 < P_{\{2,3\}}(N_2)$. In this case, M_1 is slower than $M_{\{2,3\}}$ and Buffer B_1 tends to be empty. Thus, $\bar{n}_1$ is very small when N_2 is large. There is a dramatic drop of $\bar{n}_1$ as N_2 increases from 50 to 100. This is because when N_2 is small, M_1 is faster; while when N_2 is large,

$M_{\{2,3\}}$ is faster. The dramatic drop of $\bar{n}_1$ is due to the shift of the faster machine in $M_1 - B_1 - M_{\{2,3\}}$. When N_2 is somewhere around 70, $P_1 = P_{\{2,3\}}(N_2)$ and $\bar{n}_1 = 0.5N_1 = 500$. A similar but less drastic drop in $\bar{n}_1$ can be seen when $N_1 = 500$. However, when N_1 is small, the dramatic drop cannot be observed. $\bar{n}_1$ is neither a concave nor a convex function of N_2.

- Figure 3-8(c) shows $\bar{n}_2(N_2)$. Because $P_3 \geq P_{\{1,2\}}(\infty)$ $\left(> P_{\{1,2\}}(N_1)\right)$, in the two-machine one-buffer line $M_{\{1,2\}} - B_2 - M_3$, M_3 is always faster than $M_{\{1,2\}}$. As we increase N_2, $\bar{n}_2$ increases up to an asymptote. In addition, $\bar{n}_2$ is neither a concave nor a convex function of N_2 when N_1 is large (although this is hard to see in the figure).

- Figure 3-8(d) shows the profit $J(N_2)$. $J(N_2)$ appears to be a concave function of N_2 for small N_1. However, when N_1 is large, $J(N_2)$ is neither a concave nor a convex function of N_2. This is clear in the figure for $N_1 = 500$ and 1000. For $N_1 = 100$, a study of $\mathrm{d}^2J/\mathrm{d}N_2^2$ (see Figure 3-9) shows that it can be both positive and negative depending on N_2. If this is indeed a property of $J(N_2)$, then it is neither a concave nor a convex function of N_2 when N_1 is small as well. Otherwise, it may be due to the inaccuracy of decomposition. For all three values of N_1, there is a unique optimal value of N_2 that maximizes the profit of the three-machine two-buffer line.

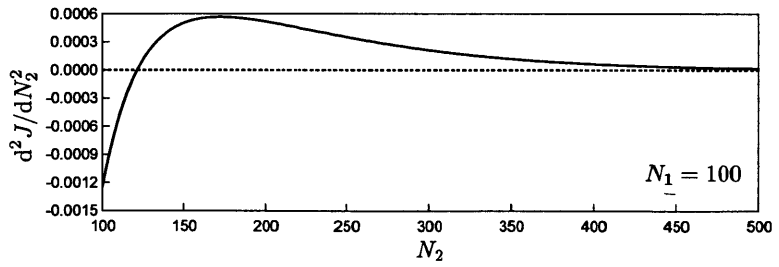


Figure 3-9: $\mathrm{d}^2J/\mathrm{d}N_2^2$ vs. N_2, Type 2

3.4.3 Type 3

Type 3 means that $P_3 \geq P_{\{1,2\}}(\infty)$ and $P_1 \leq P_{\{2,3\}}(0)$. These conditions require

$$\frac{r_1}{r_1 + p_1} \leq \frac{1}{1 + \frac{p_2}{r_2} + \frac{p_3}{r_3}}.$$

Consider the example shown in Table 3.6. We first vary N_1. The three values of N_2 we consider are 30, 100, and 500. The four quantities being considered are shown in Figure 3-10.

Table 3.6: An example of Type 3

machine	M_1	M_2	M_3
r_i	.07	.12	.12
p_i	.01	.008	.008
P_i	.875	.938	.938

- Figure 3-10(a) illustrates the production rate, which appears to be a concave function of N_1. Since $P_1 \leq P_{\{2,3\}}(0) \left(< P_{\{2,3\}}(N_2)\right)$, in $M_1 - B_1 - M^d(2)$, M_1 is always slower than $M_{\{2,3\}}$ and the production rate is upper bounded by $P_1 = .875$ for all three cases.

- Figure 3-10(b) shows $\bar{n}_1(N_1)$. Because $P_1 \leq P_{\{2,3\}}(0) \left(< P_{\{2,3\}}(N_2)\right)$, M_1 is always slower than $M_{\{2,3\}}$ regardless of N_2. Therefore, as N_1 increases, $\bar{n}_1$ increases up to an asymptote. $\bar{n}_1$ appears to be a concave function of N_1.

- Figure 3-10(c) illustrates $\bar{n}_2(N_1)$. Because $P_3 \geq P_{\{1,2\}}(\infty) \left(> P_{\{1,2\}}(N_1)\right)$, in the two-machine one-buffer line $M_{\{1,2\}} - B_2 - M_3$, M_3 is always faster than the $M_{\{1,2\}}$. As N_1 increases, $P_{\{1,2\}}(N_1)$ increases (but it is always less than P_3). So, $\bar{n}_2$ increases up to an asymptote. It appears that $\bar{n}_2$ is a concave function of N_1.

- Figure 3-10(d) shows that the profit $J(N_1)$ appears to be a concave function of N_1. However, a checking of $\mathrm{d}^2J/\mathrm{d}N_1^2$ indicates that $\mathrm{d}^2J/\mathrm{d}N_1^2$ is negative

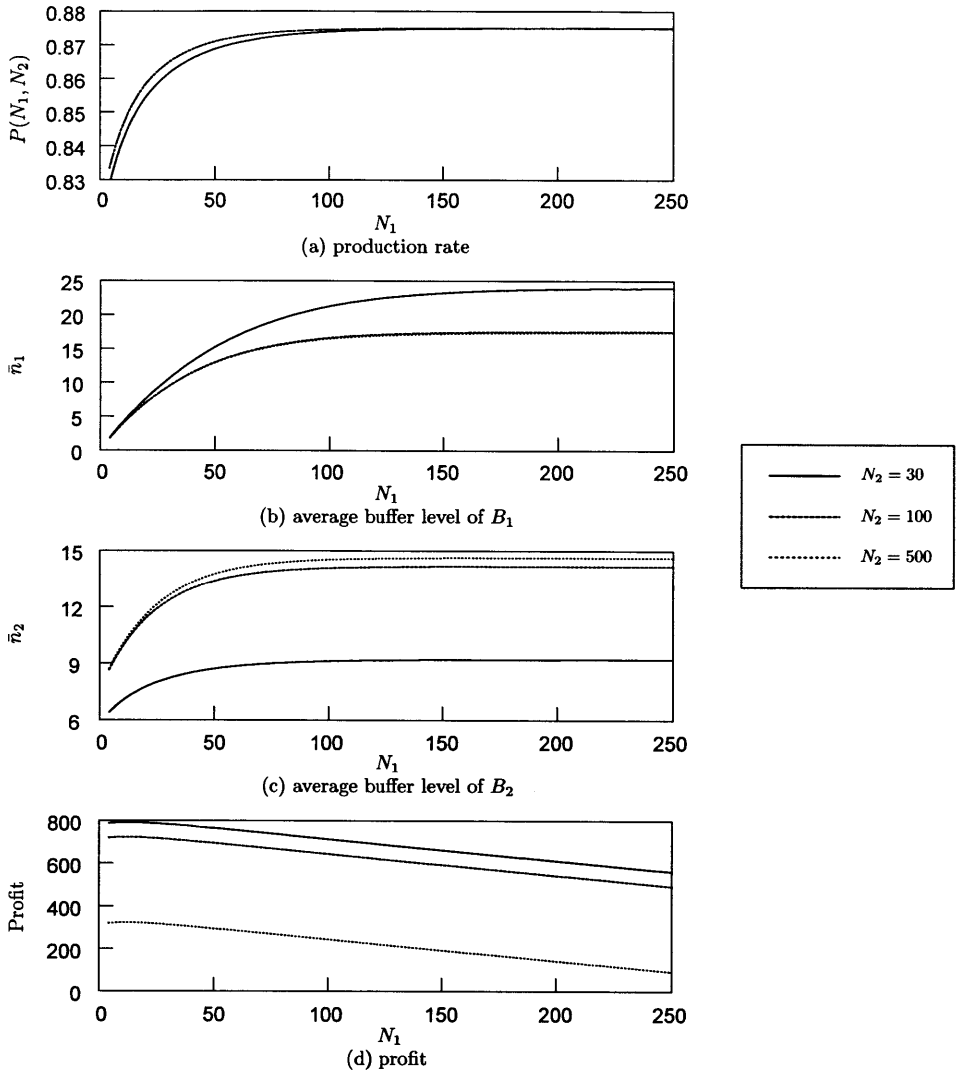
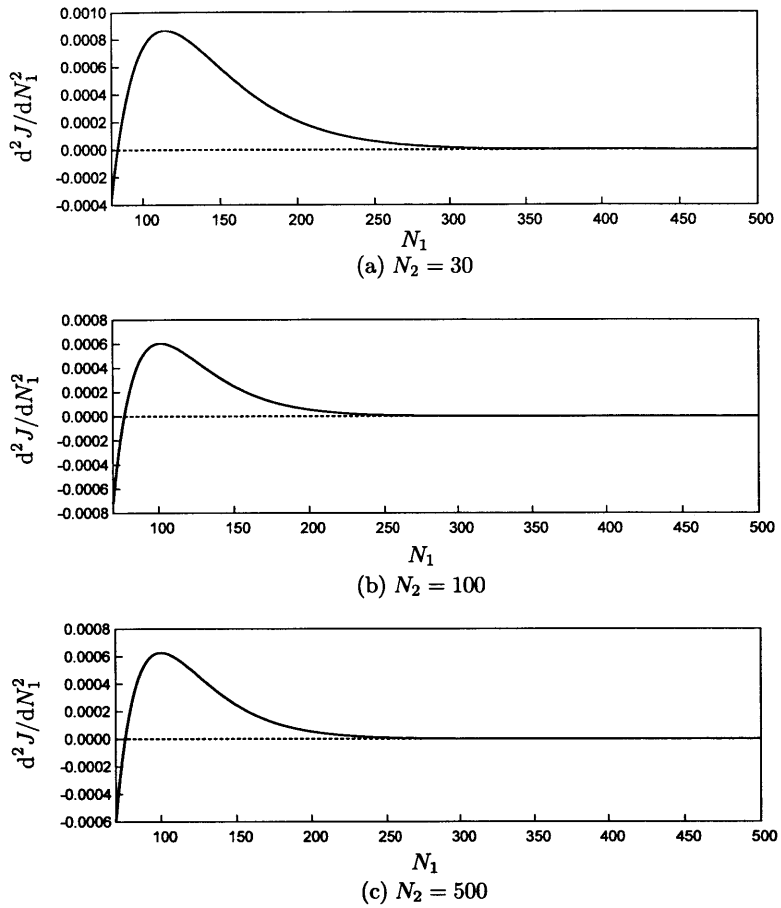


Figure 3-10: Four quantities vs. N_1, Type 3

for some values of N_1 while positive for others, in all three cases (see Figure 3-11). Note from Figure 3-11 that the magnitude of positive $\mathrm{d}^2J/\mathrm{d}N_1^2$ is very small. Therefore, the non-concavity of $J(N_1)$ is hardly to be observed in Figure 3-10(d). It is not clear if the tiny positive $\mathrm{d}^2J/\mathrm{d}N_1^2$ is indeed a property of $J(N_1)$ or is due to the approximation made in the decomposition method. If it is indeed a property of $J(N_1)$, then the profit of the line is neither a concave nor a convex function of N_1. For all three values of N_2, there is a unique optimal value of N_1 that maximizes the profit of the three-machine two-buffer line.



(a) $N_2 = 30$

(b) $N_2 = 100$

(c) $N_2 = 500$

Figure 3-11: d^2J/dN_1^2 vs. N_1, Type 3

Next, we vary N_2 and consider three values of N_1. They are 30, 100, and 500. The four quantities being considered are shown in Figure 3-12.

- Figure 3-12(a) indicates that production rate appears to be a concave function of N_2. Since $P_3 \geq P_{\{1,2\}}(\infty)$, the production rate is upper bounded by $P_{\{1,2\}}(N_1)$. Therefore, as N_2 increases, the production rate of the line approaches to $P_{\{1,2\}}(N_1)$. When N_1 is as large as 500, for instance, the production rate approaches to $P_{\{1,2\}}(500) \approx P_{\{1,2\}}(\infty) = \min(P_1, P_2) = P_1 = .875$.

- Figure 3-12(b) shows $\bar{n}_1(N_2)$. Recall that $P_1 \leq P_{\{2,3\}}(0)$ $\left(< P_{\{2,3\}}(N_2)\right)$. Thus, in $M_1 - B_1 - M_{\{2,3\}}$, M_1 is always slower than $M_{\{2,3\}}$ and $\bar{n}_1$ tends to be small compared to N_1. In addition, as N_2 increases, $M_{\{2,3\}}$ becomes faster and

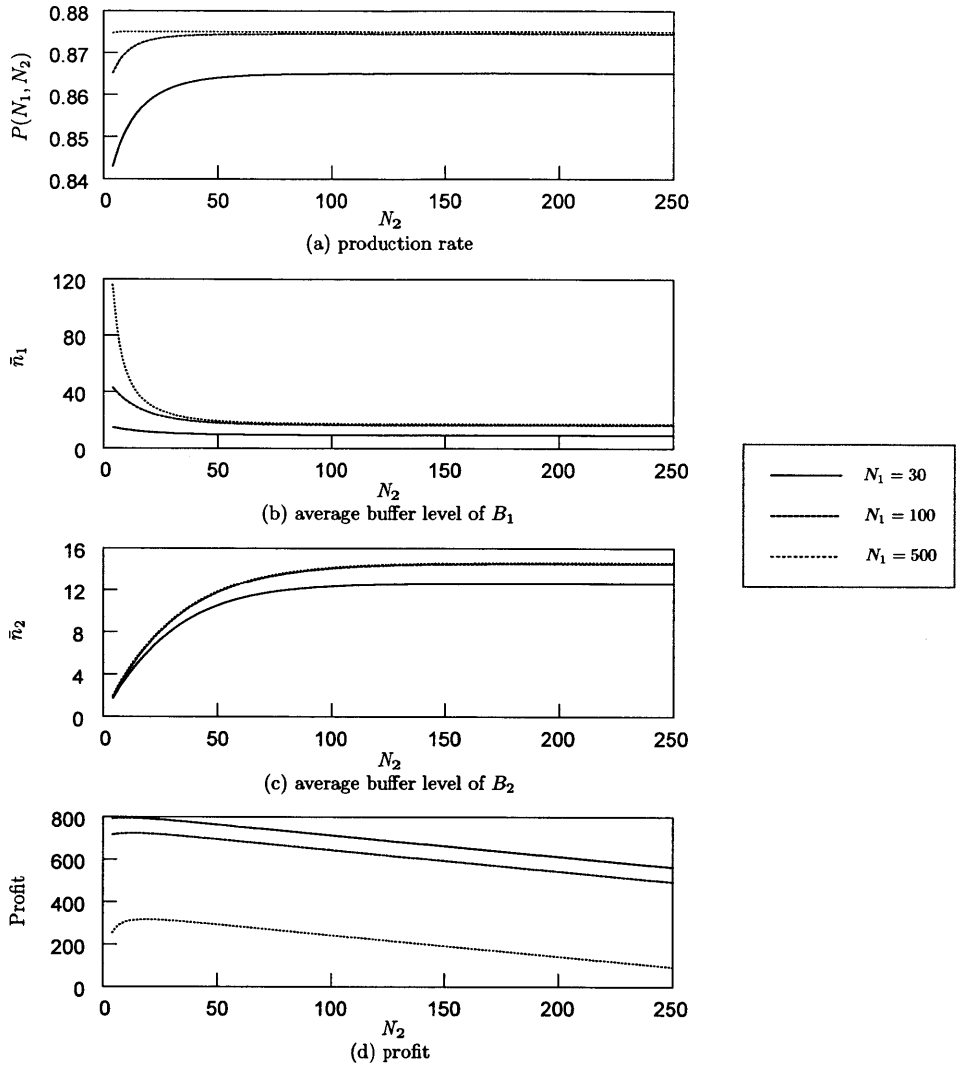


Figure 3-12: Four quantities vs. N_2, Type 3

therefore $\bar{n}_1$ becomes even smaller. Consequently, $\bar{n}_1$ decreases monotonically with N_2 and finally reaches an asymptote. It appears that $\bar{n}_1$ is a convex function of N_2.

- Figure 3-12(c) shows $\bar{n}_2(N_1)$. Because $P_3 \geq P_{\{1,2\}}(\infty)$ $\left(> P_{\{1,2\}}(N_1)\right)$, in $M_{\{1,2\}} - B_2 - M_3$, M_3 is always faster than the upstream $M_{\{1,2\}}$. As we increase N_2, $\bar{n}_2$ increases up to an asymptote. It appears that $\bar{n}_2$ is a concave function of N_2.

- Figure 3-12(d) illustrates the profit $J(N_2)$. The profit appears to be a concave

function of N_2. However, a checking of $\mathrm{d}^2J/\mathrm{d}N_2^2$ indicates that $\mathrm{d}^2J/\mathrm{d}N_2^2$ is negative for some values of N_2 while positive for others, in all three cases (see Figure 3-13). Note from Figure 3-13 that the magnitude of positive $\mathrm{d}^2J/\mathrm{d}N_2^2$ is very small. Therefore, the non-concavity of $J(N_2)$ is hardly to be observed in Figure 3-12(d). It is not clear if the tiny positive $\mathrm{d}^2J/\mathrm{d}N_2^2$ is indeed a property of $J(N_2)$ or is due to the approximation made in the decomposition method. If it is indeed a property of $J(N_2)$, then the profit of the line is neither a concave nor a convex function of N_2. For all three values of N_1, there is a unique optimal value of N_2 that maximizes the profit of the line.

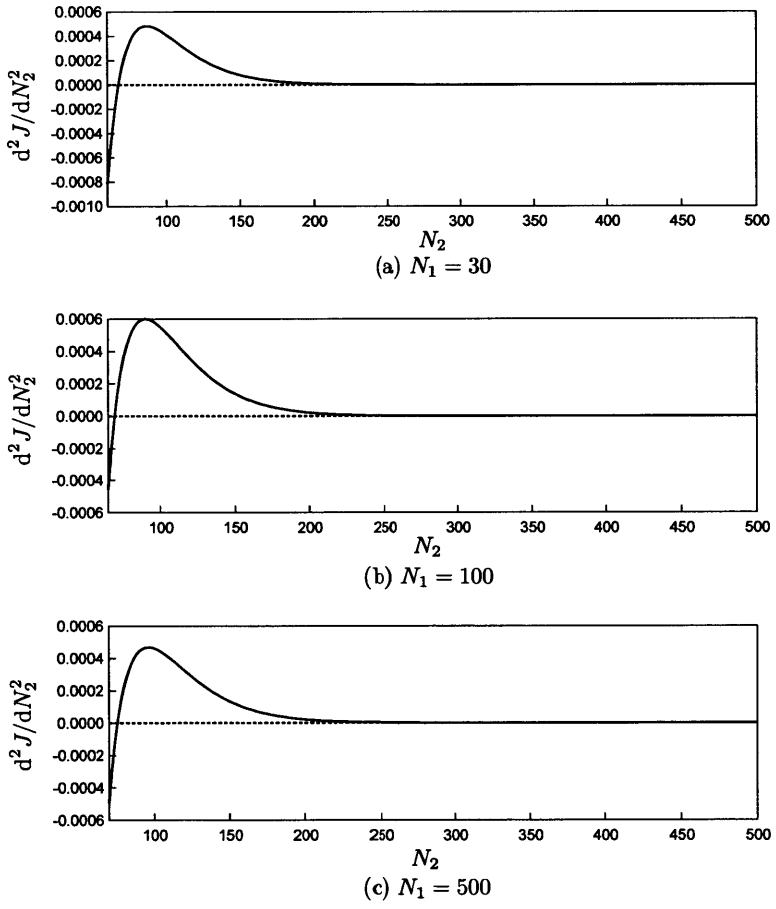


Figure 3-13: $\mathrm{d}^2J/\mathrm{d}N_2^2$ vs. N_2, Type 3

3.4.4 Type 4

Recall that Type 4 lines are reverses of Type 2 lines. In particular, in Type 4, $P_{\{1,2\}}(0) < P_3 < P_{\{1,2\}}(\infty)$ and $P_1 \geq P_{\{2,3\}}(\infty)$. These conditions require

$$\frac{1}{1+\frac{p_1}{r_1}+\frac{p_2}{r_2}} < \frac{r_3}{r_3+p_3} \leq \min\left(\frac{r_1}{r_1+p_1}, \frac{r_2}{r_2+p_2}\right).$$

Consider the example shown in Table 3.7. (Note that Type 4 with these parameters is a reverse case of the Type 2 example of Table 3.5 in Section 3.4.2.) We first vary N_1. The three values of N_2 we consider are 100, 500, and 1000. The four quantities being considered are shown in Figures 3-14. (Compare this with Figure 3-8.)

Table 3.7: An example of Type 4

machine	M_1	M_2	M_3
r_i	.1	.1	.8
p_i	.01	.01	.096
P_i	.909	.909	.893

- Figure 3-14(a) shows the production rate, which appears to be a concave function of N_1. In this type $P_{\{1,2\}}(0) < P_3 < P_{\{1,2\}}(\infty)$. For N_2 that is large enough (e.g., $N_2 = 500$ or 1000), the production rate of the line is approximately $P_{\{1,2\}}(N_1)$ when N_1 is small and increases as N_1 increases. However, when N_1 is large, the production rate is bounded by P_3. This explains why when N_1 is larger than a certain value (say N_1^F), the production rate turns to a constant value (P_3) for all $N_1 \geq N_1^F$ instead continuing to increase up to an asymptote.

- Figure 3-14(b) shows $\bar{n}_1(N_1)$. Because $P_1 \geq P_{\{2,3\}}(\infty)\,(> P_{\{2,3\}}(N_2))$, M_1 is always faster than $M_{\{2,3\}}$ regardless of N_2 in $M_1 - B_1 - M_{\{2,3\}}$. Therefore, as N_1 increases, $\bar{n}_1$ increases without a limit. $\bar{n}_1$ appears to be a convex function of N_1.

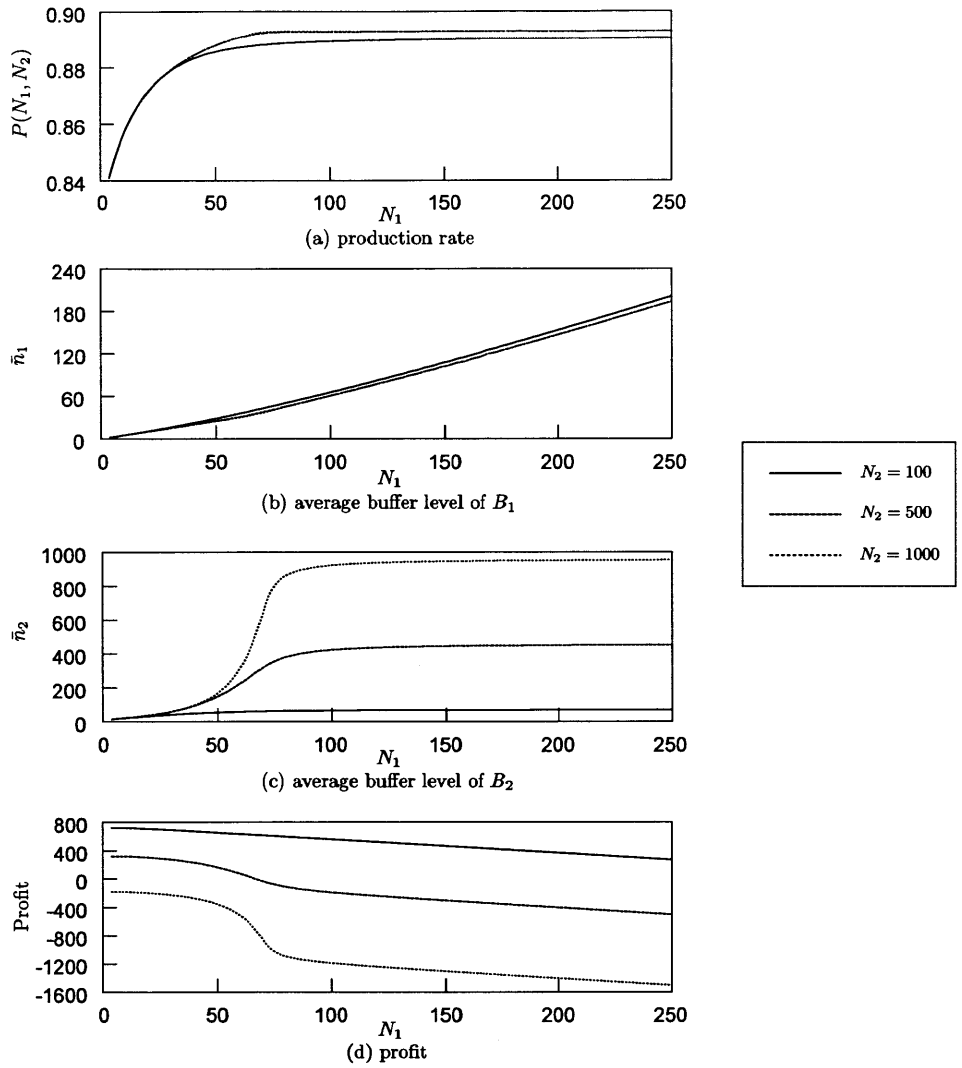


Figure 3-14: Four quantities vs. N_1, Type 4

- Figure 3-14(c) shows $\bar{n}_2(N_1)$. The case where $N_2 = 1000$ shows the most dramatic behavior of $\bar{n}_2$. Recall that $P_{\{1,2\}}(0) < P_3 < P_{\{1,2\}}(\infty)$. Thus, when N_1 is small, $P_3 > P_{\{1,2\}}(N_1)$. In other words, in $M_{\{1,2\}} - B_2 - M_3$, M_3 is faster than $M_{\{1,2\}}$ therefore the buffer level tends to be small. So, $\bar{n}_2$ is small when N_1 is small. However, when N_1 is large, $P_3 < P_{\{1,2\}}(N_1)$. In this case, M_3 is slower than $M_{\{1,2\}}$ and the buffer level tends to be high. There is a dramatic increase of $\bar{n}_2$ as N_1 increases from 50 to 100. This is because when N_1 is small, M_3 is faster; while when N_1 is large, $M_{\{1,2\}}$ is faster. The dramatic increase of

$\bar{n}_2$ is due to the shift of the faster machine in $M_{\{1,2\}} - B_2 - M_3$. When N_1 is somewhere around 70, $P_{\{1,2\}}(N_1) = P_3$ and $\bar{n}_2 = 0.5N_2 = 500$. A similar but less dramatic increase of $\bar{n}_2$ can be observed when $N_2 = 500$. However, when N_2 is small, the dramatic increase is not observed[1]. Therefore, $\bar{n}_2$ is neither a concave nor a convex function of N_1.

- Figure 3-14(d) shows the profit $J(N_1)$. It is helpful to point out that given the line parameters, the profit is monotonically decreasing with N_1 when N_2 is large. This means that, as N_1 increases, the increment of buffer space and average inventory cost outweighs the increment of revenue associated with the production rate. However, for all three values of N_2, there is a unique optimal value of N_1 that maximizes the profit of the line. The monotonically decreasing curves suggest that Buffer B_1 is undesirable. The profit appears to be a concave function of N_1 for small N_2. However, a study of $\mathrm{d}^2J/\mathrm{d}N_1^2$ (see Figure 3-15) shows that $\mathrm{d}^2J/\mathrm{d}N_1^2$ can be both positive and negative depending on N_1 when N_2 is small. It is not clear if the tiny positive value of $\mathrm{d}^2J/\mathrm{d}N_1^2$ is due to the property of $J(N_1)$ or the approximate decomposition method. When N_2 is large, the profit is neither a concave nor a convex function of N_1.

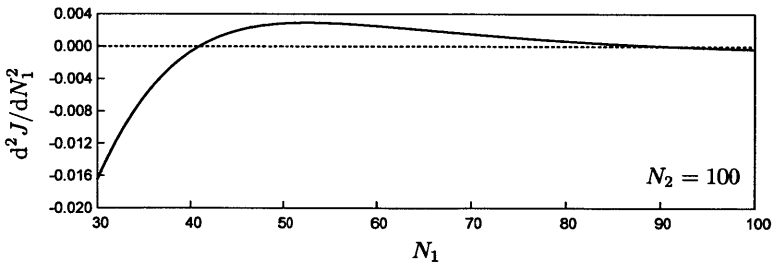


Figure 3-15: $\mathrm{d}^2J/\mathrm{d}N_1^2$ vs. N_1, Type 4

Next, we vary N_2 and consider three values of N_1. They are 30, 100, and 500. The four quantities being considered are shown in Figure 3-16. (Compare this with Figure 3-6.)

[1]Compare the analysis here with the analysis for Figure 3-8(b) of Type 2.

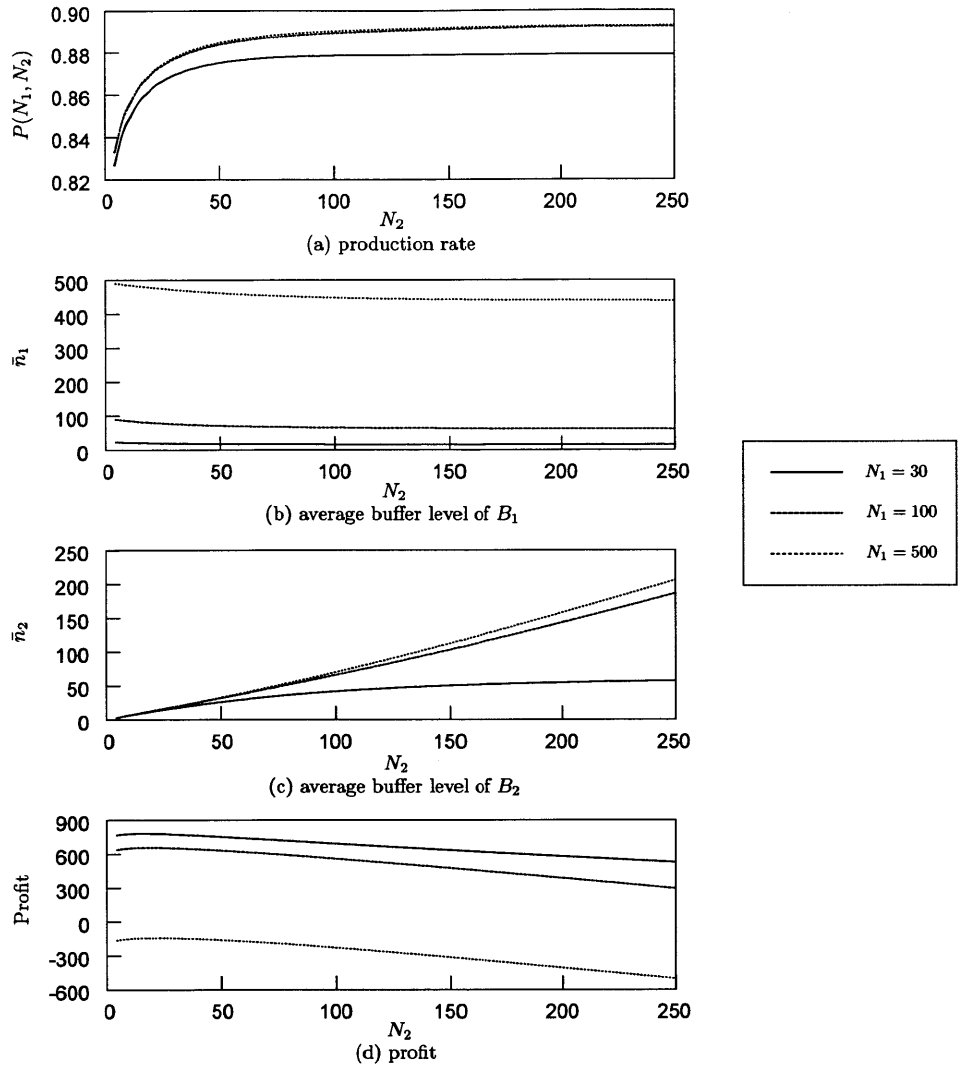


Figure 3-16: Four quantities vs. N_2, Type 4

- Figure 3-16(a) shows that the production rate appears to be a concave function of N_2. In this type $P_{\{1,2\}}(0) < P_3 < P_{\{1,2\}}(\infty)$. Thus, for small N_1, $P_3 > P_{\{1,2\}}(N_1)$ and therefore the production rate is upper bounded by $P_{\{1,2\}}(N_1)$. On the other hand, for large N_1s, $P_3 < P_{\{1,2\}}(N_1)$ and therefore the production rate is upper bounded by P_3. As we can see, for instance, when $N_1 = 100$ or 500, the production rate is upper bounded by $P_3 = .893$. However, when $N_1 = 30$, the production rate is upper bounded by $P_{\{1,2\}}(30)$ that is less than .893.

- Figure 3-16(b) shows $\bar{n}_1(N_2)$. In this type $P_1 \geq P_{\{2,3\}}(\infty)$ $\left(> P_{\{2,3\}}(N_2)\right)$. Thus, in the two-machine one-buffer line $M_1 - B_1 - M_{\{2,3\}}$, M_1 is always faster than $M_{\{2,3\}}$ and $\bar{n}_1$ tends to be close to N_1. In addition, as N_2 increases, $M_{\{2,3\}}$ becomes faster and therefore $\bar{n}_1$ gets smaller. Consequently, $\bar{n}_1$ decreases monotonically with N_2 and finally reaches an asymptote, and N_1. It appears that $\bar{n}_1$ is a convex function of N_2.

- Figure 3-16(c) illustrates $\bar{n}_2(N_2)$. For small N_1, $P_3 > P_{\{1,2\}}(N_1)$ therefore M_3 is faster than $M_{\{1,2\}}$. In this case, as N_2 increases, $\bar{n}_2$ increases up to an asymptote. However, for large N_1, $P_3 < P_{\{1,2\}}(N_1)$ therefore M_3 is slower than $M_{\{1,2\}}$. In this case, as N_2 increases, $\bar{n}_2$ increases without a limit. It appears that $\bar{n}_2$ can be either a concave or a convex function of N_2 depending on the value of N_1.

- Figure 3-16(d) shows the profit $J(N_2)$. For all three values of N_1, there is a unique optimal value of N_2 (between 0 and 50) that maximizes the profit of the three-machine two-buffer line. In particular, the profit appears to be a concave function of N_2 for large N_1. However, when N_1 is small, the profit is neither concave nor convex (although it is hard to see from the figure). These observations are further confirmed by studying $\mathrm{d}^2J/\mathrm{d}N_2^2$. When N_1 is large, $\mathrm{d}^2J/\mathrm{d}N_2^2 < 0$. When N_1 is small (e.g., $N_1 = 30$), $\mathrm{d}^2J/\mathrm{d}N_2^2$ is positive for some values of N_2 but negative for others (see Figure 3-17).

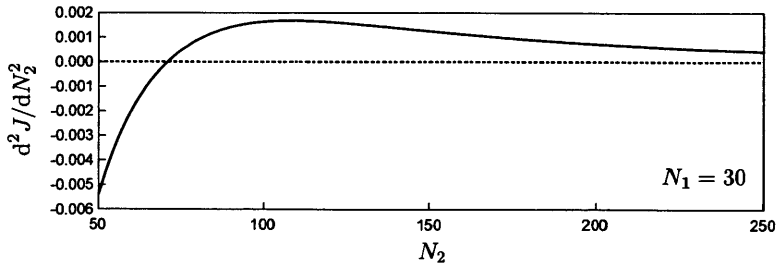


Figure 3-17: $\mathrm{d}^2J/\mathrm{d}N_2^2$ vs. N_2, Type 4

3.4.5 Type 5

In Type 5, $P_3 \leq P_{\{1,2\}}(0)$ and $P_1 \geq P_{\{2,3\}}(\infty)$. These conditions require

$$\frac{r_3}{r_3 + p_3} \leq \frac{1}{1 + \frac{p_1}{r_1} + \frac{p_2}{r_2}}.$$

Type 5 lines are reverses of Type 3 lines. Consider the example shown in Table 3.8. We first vary N_1. The three values of N_2 we consider are 30, 100, and 500. The four quantities being considered are shown in Figure 3-18. (Compare this with Figure 3-12.)

Table 3.8: An example of Type 5

machine	M_1	M_2	M_3
r_i	.12	.15	.07
p_i	.009	.009	.01
P_i	.930	.943	.875

- Figure 3-18(a) shows the production rate, which appears to be a concave function of N_1. In this type $P_1 \geq P_{\{2,3\}}(\infty)$ $(> P_{\{2,3\}}(N_2))$. Therefore M_1 is always faster than $M_{\{2,3\}}$ in $M_1 - B_1 - M_{\{2,3\}}$ and the production rate is upper bounded by $P_{\{2,3\}}(N_2)$ when N_1 is sufficiently large. In addition, when N_2 is large, $P(N_1)$ approaches to $P_{\{2,3\}}(N_2) \approx \min\{P_2, P_3\} = P_3 = .875$.

- Figure 3-18(b) shows $\bar{n}_1(N_1)$. Because $P_1 \geq P_{\{2,3\}}(\infty)$ $(> P_{\{2,3\}}(N_2))$, M_1 is always faster than $M_{\{2,3\}}$ regardless of N_2. Therefore, as N_1 increases, $\bar{n}_1$ increases without a limit. $\bar{n}_1$ appears to be a convex function of N_1.

- Figure 3-18(c) illustrates $\bar{n}_2(N_1)$. Because $P_3 \leq P_{\{1,2\}}(0)$ $(< P_{\{1,2\}}(N_1))$, M_3 is always slower than the upstream $M_{\{1,2\}}$ in $M_{\{1,2\}} - B_2 - M_3$. As N_1 increases, $M_{\{1,2\}}$ becomes faster and faster. Therefore, $\bar{n}_2$ is close to N_2 (i.e., Buffer B_2 tends to be full) and reaches an asymptote. It appears that $\bar{n}_2$ is a concave function of N_1.

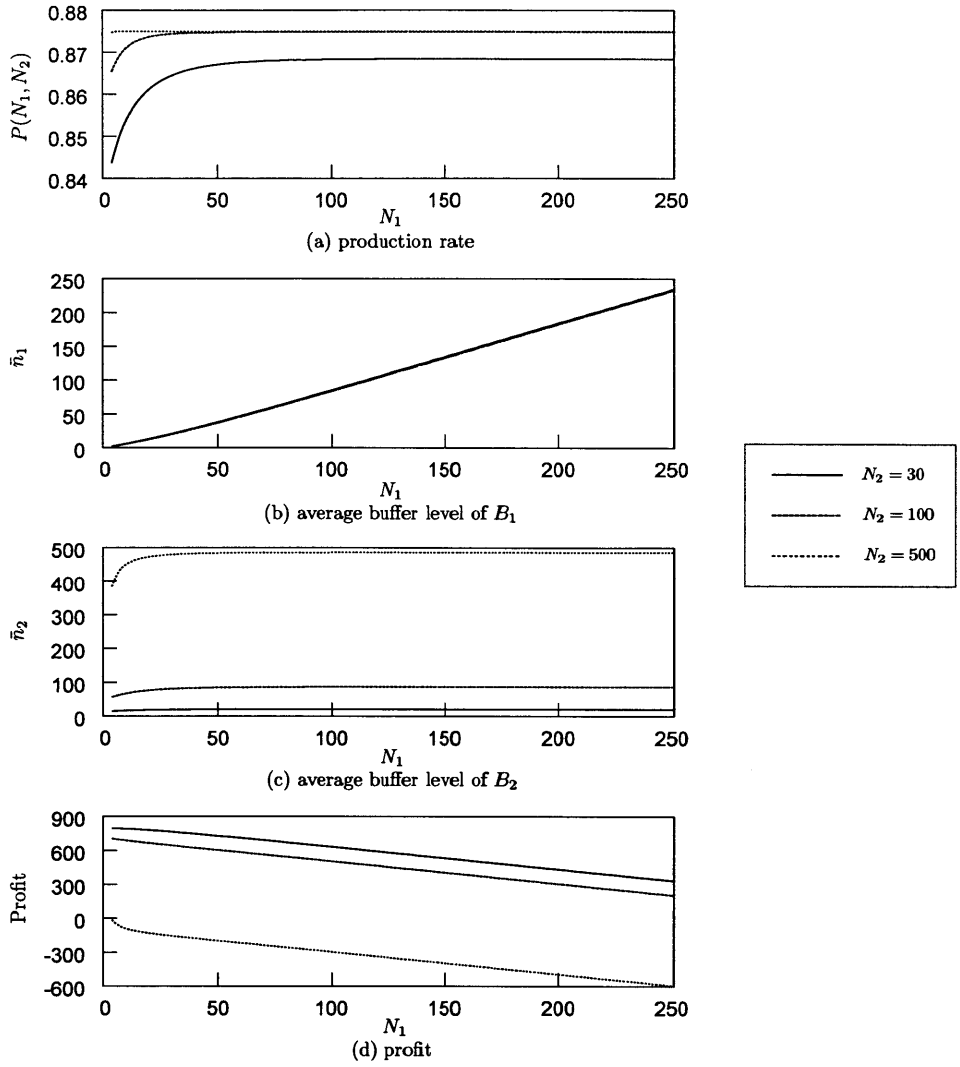


Figure 3-18: Four quantities vs. N_1, Type 5

- Figure 3-18(d) shows the profit $J(N_1)$. When N_2 is large, the profit is monotonically decreasing with N_1, which indicates that Buffer B_1 is undesirable. However, for all three values of N_2, there is a unique optimal value of N_1 that maximizes the profit of the line. The profit appears to be a concave function when N_2 is small while a convex function when N_2 is large. However, we further study $\mathrm{d}^2J/\mathrm{d}N_1^2$. We see that $\mathrm{d}^2J/\mathrm{d}N_1^2 < 0$ when N_2 is small, which is consistent with the observation. However, when N_2 is large, $\mathrm{d}^2J/\mathrm{d}N_1^2$ is positive for some values of N_1 while negative for others (see Figure 3-19). This indicates that

$J(N_1)$ is neither a concave nor a convex function of N_1 when N_2 is large.

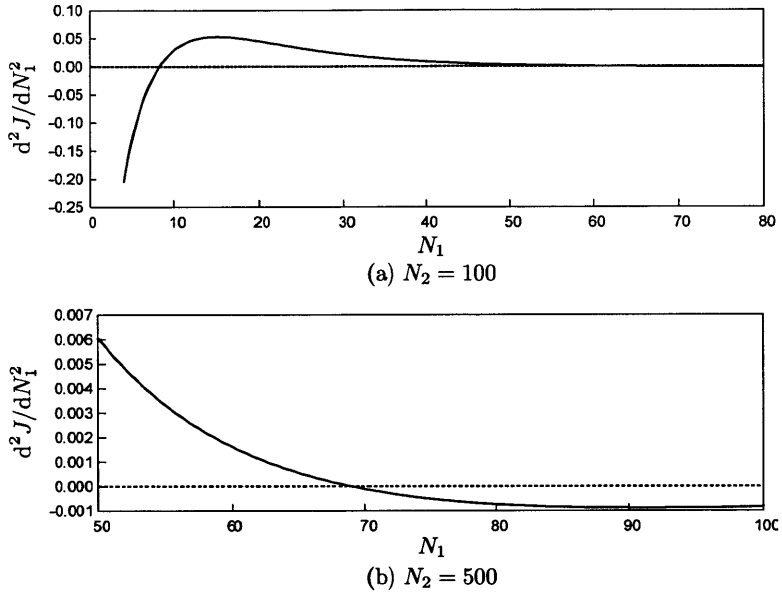


Figure 3-19: d^2J/dN_1^2 vs. N_1, Type 5

Next, we vary N_2 and consider three values of N_1. They are 30, 100, and 500. The four quantities being considered are shown in Figure 3-20. (Compare this with Figure 3-10.)

- Figure 3-20(a) indicates that the production rate appears to be a concave function of N_2. Since $P_3 \leq P_{\{1,2\}}(0)$ $\left(< P_{\{1,2\}}(N_1)\right)$, the production rate is upper bounded by the P_3. Therefore, as N_2 increases, the production rate of the line approaches to $P_3 = .875$.

- Figure 3-20(b) shows $\bar{n}_1(N_2)$. Recall that $P_1 > P_{\{2,3\}}(\infty)$ $\left(> P_{\{2,3\}}(N_2)\right)$. Thus, in the two-machine one-buffer line $M_1 - B_1 - M_{\{2,3\}}$, M_1 is always faster than $M_{\{2,3\}}$ and $\bar{n}_1$ tends to be close to N_1. As N_2 increases, $M_{\{2,3\}}$ becomes faster (though it is still slower than M_1) therefore $\bar{n}_1$ becomes smaller. Consequently, $\bar{n}_1$ decreases a little bit with N_2 (but it is still close to N_1) and finally reaches an asymptote. $\bar{n}_1$ appears to be a convex function of N_2.

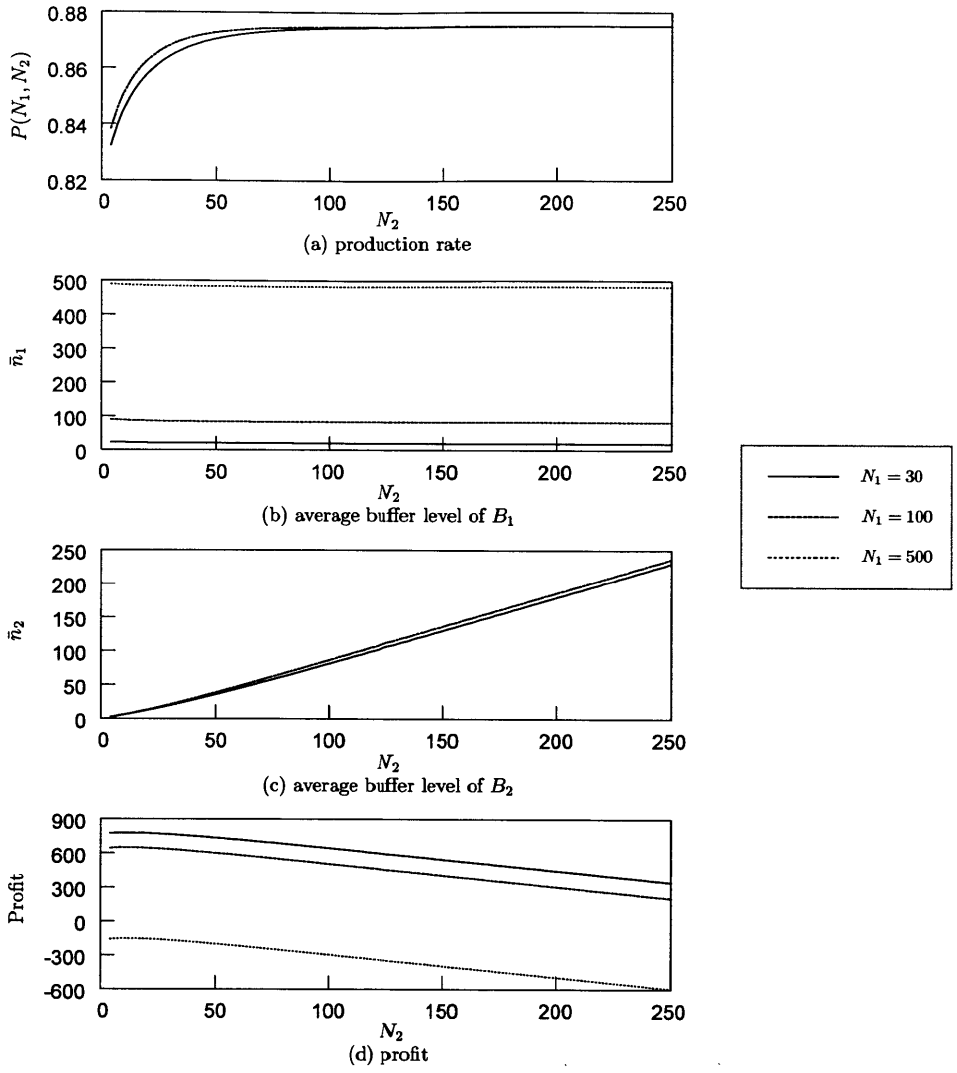


Figure 3-20: Four quantities vs. N_2, Type 5

- Figure 3-20(c) illustrates $\bar{n}_2(N_2)$. Because $P_3 < P_{\{1,2\}}(0)\,(< P_{\{1,2\}}(N_1))$, M_3 is always slower than the upstream $M_{\{1,2\}}$ in $M_{\{1,2\}} - B_2 - M_3$. As we increase N_2, $\bar{n}_2$ increases without a limit. It appears that $\bar{n}_2$ is a convex function of N_2.

- Figure 3-20(d) shows the profit $J(N_2)$. For all three values of N_1, there is a unique optimal value of N_2 that maximizes the profit of the three-machine two-buffer line. The profit appears to be a concave function of N_2. However, after studying $\mathrm{d}^2J/\mathrm{d}N_2^2$, we find that $\mathrm{d}^2J/\mathrm{d}N_2^2 < 0$ when N_1 is large. However, when

N_1 is small (e.g., $N_1 = 30$), $\mathrm{d}^2J/\mathrm{d}N_2^2$ is positive for some values of N_2 while negative for others (see Figure 3-21). It is not clear if this is due to the property of $J(N_2)$ or the decomposition.

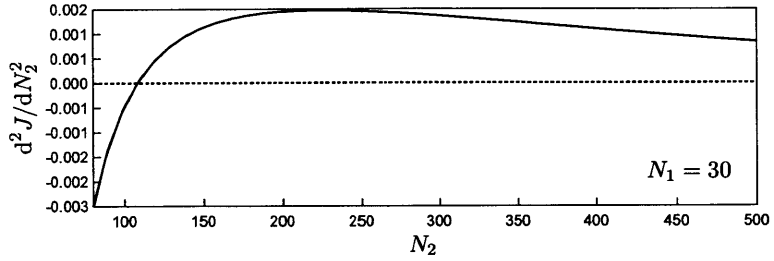


Figure 3-21: $\mathrm{d}^2J/\mathrm{d}N_2^2$ vs. N_2, Type 5

3.4.6 Summary about the Qualitative Behavior of Feasible Cases

In the previous section, the qualitative behavior of the production rate, $\bar{n}_1$, $\bar{n}_2$, and the profit of those five feasible types are studied. The apparent concavity, convexity, non-concavity, or non-convexity properties of these four quantities as functions of N_1 and N_2 in all five feasible types are summarized in Tables 3.9 and 3.10.

Table 3.9: Apparent qualitative behavior of four quantities as functions of N_1

	$P(N_1)$	$\bar{n}_1(N_1)$	$\bar{n}_2(N_1)$	$J(N_1)$
Type 1	concave	convex	concave	concave
Type 2	concave	convex for small N_2; concave for large N_2	concave	concave for small N_2; neither concave nor convex for large N_2
Type 3	concave	concave	concave	concave
Type 4	concave	convex	concave for small N_2; neither concave nor convex for large N_2	concave for small N_2; neither concave nor convex for large N_2
Type 5	concave	convex	concave	concave for small N_2; neither convex nor concave for large N_2

The following conclusions can be made from these observations:

Table 3.10: Apparent qualitative behavior of four quantities as functions of N_2

	$P(N_2)$	$\bar{n}_1(N_2)$	$\bar{n}_2(N_2)$	$J(N_2)$
Type 1	concave	convex	concave	concave
Type 2	concave	convex for small N_1; neither concave nor convex for large N_1	concave for small N_1; neither concave nor convex for large N_1	concave for small N_1; neither concave nor convex for large N_1
Type 3	concave	convex	concave	concave
Type 4	concave	convex	concave for small N_1; convex for large N_1	neither concave nor convex for small N_1; concave for large N_1
Type 5	concave	convex	convex	concave

- For all five feasible types, the production rate always appears to be a concave function of N_1 and N_2.

- The average inventories $\bar{n}_1$ and $\bar{n}_2$ may not be necessary a concave or convex function or N_1 and/or N_2. The shape of the curve depends on the type of the three-machine two-buffer line.

- The profit is not necessary a concave or convex function of N_1 and/or N_2. However, for any type, we can always find a unique global maximum on the profit curve. This implies that a gradient method can be applied to find the global maximum. As we will develop in Chapter 4, in order to solve the profit maximization problem with a production rate constraint, we introduce a corresponding unconstrained problem, for which the gradient method is adopted to solve it.

3.5 Qualitative Behavior of Average Buffer Levels in Longer Lines

In this section, we discuss how understanding three-machine two-buffer lines gives insight into longer lines. In particular, we study a nine-machine eight-buffer line, whose parameters are listed in Table 3.11.

Table 3.11: Parameters of the nine-machine eight-buffer line

machine	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9
r_i	.12	.12	.12	.12	.12	.12	.16	.16	.16
p_i	.01	.01	.01	.01	.01	.01	.01	.01	.01
buffer	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	–
N_i	60	60	60	1000	80	50	20	20	–

First, we study how the average inventory of Buffer B_4 of the nine-machine eight-buffer line changes as the size of B_6 varies. It can be seen that Buffers B_4 and B_6 break the original night-machine eight-buffer line into three segments and therefore it can be viewed as a three-machine two-buffer line (see Figure 3-22). In particular, we view the sub line $M_1 - B_1 - M_2 - B_2 - M_3 - B_3 - M_4$ as Machine $M(1)$ of the three-machine two-buffer line. Similarly, sub lines $M_5 - B_5 - M_6$ and $M_7 - B_7 - M_8 - B_8 - M_9$ are considered as Machines $M(2)$ and $M(3)$ of the three-machine two-buffer line. Note that we use notation $M(i)$ to denote the ith machine of the three-machine two-buffer line to distinguish Machine M_i of the original nine-machine eight-buffer line. Moreover, the production rate of Machine $M(i)$ is denoted by $P(i)$. Similarly, B_4 and B_6 of the original line are considered as Buffers $B(1)$ and $B(2)$ of the three-machine two-buffer line.

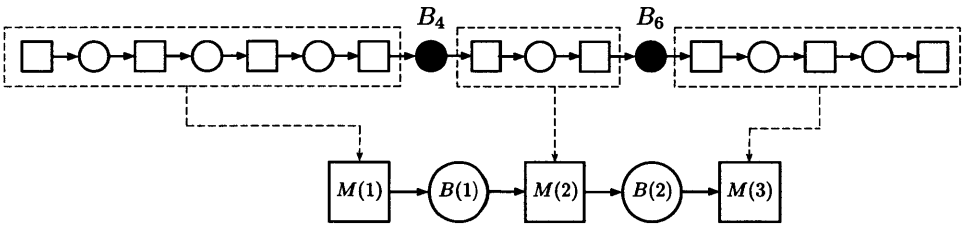


Figure 3-22: Scenario 1 of the nine-machine eight-buffer line

With the decomposition method of Gershwin (1987a), we evaluate the production rates of the three machines. They are $P(1) = .8990$ parts per time unit, $P(2) = .9122$ parts per time unit, and $P(3) = .9097$ parts per time unit. In addition, the production rate of Machines $M(2)$ and $M(3)$ with a zero buffer between them is $P_{(2),(3)}(0) = .8614$ parts per time unit, according to the analytical solution of Buzacott

(1967a). Therefore, it can be seen that these parameters satisfy $P(3) \geq P_{(1),(2)}(\infty)$ and $P_{(2),(3)}(0) < P(1) < P_{(2),(3)}(\infty)$. Thus, the three-machine two-buffer line with respect to Buffers B_4 and B_6 of the original line is a Type 2 line. We vary the size of B_6 while using the values in Table 3.11 for other buffers. The curve for the average buffer level $\bar{n}_4$ as a function of N_6 is illustrated in Figure 3-24(a).

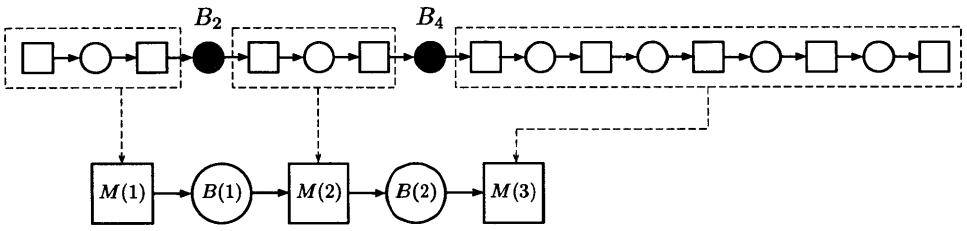


Figure 3-23: Scenario 2 of the nine-machine eight-buffer line

Next, we study how the average inventory of Buffer B_4 of the nine-machine eight-buffer line changes as the size of Buffer B_2 varies. As before, B_2 and B_4 break the original nine-machine eight-buffer line into three segments and therefore it can be viewed as a three-machine two-buffer line (see Figure 3-23). In particular, we view the sub line $M_1 - B_1 - M_2$ as Machine $M(1)$ of the three-machine two-buffer line. Similarly, sub lines $M_3 - B_3 - M_4$ and $M_5 - B_5 - M_6 - B_6 - M_7 - B_7 - M_8 - B_8 - M_9$ are considered as Machines $M(2)$ and $M(3)$ of the three-machine two-buffer line. The production rates of $M(1)$, $M(2)$, and $M(3)$ are $P(1) = .9092$ parts per time unit, $P(2) = .9092$ parts per time unit, and $P(3) = .9014$ parts per time unit, respectively. In addition, the production rate of Machines $M(1)$ and $M(2)$ with a zero buffer between them is $P_{(1),(2)}(0) = .8535$ parts per time unit. Therefore, these parameters satisfy $P(1) \geq P_{(2),(3)}(\infty)$ and $P_{(1),(2)}(0) < P(3) < P_{(1),(2)}(\infty)$. Thus, the three-machine two-buffer line with respect to B_2 and B_4 of the original line is a Type 4 line. We vary the size of B_2 while using the values in Table 3.11 for other buffers. Then the curve for the average buffer level $\bar{n}_4$ as a function of N_2 is illustrated in Figure 3-24(b).

Figure 3-24 shows that $\bar{n}_4$ changes very differently as N_2 and N_6 vary. This example demonstrates that when studying the average inventory level of a buffer as

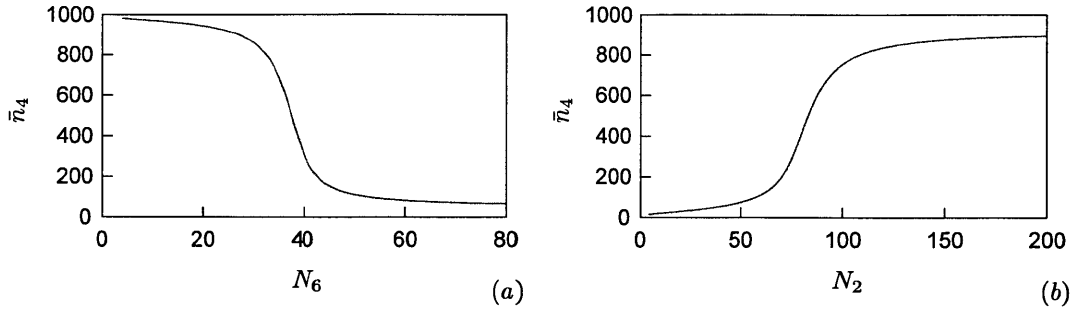


Figure 3-24: $\bar{n}_4$ as a function of N_2 or N_6 in a nine-machine eight-buffer line

a function of the size of another buffer in a long line, we can always divide the long line into three segments and view it as a three-machine two-buffer line with respect to those two buffers. More importantly, for a given buffer in the long line, it is possible that its average buffer level exhibits totally different behavior as we change the sizes of different buffers, because each resulting three-machine two-buffer line falls into a specific type of those five feasible types discussed in Section 3.4 and different types exhibit distinct qualitative behaviors. Therefore, understanding three-machine two-buffer lines provides us insight into longer lines. In the following section, we further study the profit as a function of both N_1 and N_2 for each feasible type.

3.6 Profit Analysis of Five Feasible Types

In the previous section, we study the profit of the line as functions of N_1 and N_2 individually. In this section, we study the profit again but as a function of both N_1 and N_2. Three dimensional graphs (with profit on the vertical axis, while N_1 and N_2 on the horizontal axes) will be provided to demonstrate the qualitative behavior of the profit. The profit ($ per time unit) of the line is computed by

$$J(N_1, N_2) = 1000P(N_1, N_2) - N_1 - N_2 - \bar{n}_1 - \bar{n}_2.$$

3.6.1 Type 1

The line with parameters listed in Table 3.4 is considered. The result is shown in Figure 3-25. In the figure, the profit surface $J(N_1, N_2)$ is provided. In addition, the *iso-profit* lines are projected on the $N_1 - N_2$ plane. It appears that the profit surface is concave in N_1 and N_2. However, the inner bending iso-profit contour of $J(N_1, N_2) = 400$ indicates minor local non-concavity of the surface. However, it is not clear if this is due to the properties of $J(N_1, N_2)$ or the decomposition method. The red cross indicates the unique global optimal solution that maximizes the profit of the line (without a production rate constraint).

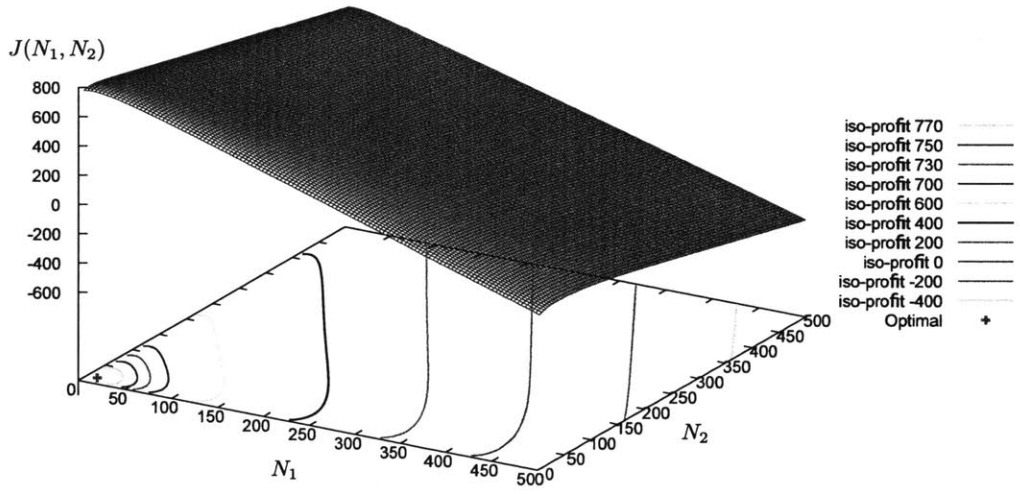


Figure 3-25: Profit vs. N_1 and N_2, Type 1

3.6.2 Type 2

For Type 2, we consider the line whose parameters are listed in Table 3.5. The result is illustrated in Figure 3-26. From the shapes of the iso-profit lines as well as the profit surface, it is clear that the profit $J(N_1, N_2)$ is neither a concave nor a convex function of N_1 and N_2. However, there is a unique global optimal solution (i.e., the red cross) that maximizes the profit of this Type 2 line.

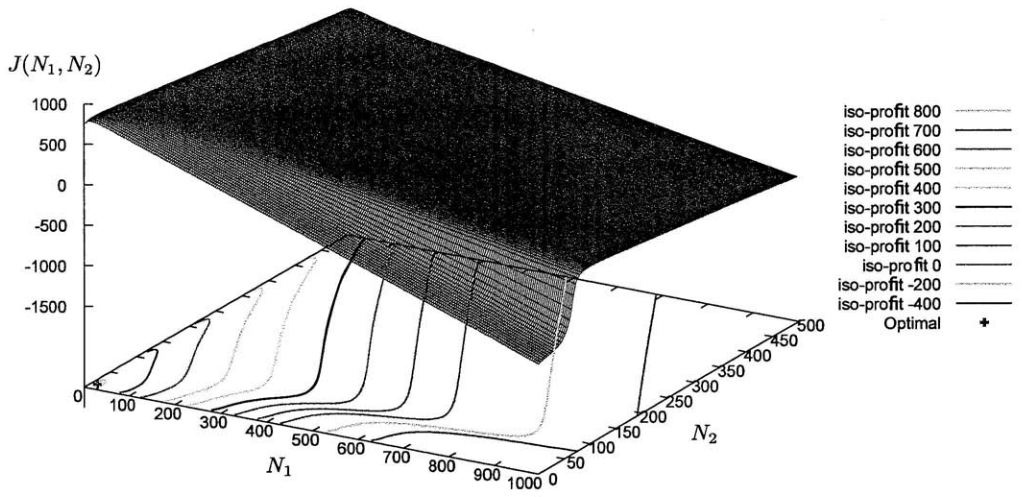


Figure 3-26: Profit vs. N_1 and N_2, Type 2

3.6.3 Type 3

Next, we study the line with parameters provided in Table 3.6 for Type 3. The result is shown in Figure 3-27. From the shapes of the iso-profit lines as well as the profit surface, it appears that the profit is a concave function of N_1 and N_2. This is consistent with the observation for Type 3 summarized in Tables 3.9 and 3.10. The red cross indicates the unique global optimal solution that maximizes the profit of the line.

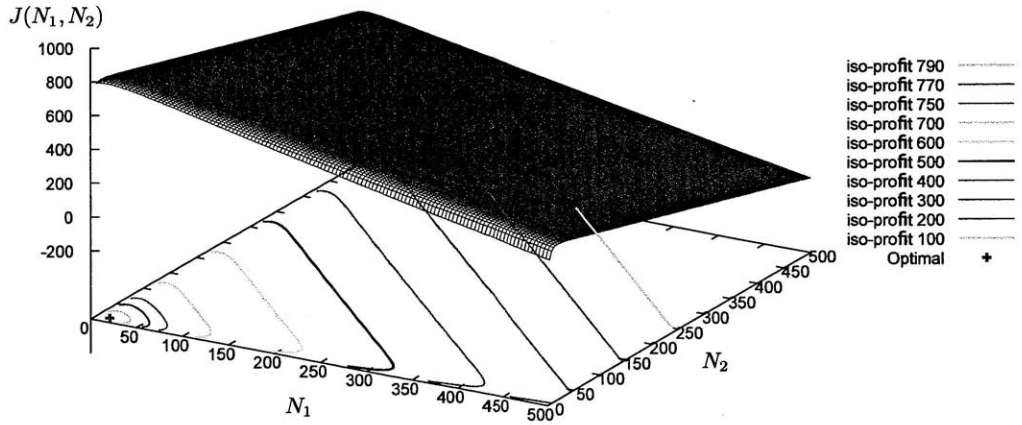


Figure 3-27: Profit vs. N_1 and N_2, Type 3

3.6.4 Type 4

Next, we study the line with parameters listed in Table 3.7 for Type 4. The result is demonstrated in Figure 3-28. From the shapes of the iso-profit lines as well as the profit surface, it is clear that the profit $J(N_1, N_2)$ is neither a concave nor a convex function of N_1 and N_2 in Type 4. However, there is a unique global optimal solution (i.e., the red cross) that maximizes the profit of this Type 4 line.

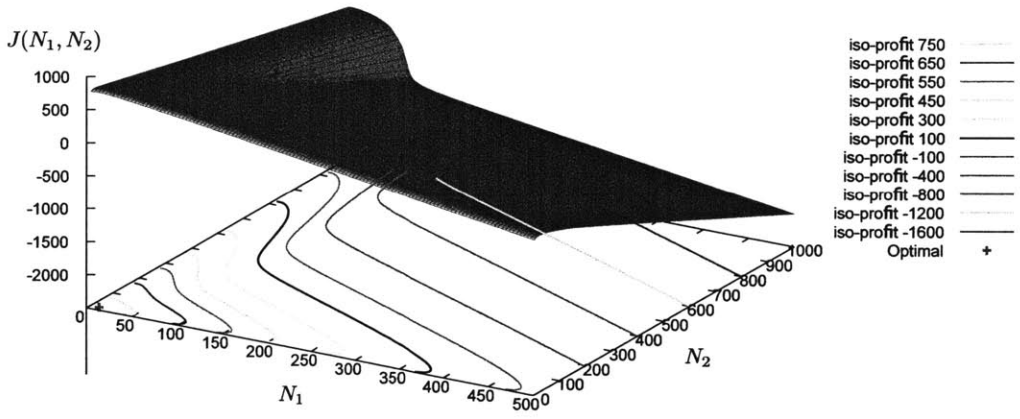


Figure 3-28: Profit vs. N_1 and N_2, Type 4

3.6.5 Type 5

Finally, for Type 5 we study the line of Table 3.8. The result is shown in Figure 3-29. From the shapes of the iso-profit lines as well as the profit surface, it is clear that the profit $J(N_1, N_2)$ is neither a concave nor a convex function of N_1 and N_2 in Type 5. However, there is a unique global optimal solution that maximizes the profit of this Type 5 line.

3.6.6 Summary about the Profit $J(N_1, N_2)$

We make two observations about the profit of the line as a function of both N_1 and N_2 from the results above:

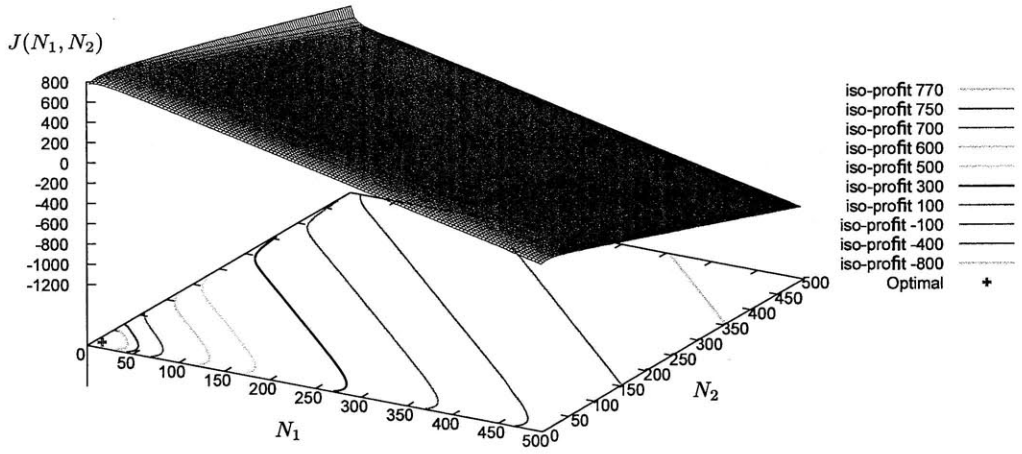


Figure 3-29: Profit vs. N_1 and N_2, Type 5

1. The profit of the line is not necessarily a concave or convex function of buffer spaces in all five feasible types.

2. In each of these five types, the profit has a single global maximum (and no other local maxima).

As we will indicate in Chapter 4, to solve the profit maximization problem subject to the production rate constraint, we introduce a corresponding unconstrained problem without the production rate constraint. We are going to adopt a gradient method of nonlinear programming (Bertsekas 1999) to solve the unconstrained problem. For the gradient method to solve the unconstrained problem correctly, it requires the profit function (i.e., $J(N_1, N_2)$ here, or $J(\mathbf{N})$ for longer lines) being searched to have a single global maximum. The results above on these five types satisfy this requirement, as in each of these five types, no matter whether the profit is a concave/convex function of buffer spaces or not, it appears that the profit has a single global maximum.

To further investigate this point, we study 5000 randomly generated three-machine two-buffer lines and study their profits. Both machine parameters and buffer cost coefficients are randomly chosen. For each of the five types, we study 1000 cases. In each case, we search in the (N_1, N_2) space and count the number of local profit

maxima. In all these 5000 lines studied, there is only a single global profit maximum for each line[2]. In addition, Schor (1995) encounters the same issue and shows that $J(\mathbf{N})$ has a single maximum through numerical evidence and an intuitive argument for both two-machine lines and long lines. Schor (1995) adopts a two-step gradient method to solve the profit maximization problem without a production rate constraint (i.e., our unconstrained problem), and the accuracy of his algorithm based on the two-step gradient method is verified by comparing with an exhaustive search method. Gershwin and Schor (2000) confirms this point. The numerical evidence and the argument of Schor (1995) indicate that assumption that $J(\mathbf{N})$ has a single maximum is reasonable.

Before we conclude this section, we want to mention that the optimal buffer distribution that maximizes the profit of the line is bounded. To argue this, we analyze how each of the three components in the profit expression (1.2) (i.e., the revenue that is associated with the production rate, the buffer space cost, and the average inventory holding cost) changes as buffer sizes go to ∞.

- Due to the concavity and monotonicity of $P(\mathbf{N})$, the production rate will increase up to an asymptote as $N_i, i = 1, \cdots, k-1$ go to ∞. As a result, the revenue $AP(\mathbf{N})$ is bounded and approaches to an asymptote as well.

- The average inventory level of a given buffer depends on the relative speeds of the sub-lines upstream and downstream of the buffer. If the upstream is faster than the downstream, then the average inventory level will not be bounded, otherwise it will approach to an asymptote. Therefore, the average inventory holding cost of each buffer can be either unlimited or bounded, as $N_i, i = 1, \cdots, k-1$ go to ∞.

- The buffer space cost is a linear function of buffer sizes. Therefore, it is unbounded as $N_i, i = 1, \cdots, k-1$ go to ∞.

According to the analysis above, we know that the positive term in the profit expression (i.e., the revenue associated with the production rate) will be bounded.

[2]See Appendix G for details.

However, at least one of the two negative cost terms is unbounded. Therefore, we conclude that the profit goes to $-\infty$ when $N_i, i = 1, \cdots, k-1$ go to ∞. Therefore, the optimal buffer distribution that maximizes the profit of the line is bounded.

3.7 Summary

In this chapter, we study the qualitative behavior of the average inventory levels of three-machine two-buffer lines in a systematic manner and extend the scope of the research to the profit of such lines. A given three-machine two-buffer line can be considered as two two-machine one-buffer building blocks with respect to Buffers B_1 and B_2, respectively. For each building block, there are three possible cases in terms of the relative speeds of the upstream machine and the downstream machine. Therefore, there are nine possible types for a three-machine two-buffer line, which are determined by machine parameters. However, as we have shown, only five out of the nine types are feasible, while the other four types are infeasible.

For each feasible type, the following four quantities of the line are studied as functions of N_1 and N_2 individually: the production rate, the average inventory of B_1, the average inventory of B_2, and the profit. A set of important observations about these quantities is drawn from these results. The methodology is then extended to study how the average inventory level of a buffer changes as the size of another buffer varies in a longer line. It is illustrated that understanding three-machine two-buffer lines gives insight into longer lines.

Finally, we study the profit of three-machine two-buffer lines as a function of both N_1 and N_2. For each feasible type, no matter whether the profit of the line is a concave/convex function of buffer sizes or not, the profit appears to have a single global maximum. Further numerical evidence and literature review indicate that assumption that $J(\mathbf{N})$ has a single maximum is reasonable. Therefore, as we will indicate in Chapter 4, a gradient method is appropriate to solve the unconstrained profit maximization problem without the production rate constraint.

Chapter 4

Production Line Profit Maximization

In this chapter, we develop an efficient algorithm for production line profit maximization through buffer size optimization. We consider both buffer space cost and average inventory cost, and we include a nonlinear production rate constraint. To solve the problem, a corresponding unconstrained problem is introduced and a nonlinear programming approach is adopted. The material presented in this chapter is an extension of Shi and Gershwin (2009a).

In particular, we develop the algorithm with the deterministic processing time model of Gershwin (1987a), (1994). However, the algorithm can be applied to the other two production line models (i.e., the deterministic multiple failure mode model of Tolio and Matta 1998 and the continuous multiple failure mode model of Levantesi et al. 2003) as well. Some numerical results about the algorithm on the other two models are included in Shi and Gershwin (2009b). We provide more experiments about this in Section 4.4.

As indicated in Chapter 1, production line profit maximization is one of the three major topics of this thesis. The algorithm presented in this chapter will be extended to single closed-loop systems in Chapter 6 and to lines with an additional maximum part waiting time constraint in Chapter 7. Some valuable insights about optimal design of long lines are discussed in Chapters 8 and 9 as well.

The rest of this chapter is organized as follows. We present the model of the line in Section 4.1. The algorithm is then developed in Section 4.2. Numerical results and analysis are provided to show the accuracy and efficiency of the algorithm in Section 4.3, following by more numerical results about the algorithm on the other line models in Section 4.4.

4.1 Problem Statement, Assumptions, and Notation

4.1.1 Model of the Line

The model described here is the deterministic processing time model of Gershwin (1987a), (1994). We make all the assumptions and approximations of that model, follow all his conventions, and use his notation. We outline the key features of the model below.

In the model, we denote the ith machine by M_i and the ith buffer by B_i. The line consisting of k machines and $k-1$ buffers is called a k-machine, $k-1$-buffer line, or k-machine line for short. Processing times of all machines are equal, deterministic, and constant. Time is scaled so that operations take one time unit. We further assume that all the machines start their operations at the same instant. Transportation time is negligible compared to the operation time.

In addition, N_i, the size of Buffer $B_i, \forall i = 1, \cdots, k-1$, are decision variables. Therefore, there are $k-1$ decision variables for a k-machine, $k-1$-buffer line. Machines are unreliable and are parameterized by probabilities of failure and repair. Specifically, the parameters of Machine M_i are p_i, the probability of a failure during a time unit while the machine is operating; and r_i, the probability of a repair during a time unit while the machine is down. As a consequence, the times to failure and to repair are geometrically distributed. By convention, repairs and failures occur at the beginnings of time units and changes in the buffer levels take place at the ends of time units. Machine parameters are fixed.

We define P to be the production rate of a line. Although the production rate P is a function of machines and their reliability, we vary only buffer sizes, so we write $P = P(N_1, \cdots, N_{k-1})$, or $P(\mathbf{N})$ for short, where $\mathbf{N}$ is the vector $(N_1, \cdots, N_{k-1})$. $P(\mathbf{N})$ is a nonlinear function of buffer sizes $\mathbf{N}$, and is calculated numerically by decomposition (Gershwin 1987a) for lines having more than two machines.

We have defined the profit of the line in Equations (1.1) and (1.2) in Chapter 1. As a reminder, the profit of a k-machine, $k-1$-buffer line is formulated as

$$\text{Profit} = AP(N_1, \cdots, N_{k-1}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i - Z,$$

where $A > 0$ (\$/part) is the revenue coefficient associated with the production rate $P(\mathbf{N})$, b_i and c_i (\$/part/time unit) are cost coefficients associated with the buffer space and average inventory for the ith buffer, respectively, and Z stands for all costs other than those due to buffer sizes, average inventory, and raw material. Since Z is independent of $\mathbf{N}$, we simplify the formulation above and write our objective function as

$$J(N_1, \cdots, N_{k-1}) = AP(N_1, \cdots, N_{k-1}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i, \tag{4.1}$$

where we refer to $J(N_1, \cdots, N_{k-1})$ as the profit of the line. To simplify terminology, the first item on the right side of Equation (4.1) can be seen as the total revenue of the line; while the other two items together can be interpreted as the total cost of the line. Allowing different buffers to have different cost coefficients is realistic as we know that, for example, the cost of buffer space in a clean room is much expensive than elsewhere in a factory.

4.1.2 Problem Formulations

In this section, we introduce mathematical models. Our prime goal is a constrained problem, in which we aim at maximizing profits of production lines subject to a production rate constraint. In order to solve the constrained problem, we present a corresponding unconstrained problem, in which we drop the production rate con-

straint. We introduce the two problems here and leave the reason for introducing the unconstrained problem to Section 4.2.

The constrained problem

The constrained problem is formulated as follows:

$$\begin{aligned} \max_{\mathbf{N}} \quad & J(N_1, \cdots, N_{k-1}) = AP(N_1, \cdots, N_{k-1}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i \\ \text{subject to} \quad & P(N_1, \cdots, N_{k-1}) \geq \hat{P}, \\ & N_i \geq 0, \forall i = 1, \cdots, k-1, \end{aligned} \tag{4.2}$$

where $\hat{P}$ is the target demand rate or the required production rate. The first constraint is the production rate constraint. Note that it is nonlinear (and see more properties of $P(\mathbf{N})$ discussed in Chapter 2). The second constraint is called the buffer size constraint. It comes from the natural property of buffer sizes since it is not possible to have negative buffer sizes.

However, it is necessary for us to further limit the buffer sizes. This is because we use the decomposition to evaluate the production rate of the line, and the decomposition is based on an analytical solution of the two-machine line (Gershwin 1994). Therefore we follow the model convention and let $N_i \geq 4, \forall i$. For a line having buffer sizes less than 4, there are different ways to measure its performance and we do not discuss them in this chapter. Therefore, we only focus on lines whose buffer sizes are all ≥ 4. In the following, let $N_{\min}$ denote the minimum of the buffer size and we re-write the constrained problem as:

$$\begin{aligned} \max_{\mathbf{N}} \quad & J(N_1, \cdots, N_{k-1}) = AP(N_1, \cdots, N_{k-1}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i \\ \text{subject to} \quad & P(N_1, \cdots, N_{k-1}) \geq \hat{P}, \\ & N_i \geq N_{\min}, \forall i = 1, \cdots, k-1. \end{aligned} \tag{4.3}$$

The unconstrained problem

In the unconstrained problem, we drop the production rate constraint. Thus, the unconstrained problem is

$$\begin{aligned} \max_{\mathbf{N}} \quad & J(N_1, \cdots, N_{k-1}) = AP(N_1, \cdots, N_{k-1}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i \\ \text{subject to} \quad & N_i \geq N_{\min}, \forall i = 1, \cdots, k-1. \end{aligned} \tag{4.4}$$

This is a convenient, although not quite accurate, name since we still have the buffer size constraint. As we show in Section 4.2, the unconstrained problem can be solved easily by a gradient method. We will further illustrate the relationship between the two problems and reveal how we take advantage of the unconstrained problem to solve the constrained one.

4.2 Solution Technique

In this section, we present the algorithm for solving the constrained problem (4.3). We realize that in (4.3), both the objective function and the production rate constraint are nonlinear. Therefore, it is difficult to solve (4.3) directly. As a result, instead of solving it directly, we adopt a two-step strategy in which we introduce a new variable A'. We replace A by A' in the unconstrained problem (4.4) and solve it iteratively for different values of A'.

To solve the unconstrained problem (4.4), we take advantage of the analytical form of the two-machine-line evaluation, which enables us to treat N_i as continuous variables. (A discussion about this appears in Section 2.1 and we provide the continuous variable version of the solution of the two-machine line in Appendix A). Consequently, though our model is a deterministic processing time discrete state model, we can still treat N_i as continuous variables. In addition, since we evaluate $P(\mathbf{N})$ by the decomposition and the analytic two-machine-line evaluation, and treat N_i as continuous variables, we are able to treat $P(\mathbf{N})$ and $J(\mathbf{N})$ as continuously differentiable functions[1]. Therefore, we adopt a gradient method to solve the unconstrained problem. However, we need to indicate that due to the lack of an analytical expression of the profit of a line having more than two machines, we compute gradients according to a forward difference formula (see Section 4.2.4). Also see Levantesi et al. (2001) for analytical work of the derivatives.

It is important to point out that gradient methods are appropriate when the space being searched has a single maximum. This requires our objective function of the unconstrained problem, $J(N_1, \cdots, N_{k-1})$, to have only one maximum point for the proposed optimization method to work correctly. Schor (1995) encounters the same issue and shows that $J(N_1, \cdots, N_{k-1})$ has single maximum through numerical evidence and an intuitive argument. Schor (1995) introduces a two-step gradient method to solve our unconstrained problem. His results are demonstrated to be correct when compared with both exhaustive search, simulation and the optimization method of (Seong et al. 1994). Gershwin and Schor (2000) confirm this. Moreover, the discussion in Chapter 3 provides numerical evidence to show that no matter whether the profit of the line is a concave/convex function of buffer sizes or not, the profit appears to have a single global maximum. The numerical evidence and the literature review indicate that the profit of production lines has a single global maximum is a reasonable assumption.

[1]See Section 2.1 for the discussion about the continuity of $P(\mathbf{N})$ and $J(\mathbf{N})$.

4.2.1 Algorithm Derivation

We solve the unconstrained problem (4.4) and let $(N_1^u, \cdots, N_{k-1}^u)$ be its solution. This yields two cases:

1. The solution $(N_1^u, \cdots, N_{k-1}^u)$ satisfies $P(N_1^u, \cdots, N_{k-1}^u) \geq \hat{P}$. In this case, since the production rate constraint is satisfied, the solution of the constrained problem (4.3) is $(N_1^\star, \cdots, N_{k-1}^\star) = (N_1^u, \cdots, N_{k-1}^u)$.

2. The solution $(N_1^u, \cdots, N_{k-1}^u)$ satisfies $P(N_1^u, \cdots, N_{k-1}^u) < \hat{P}$. Therefore, it is not the solution of the constrained problem. In this case, we replace the A of (4.4) by A' and consider the following unconstrained problem:

$$\begin{aligned} \max_{\mathbf{N}} \quad & J(N_1, \cdots, N_{k-1}) = A'P(N_1, \cdots, N_{k-1}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i \\ \text{subject to} \quad & N_i \geq N_{\min}, \forall i = 1, \cdots, k-1, \end{aligned} \tag{4.5}$$

   Let $(N_1', \cdots, N_{k-1}')$ be the solution to this problem and $P' = P(N_1', \cdots, N_{k-1}')$. Then, we claim the following.

Assertion The constrained problem

$$\begin{aligned} \max_{\mathbf{N}} \quad & J(N_1, \cdots, N_{k-1}) = A'P(N_1, \cdots, N_{k-1}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i \\ \text{subject to} \quad & P(N_1, \cdots, N_{k-1}) \geq \hat{P}, \\ & N_i \geq N_{\min}, \forall i = 1, \cdots, k-1 \end{aligned} \tag{4.6}$$

has the same solution for all A' in which the solution of the unconstrained problem (4.5) has $P' \leq \hat{P}$.

This is because the solution of problem (4.6) will satisfy $P(N_1, \cdots, N_{k-1}) = \hat{P}$. Therefore, the objective function is equivalent to $A'\hat{P} - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i$. Since

the first term $A'\hat{P}$ is independent of all of the N_i, it has no effect on the solution of the problem.

We claim in the assertion that if the optimal solution of the unconstrained problem (4.4) is not the solution of the constrained problem (4.3), then the solution of the constrained problem (4.3), $(N_1^\star, \cdots, N_{k-1}^\star)$, satisfies $P(N_1^\star, \cdots, N_{k-1}^\star) = \hat{P}$. Therefore, to solve the constrained problem (4.3), we replace A by A' in (4.4) and solve problem (4.5) for different A's. We need to find the value of A' such that the solution to problem (4.5) satisfies $P(N_1', \cdots, N_{k-1}') = \hat{P}$. Then, this solution is the same as that of the original constrained problem (4.3).

We provide an illustration of this assertion in Figure 4-1. Consider a three-machine two-buffer line whose parameters are $r_1 = .1$, $p_1 = .01$, $r_2 = .11$, $p_2 = .01$, $r_3 = .1$, and $p_3 = .009$. The coefficients are $b_i = c_i = 1, \forall i$. The revenue coefficient is $A = 1500$. The target production rate is $\hat{P} = .88$. The profit of the line as a function of (N_1, N_2) is drawn in Figure 4-1(a). In Figure 4-1(a), the blue region on the profit surface is the feasible region for the problem under the production rate constraint and the blue region is also projected on the $N_1 - N_2$ plane. The red dot $(N_1 = 39.68, N_2 = 42.36)$[2] indicates the point that maximizes the profit of the line while satisfying the production rate constraint. Note that it is on the boundary of the blue region, which means that the production rate constraint is satisfied with equality. The black dot $(N_1 = 17.57, N_2 = 20.44)$ is the optimal solution of the corresponding unconstrained problem. As it is not within the blue region, it does not satisfy the production rate constraint. Iso-profit contours are also provided in the figure.

Next, we replace $A = 1500$ by $A' = 2500$ for the line. The result is shown in Figure 4-1(b). It can be seen that the optimal solution of the unconstrained problem with $A' = 2500$ (i.e., the black dot inside the 2085 iso-profit contour) still does not satisfy the production rate constraint, as it is outside the blue region. More importantly, we see the that optimal solution of the constrained problem with $A' = 2500$ (i.e., the red dot in Figure 4-1(b)) is exactly the same as the optimal solution of the

[2]For demonstration purpose, we keep buffer sizes to be non-integer as we have argued that buffer sizes can be treated as continuous variables.

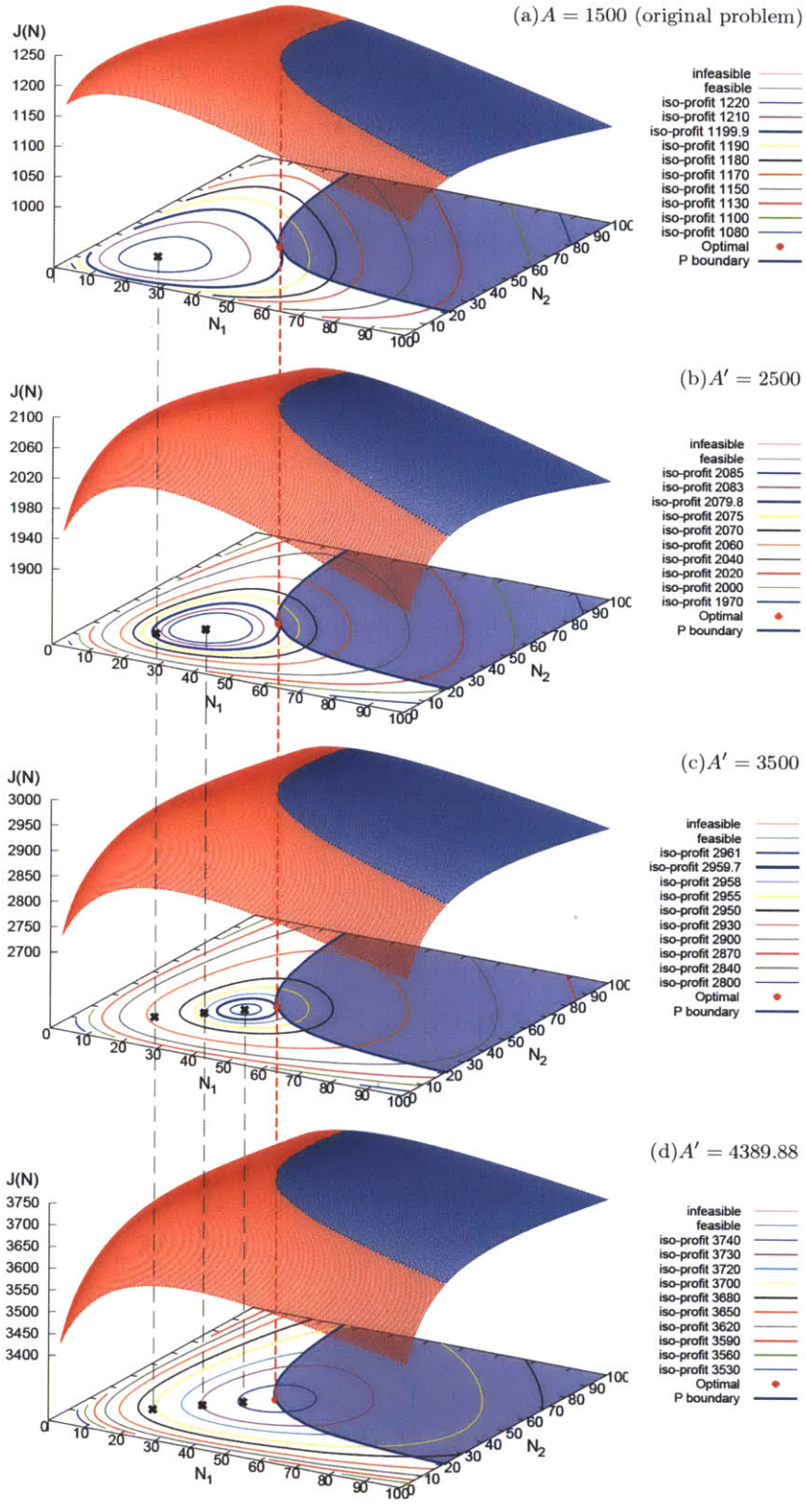


Figure 4-1: An example of the assertion

original problem where $A = 1500$. If we further increase A' to 3500, we will make the same observation (see Figure 4-1(c)). Finally, we find a certain value of A' (4389.88 in this example) such that the optimal solution of the unconstrained problem with $A' = 4389.88$ satisfies the production rate constraint and therefore it is also the optimal solution of the constrained problem when $A' = 4389.88$ (the red dot in Figure 4-1(d)). Figures 4-1(a), (b), (c), and (d) demonstrate that the optimal solution when $A' = 4389.88$ is indeed the solution of the other three constrained problems where $A' = A = 1500$ (original problem), $A' = 2500$, and $A' = 3500$, respectively. This illustrates the assertion that the constrained problem (4.6) has the same solution for all A' in which the solution of the unconstrained problem (4.5) has $P' \leq \hat{P}$. Therefore, as long as we find the value of A' such that the solution to problem (4.5) satisfies $P(N'_1, \cdots, N'_{k-1}) = \hat{P}$, then this solution is the same as that of the original constrained problem (4.3). In this example, $A' = 4389.88$ and the optimal solution of the unconstrained problem with this A' is $N'_1 = 39.68$ and $N'_2 = 42.36$, and it satisfies $P(N'_1, N'_2) = \hat{P}$. Therefore, it is also the optimal solution of the original constrained problem where $A = 1500$.

Proof of Assertion

Here, we formally prove this assertion by the Karush-Kuhn-Tucker (KKT) conditions of nonlinear programming (Bertsekas 1999). We first convert the constrained problem (4.3) into the minimization form:

$$\begin{aligned} \min_{\mathbf{N}} \quad & -J(N_1, \cdots, N_{k-1}) = -AP(N_1, \cdots, N_{k-1}) + \sum_{i=1}^{k-1} b_i N_i + \sum_{i=1}^{k-1} c_i \bar{n}_i \\ \text{subject to} \quad & \hat{P} - P(N_1, \cdots, N_{k-1}) \leq 0, \\ & N_{\min} - N_i \leq 0, \forall i = 1, \cdots, k-1. \end{aligned} \tag{4.7}$$

We have argued that we can treat N_i as continuous variables, and $P(\mathbf{N})$ and $J(\mathbf{N})$ as continuously differentiable functions. Let us consider the KKT conditions. A statement of the KKT conditions (Bertsekas 1999) is: Let $x^\star$ be a local minimum

of the problem

$$\min \quad f(x)$$

$$\text{subject to} \quad h_1(x) = 0, \cdots, h_m(x) = 0, \tag{4.8}$$

$$g_1(x) \leq 0, \cdots, g_r(x) \leq 0,$$

where f, h_i, and g_j are continuously differentiable functions from $\Re^n$ to $\Re$. Assume that $x^\star$ is regular[3]. Then there exist unique Lagrange multipliers $\lambda_1^\star, \cdots, \lambda_m^\star$ and $\mu_1^\star, \cdots, \mu_r^\star$, satisfying the following conditions:

$$\nabla_x L(x^\star, \lambda^\star, \mu^\star) = 0,$$

$$\mu_j^\star \geq 0, j = 1, \cdots, r, \tag{4.9}$$

$$\mu_j^\star g_j(x^\star) = 0, j = 1, \cdots, r.$$

where $L(x, \lambda, \mu) = f(x) + \sum_{i=1}^m \lambda_i h_i(x) + \sum_{j=1}^r \mu_j g_j(x)$ is called the Lagrangian function.

Before we apply the KKT conditions to our problem, we need to point out a necessary condition that guarantees the existence of Lagrange multipliers. One appropriate for our problem is the Slater constraint qualification for convex inequalities (Bertsekas 1999), which is: Let $x^\star$ be a local minimum of the problem (4.8), where f and g_j are continuously differentiable functions from $\Re^n$ to $\Re$, and the functions h_i are linear. Assume that the functions g_j are convex and that there exists a feasible vector $\bar{x}$ satisfying $g_j(\bar{x}) < 0, \forall j \in M(x^\star)$. Then $x^\star$ satisfies the KKT conditions.

[3]Let $M(x)$ be the set of active inequality constraints, i.e., $M(x) = \{j | g_j(x) = 0\}$. A feasible vector x is regular if the equality constraint gradients $\nabla h_i(x), i = 1, \cdots, m$, and the active inequality constraint gradients $\nabla g_j(x), j \in M(x)$, are linearly independent. Also x is regular in the exceptional case where there are no equality constraints and all the inequality constraints are inactive at x (Bertsekas 1999).

Let us now consider our constrained problem (4.7). There are no equality constraints in the problem, but there are k inequality constraints:

$$\begin{aligned} g_0(\mathbf{N}) &= \hat{P} - P(N_1, \cdots, N_{k-1}) \leq 0, \\ g_i(\mathbf{N}) &= N_{\min} - N_i \leq 0, \forall i = 1, \cdots, k-1. \end{aligned} \tag{4.10}$$

Due to the concavity of $P(\mathbf{N})$, $g_0(\mathbf{N})$ is a convex function. All other $g_i(\mathbf{N})$ are linear so that they are also convex. It is not hard to find a feasible vector to make our problem satisfy the Slater constraint qualification. Since the required production rate, $\hat{P}$, has to be feasible for the line, there exists sufficiently large $\hat{\mathbf{N}}$ such that $P(\hat{N}_1, \cdots, \hat{N}_{k-1}) > \hat{P}$ so $g_0(\hat{N}_1, \cdots, \hat{N}_{k-1}) < 0$. In addition, $g_i(\hat{N}_1, \cdots, \hat{N}_{k-1}) < 0, \forall i = 1, \cdots, k-1$ because $N_{\min} - \hat{N}_i < 0, \forall i = 1, \cdots, k-1$. Hence, our constrained problem satisfies the Slater constraint qualification[4], and there exist unique Lagrange multipliers $\mu_i^\star, i = 0, \cdots, k-1$ for (4.7) to satisfy the KKT conditions:

$$-\nabla J(\mathbf{N}^\star) + \mu_0^\star \nabla \left(\hat{P} - P(\mathbf{N}^\star) \right) + \sum_{i=1}^{k-1} \mu_i^\star \nabla \left(N_{\min} - N_i^\star \right) = 0 \tag{4.11}$$

or

$$- \begin{pmatrix} \frac{\partial J(\mathbf{N}^\star)}{\partial N_1} \\ \frac{\partial J(\mathbf{N}^\star)}{\partial N_2} \\ \vdots \\ \frac{\partial J(\mathbf{N}^\star)}{\partial N_{k-1}} \end{pmatrix} - \mu_0^\star \begin{pmatrix} \frac{\partial P(\mathbf{N}^\star)}{\partial N_1} \\ \frac{\partial P(\mathbf{N}^\star)}{\partial N_2} \\ \vdots \\ \frac{\partial P(\mathbf{N}^\star)}{\partial N_{k-1}} \end{pmatrix} - \mu_1^\star \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} - \mu_2^\star \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} - \cdots - \mu_{k-1}^\star \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{4.12}$$

[4]For some discussion about the constraint qualifications for the case of inequality constraints only, see Exercise 3.35 of Bertsekas (1999).

and

$$\mu_i^\star \geq 0, \forall i = 0, \cdots, k-1, \tag{4.13}$$

$$\mu_0^\star \left(\hat{P} - P(\mathbf{N}^\star) \right) = 0, \tag{4.14}$$

$$\mu_i^\star (N_{\min} - N_i^\star) = 0, \forall i = 1, \cdots, k-1, \tag{4.15}$$

where $\mathbf{N}^\star$ is the optimal solution of our constrained problem.

Next, we show that finding the Lagrange multipliers $\mu_i^\star, i = 0, \cdots, k-1$ and the optimal solution $\mathbf{N}^\star$ to satisfy the KKT conditions (4.12) to (4.15) is equivalent to solving the constrained problem (4.3) by our algorithm. Suppose that $\mathbf{N}^\star$ is an interior solution[5]. (In most of our experiments, the optimal solutions have this feature, but we provide a set of special cases where some $N_i^\star = N_{\min}$ in Section 4.3.1.) In the interior solution case, by condition (4.15), we know that $\mu_i^\star = 0, \forall i = 1, \cdots, k-1$. Hence, we can simplify the KKT conditions (4.12) to (4.15) to

$$-\begin{pmatrix} \frac{\partial J(\mathbf{N}^\star)}{\partial N_1} \\ \frac{\partial J(\mathbf{N}^\star)}{\partial N_2} \\ \vdots \\ \frac{\partial J(\mathbf{N}^\star)}{\partial N_{k-1}} \end{pmatrix} - \mu_0^\star \begin{pmatrix} \frac{\partial P(\mathbf{N}^\star)}{\partial N_1} \\ \frac{\partial P(\mathbf{N}^\star)}{\partial N_2} \\ \vdots \\ \frac{\partial P(\mathbf{N}^\star)}{\partial N_{k-1}} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{4.16}$$

$$\mu_0^\star \left(\hat{P} - P(\mathbf{N}^\star) \right) = 0, \tag{4.17}$$

where $\mu_0^\star \geq 0$. We know, since $\mathbf{N}^\star$ is not the optimal solution of the unconstrained problem, that $\nabla J(\mathbf{N}^\star) \neq 0$. $\nabla J(\mathbf{N}^\star) \neq 0$ means that not all $\partial J(\mathbf{N}^\star)/\partial N_i$ are equal to 0. Thus, $\mu_0^\star \neq 0$ since otherwise condition (4.16) would be violated. By condition (4.17), the optimal solution $\mathbf{N}^\star$ satisfies $P(\mathbf{N}^\star) = \hat{P}$. Since g_0 is the only active

[5]An interior solution means that all $N_i^\star$ in $\mathbf{N}^\star$ are greater than $N_{\min}$. We discuss the case in which some $N_i^\star$ are on the boundary, i.e., $N_i^\star = N_{\min}$, in Appendix B.

inequality constraint, $\mathbf{N}^\star$ is regular. In addition, conditions (4.16) and (4.17) reveal how we could find $\mu_0^\star$ and $\mathbf{N}^\star$. For every $\mu_0^\star$, condition (4.16) determines $\mathbf{N}^\star$ since there are $k-1$ equations and $k-1$ unknowns. Therefore, we can think of $\mathbf{N}^\star = \mathbf{N}^\star(\mu_0^\star)$. We search for a value of $\mu_0^\star$ such that $P(\mathbf{N}^\star(\mu_0^\star)) = \hat{P}$. As we indicate in the following, this is exactly what our algorithm does.

Replacing $\mu_0^\star$ by $\mu_0 > 0$ in constraint (4.16) gives

$$-\begin{pmatrix} \frac{\partial J(\bar{\mathbf{N}})}{\partial N_1} \\ \\ \frac{\partial J(\bar{\mathbf{N}})}{\partial N_2} \\ \\ \vdots \\ \\ \frac{\partial J(\bar{\mathbf{N}})}{\partial N_{k-1}} \end{pmatrix} - \mu_0 \begin{pmatrix} \frac{\partial P(\bar{\mathbf{N}})}{\partial N_1} \\ \\ \frac{\partial P(\bar{\mathbf{N}})}{\partial N_2} \\ \\ \vdots \\ \\ \frac{\partial P(\bar{\mathbf{N}})}{\partial N_{k-1}} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{4.18}$$

where $\bar{\mathbf{N}}$ is the unique solution of (4.18). Note that $\bar{\mathbf{N}}$ is the solution of the following optimization problem

$$\begin{aligned} \min_{\mathbf{N}} \quad & -\bar{J}(\mathbf{N}) = -J(\mathbf{N}) + \mu_0 \left(\hat{P} - P(\mathbf{N})\right) \\ \text{subject to} \quad & N_{\min} - N_i \leq 0, \forall i = 1, \cdots, k-1, \end{aligned} \tag{4.19}$$

which is equivalent to

$$\begin{aligned} \max_{\mathbf{N}} \quad & \bar{J}(\mathbf{N}) = J(\mathbf{N}) - \mu_0 \left(\hat{P} - P(\mathbf{N})\right) \\ \text{subject to} \quad & N_{\min} - N_i \leq 0, \forall i = 1, \cdots, k-1, \end{aligned} \tag{4.20}$$

or

$$\begin{aligned} \max_{\mathbf{N}} \quad & \bar{J}(\mathbf{N}) = AP(\mathbf{N}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i - \mu_0 \left(\hat{P} - P(\mathbf{N})\right) \\ \text{subject to} \quad & N_{\min} - N_i \leq 0, \forall i = 1, \cdots, k-1, \end{aligned} \tag{4.21}$$

or

$$\begin{aligned} \max_{\mathbf{N}} \quad & \bar{J}(\mathbf{N}) = (A + \mu_0) P(\mathbf{N}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i \\ \text{subject to} \quad & N_i \geq N_{\min}, \forall i = 1, \cdots, k-1, \end{aligned} \tag{4.22}$$

or, finally,

$$\begin{aligned} \max_{\mathbf{N}} \quad & \bar{J}(\mathbf{N}) = A' P(\mathbf{N}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i \\ \text{subject to} \quad & N_i \geq N_{\min}, \forall i = 1, \cdots, k-1. \end{aligned} \tag{4.23}$$

where $A' = A + \mu_0$. This is exactly the unconstrained problem (4.5), and $\bar{\mathbf{N}}$ is its optimal solution. Note that $\mu_0 > 0$ indicates that $A' > A$. In addition, the KKT condition (4.17) indicates that the optimal solution of the constrained problem, $\mathbf{N}^\star$, satisfies $P(\mathbf{N}^\star) = \hat{P}$. This means that, for every $A' > A$ (or $\mu_0 > 0$), we can find the corresponding optimal solution $\bar{\mathbf{N}}$ satisfying condition (4.18) by solving problem (4.5), and, we need to find the A' such that the solution to problem (4.5), denoted as $\mathbf{N}'(A')$, satisfies $P(\mathbf{N}'(A')) = \hat{P}$. Then, $\mu_0 = A' - A$ and $\mathbf{N}'(A')$ satisfy conditions (4.16) and (4.17). Hence, $\mu_0 = A' - A$ is exactly the Lagrange multiplier satisfying the KKT conditions of our constrained problem, and $\mathbf{N}^\star = \mathbf{N}'(A')$ is the optimal solution of our constrained problem. Consequently, solving the constrained problem (4.3) through our algorithm is essentially finding the unique Lagrange multipliers and optimal solution of the problem. We have proven our assertion.

Therefore, in our algorithm, we conduct an one-dimensional search over $A' > A$ and stop after we find the A' such that the solution to the unconstrained problem satisfies $P(N_1', \cdots, N_{k-1}') = \hat{P}$. We conclude that it is also the desired solution of the constrained problem. We state the algorithm for solving (4.3) in Section 4.2.2.

4.2.2 Algorithm Statement

1. Check the feasibility of the problem. We describe this in Section 4.2.5.

2. Solve the unconstrained problem (4.4). If the solution $(N_1^u, \cdots, N_{k-1}^u)$ satisfies $P(N_1^u, \cdots, N_{k-1}^u) \geq \hat{P}$, stop. The solution of the constrained problem is $(N_1^\star, \cdots, N_{k-1}^\star) = (N_1^u, \cdots, N_{k-1}^u)$. This step is also the necessity check of the algorithm as we point out in Section 4.2.5.

3. If $(N_1^u, \cdots, N_{k-1}^u)$ does not satisfy the production rate constraint, do a one-dimensional search over $A' > A$ to find A' such that the solution of the unconstrained problem (4.5) satisfies $P(N_1', \cdots, N_{k-1}') = \hat{P}$. Stop. The desired solution is $(N_1^\star, \cdots, N_{k-1}^\star) = (N_1', \cdots, N_{k-1}')$.

4.2.3 An Example of the Algorithm

We provide an example to show the algorithm. It is a three-machine two-buffer line with parameters $r_1 = .12$, $p_1 = .01$, $r_2 = .09$, $p_2 = .01$, $r_3 = .11$, and $p_3 = .01$. The profit function is $J(N_1, N_2) = 1000P(N_1, N_2) - .5N_1 - N_2$. Suppose first that the required production rate, $\hat{P}$, is .85. Solving the unconstrained problem and letting (N_1^u, N_2^u) be the optimal solution yield $P(N_1^u, N_2^u) = .8576$. Since $P(N_1^u, N_2^u) > \hat{P} = .85$, the solution (N_1^u, N_2^u) is equivalent to the solution of the constrained problem and no further search on $A' > A$ is needed.

Next suppose that the required production rate, $\hat{P}$, is .88. In this case, the optimal solution of the unconstrained problem does not satisfy $P(N_1^u, N_2^u) \geq \hat{P}$. Thus, we need to conduct the one-dimensional search over $A' > A$ to find the optimal solution of the constrained problem. So, we solve the unconstrained prob-

lem for $A' = 1000, 2000, \cdots, 6000$, and the optimal solutions for distinct A' are shown in Figure 4-2. Figure 4-2 also shows that the required production rate is $\hat{P} = .88$. The locus of unconstrained optima of the cost function is sketched. Let $(N_1'(A'), N_2'(A'))$ be the optimal solution of the unconstrained problem for a certain A'. Note that for A' =1000, 2000, and 3000, the unconstrained optima have $P(N_1'(A'), N_2'(A')) < .88$ while for A' =4000, 5000, and 6000, the unconstrained optima have $P(N_1'(A'), N_2'(A')) > .88$. Therefore, if the problem to be solved is to maximize $1000P - .5N_1 - N_2$ subject to $P \geq .88$, then the solution is the intersection of $P = .88$ and the locus of unconstrained optima.

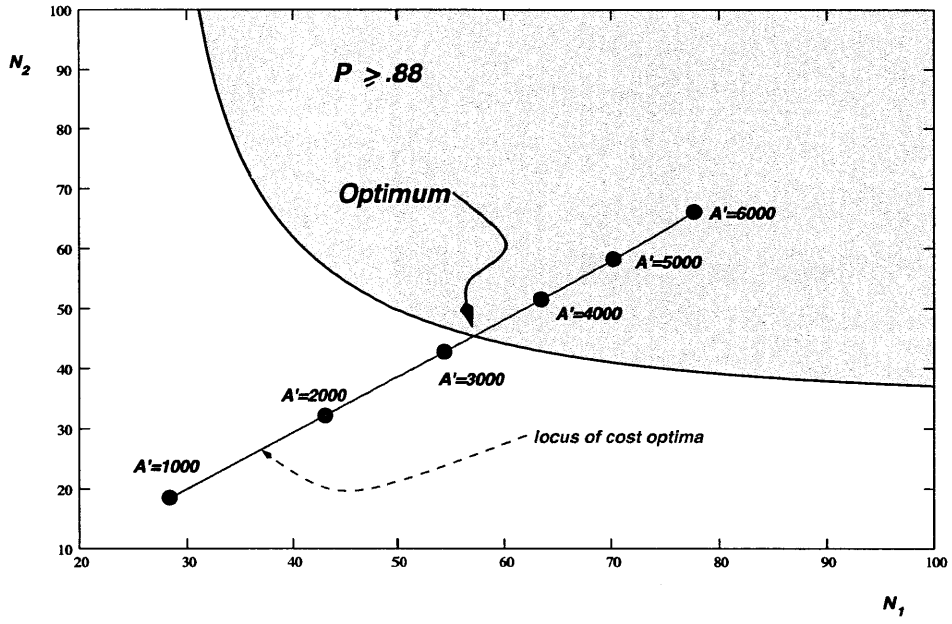


Figure 4-2: An example of the algorithm

4.2.4 Detailed Description of the Algorithm

We describe in detail the algorithm we propose to solve the constrained problem (4.3) here. The algorithm includes solving the unconstrained problem and the one-dimensional search over $A' > A$, if necessary. We have explained that we can treat the decision variables, N_i, as continuous variables, and $J(N_1, \cdots, N_{k-1})$ as a continuously differentiable function. Therefore, we adopt a gradient method that is based on the

decomposition (Gershwin 1987a) and the DDX algorithm (Dallery et al. 1988) to solve the unconstrained problem.

Gradient method for the unconstrained problem

We solve the unconstrained problem with a gradient method. An initial guess of buffer distribution $(N_1^0, \cdots, N_{k-1}^0)$ is selected first. For example, N_i^0 can be chosen as the minimum value to satisfy $P(\infty, \cdots, N_i^0, \cdots, \infty) \geq \hat{P}$. If this inequality is satisfied with equality in all i, then the initial guess satisfies $P(N_1^0, \cdots, N_{k-1}^0) \leq \hat{P}$. This is how we choose the initial guess in our implementation of the algorithm. However, it is helpful to indicate that our algorithm does not require that $P(N_1^0, \cdots, N_{k-1}^0) \leq \hat{P}$. We have verified this with experiments and the initial guesses $(N_1^0, \cdots, N_{k-1}^0)$ such that $P(N_1^0, \cdots, N_{k-1}^0) > \hat{P}$ lead to the same solution.

Then, we calculate the gradient direction to move in $(N_1, \cdots, N_{k-1})$ space. A line search is then conducted in that direction until a maximum is encountered. This becomes the next guess. A new direction is chosen and the process continues until no further improvement can be achieved. There is no analytical expression to compute profits of lines having more than two machines. Consequently, to determine the search direction, we compute the gradient, $\mathbf{g}$, according to a forward difference formula, which is

$$g_i = \frac{J(N_1, \cdots, N_i + \delta N_i, \cdots, N_{k-1}) - J(N_1, \cdots, N_i, \cdots, N_{k-1})}{\delta N_i} \tag{4.24}$$

where g_i is the gradient component of Buffer B_i, J is the profit of the line and can be obtained by equation (4.1), and δN_i is the increment of Buffer B_i. Since we treat N_i as continuous variables, in the gradient calculation above, we choose $\delta N_i = .01$, which has proved to be a good choice in all experiments we have conducted.

Apart from acquiring the gradient direction, we still need to determine the step size, a. We conduct a bisection search to find a such that $J(N_1 + ag_1, \cdots, N_{k-1} + ag_{k-1})$, or $J(\mathbf{N} + a\mathbf{g})$, is maximized. Then, we calculate the next gradient and repeat. This process ends when there is no improvement in profit or when all components of

the gradient are sufficiently small. A block diagram of this gradient method appears in Figure 4-3.

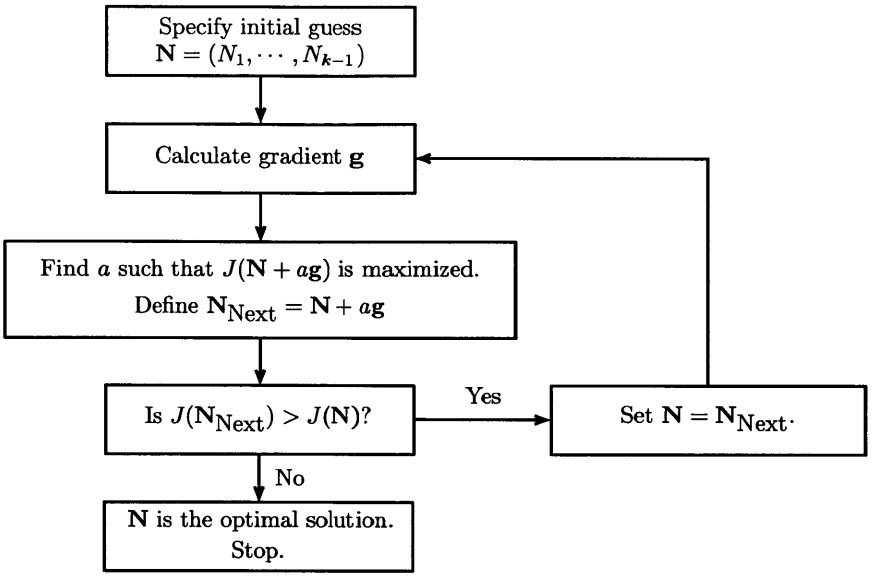


Figure 4-3: Block diagram of the gradient method

One-dimensional search over $A' > A$

To find A' such that the optimal solution of the unconstrained problem (4.5) satisfies $P(N'_1, \cdots, N'_{k-1}) = \hat{P}$, we use the Newton Chord method (Isaacson and Keller 1994), which is an efficient way to find t' such that $f(t') = 0$ for a given function $f(\cdot)$. Thus, in our algorithm, for any particular value of A', we define $f(A')$ as

$$f(A') = P(\mathbf{N}'(A')) - \hat{P} \tag{4.25}$$

where $\mathbf{N}'(A')$ is the optimal buffer allocation associated with that A' and $\hat{P}$ is the required production rate. The Newton Chord method in our algorithm consists of the following three steps:

1. Guess A'_0 and A'_1. Calculate the approximate slope

$$s = \frac{f(A'_1) - f(A'_0)}{A'_1 - A'_0}. \tag{4.26}$$

2. Choose A'_2 so that

$$f(A'_0) + (A'_2 - A'_0)s = 0, \tag{4.27}$$

or

$$A'_2 = -\frac{f(A'_0)}{s} + A'_0. \tag{4.28}$$

3. Repeat with $A'_0 = A'_1$ and $A'_1 = A'_2$ until $|f(A'_0)|$ is small enough.

In our implementation, the termination criterion is $|f(A'_0)| \leq 10^{-4}$. The first two values of A' are $A'_0 = A$ and $A'_1 = 1.5A$.

4.2.5 Implementation Issues

Implementation issues about the algorithm include the feasibility and necessity of the algorithm, the initialization of the DDX algorithm of the decomposition, and the conversion from continuous solutions to integers.

Before running the algorithm, we should ensure that the required production rate, $\hat{P}$, is feasible for the line to be optimized. This means that $\hat{P}$ should satisfy

$$\hat{P} < \min_i \frac{r_i}{r_i + p_i} \tag{4.29}$$

where $r_i/(r_i + p_i)$ is the isolated production rate of Machine M_i. If this fails, no set of buffers can satisfy the production rate constraint.

We also have to make sure that there is a need to conduct the one-dimensional search over $A' > A$ of our algorithm. This actually can be decided after we first solve the unconstrained problem. We have indicated this in Section 4.2.1 and restate it as the second step of the algorithm in Section 4.2.2. If the unconstrained problem (4.4) has a solution $(N_1^u, \cdots, N_{k-1}^u)$ in which $P(N_1^u, \cdots, N_{k-1}^u) \geq \hat{P}$, then we do not need to implement the one-dimensional search over $A' > A$ and the solution is equivalent to that of the constrained problem (4.3).

Furthermore, since we apply the DDX algorithm for the decomposition and it is an iterative algorithm, we must initialize it whenever it is called in our algorithm.

We use the most recent value of $(N_1, \cdots, N_{k-1})$ from the last evaluation, instead of a standard initialization, to reduce the numbers of iterations and the two-machine-line evaluations.

Finally, it is necessary to point out that, to use the result of the algorithm in practical production line design for factories, we have to convert it back to integers. To do this, for each component in $\mathbf{N}^\star$, let $N_i^U = \lceil N_i^\star \rceil$, the smallest integer that is larger than $N_i^\star$, and $N_i^L = \lfloor N_i^\star \rfloor$, the largest integer that is smaller than $N_i^\star$. Then, we compute the production rates and the profits for these 2^{k-1} combinations of $(N_1, \cdots, N_{k-1})$, where N_i is equal to either N_i^L or N_i^U. Note that, since $(N_1^\star, \cdots, N_{k-1}^\star)$ satisfies the production rate constraint, then $(N_1^U, \cdots, N_{k-1}^U)$ must satisfy the production rate constraint as well because of the monotonicity of $P(\mathbf{N})$. Therefore, among all 2^{k-1} candidates, there must be at least one feasible combination that satisfies the production rate constraint. Among all feasible combinations of integer buffer sizes, we choose the one that maximizes the profit of the line as the final integer solution of optimal buffer sizes.

4.3 Numerical Results and Analysis

Numerical results are provided to show not only the efficiency of our algorithm but also its implementation process. Hence, taking a four-machine three-buffer line as an example, we first explain how $P(\mathbf{N}^\star)$ changes with A'. Then, we apply the algorithm to both short and long lines to illuminate its efficiency. Computation issues are discussed at the end of this section. In the implementation of the algorithm, we let $N_{\min}$ be $4+\epsilon$, where $\epsilon = 10^{-6}$. In addition, the algorithm is written with Matlab and run for all experiments on a computer with a 2.4 GHz Intel Core 2 Duo CPU.

4.3.1 Behavior of the Algorithm

We consider two four-machine three-buffer lines to study the behavior of the algorithm. The parameters of the first four-machine line are summarized in Table 4.1. All cost coefficients are set to be 1 and therefore the profit is calculated as

$$J(N_1, \cdots, N_3) = AP(N_1, \cdots, N_3) - \sum_{i=1}^{3} N_i - \sum_{i=1}^{3} \bar{n}_i. \tag{4.30}$$

To study the behavior of the algorithm, we run it for this line to generate the curve of $P(\mathbf{N}^\star)$ versus A'. We vary the A' from 0 to 1000 with a step size of 1 and the desired curve is shown in Figure 4-4. There are four segments in the curve. The flat segment stands for the case in which the optimal sizes of all buffers are $N_{\min}$. For A' ranging from 0 to 257, the optimal buffer sizes for the line stay on the boundary of the feasible region so they are $(N_{\min}, N_{\min}, N_{\min})$. The production rates associated with those A' are identical, forming the horizontal segment in the curve. After A' passes 257, one of the optimal buffer sizes ($N_2^\star$) becomes greater than $N_{\min}$, i.e., it turns to a non-minimal value from $N_{\min}$. From that time, the production rate begins to monotonically increase with A'. Each time a new buffer becomes non-minimal, the derivative of the production rate curve changes slightly, but the curve remains continuous. Figure 4-4 suggests that $P(\mathbf{N}^\star)$ for which not all components in $\mathbf{N}^\star$ are $4 + \epsilon$ increases with A' monotonically. This system behavior further verifies our algorithm; as A' increases, $P(\mathbf{N}^\star)$ increases monotonically so we can eventually find the A' such that $P(\mathbf{N}^\star) \geq \hat{P}$ (for $\hat{P}$ feasible).

Table 4.1: Parameters for the system behavior, Experiment 1

Machine	M_1	M_2	M_3	M_4
r_i	.1	.2	.13	.09
p_i	.01	.02	.01	.01
P_i	.909	.909	.929	.900

We study another four-machine three-buffer line, where we vary the target production rate $\hat{P}$ and study $N_i^\star$ and $J(\mathbf{N}^\star)$. Parameters of the line are listed in table

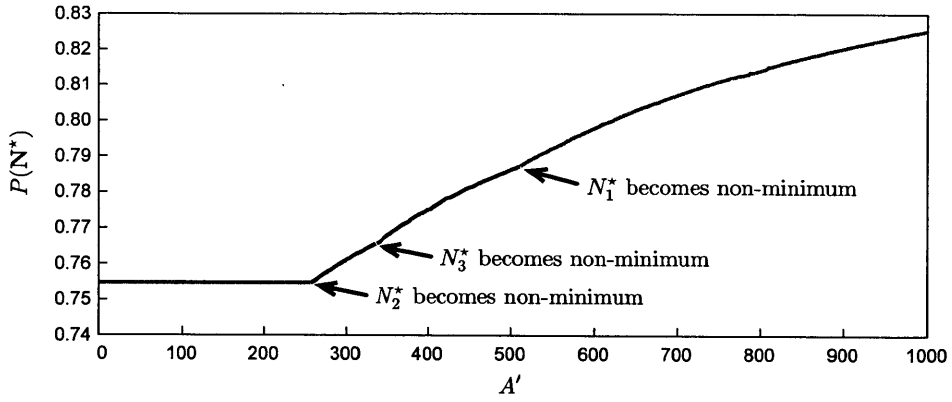


Figure 4-4: System behavior of the algorithm

Table 4.2: Parameters for the system behavior, Experiment 2

Machine	M_1	M_2	M_3	M_4
r_i	.1	.16	.1	.12
p_i	.01	.01	.01	.009
P_i	.909	.941	.909	.930
Buffer	B_1	B_2	B_3	
b_i	1	30	1	
c_i	1	1	1	

4.2. Note that the buffer size coefficient of B_2 is 30, which is much larger than those of the other two buffers. The revenue coefficient $A = 3000$. We notice that M_1 and M_3 have the smallest isolated production rate, which is .909. Therefore, for $\hat{P}$ to be feasible for the line, it has to be smaller than .909. Therefore, we vary $\hat{P}$ from .8 to .9088. The results of the example[6] are illustrated in Figure 4-5. Part of the results is also listed in Table 4.3.

We observe that if we optimize the line without the production rate constraint, then the production rate associated with the optimal solution is .8458. This means that if the target production rate $\hat{P} \leq .8458$, the optimal solutions of the line for different $\hat{P}$ will be the same and the production rate constraint will be inactive. This is demonstrated in both Figure 4-5 and Table 4.3. For instance, in Figure 4-5, $N_1^\star$,

[6]In this example, we keep the optimal buffer sizes as non-integers.

Table 4.3: Optimal results of algorithm behavior, Experiment 2

$\hat{P}$	$N_1^\star$	$N_2^\star$	$N_3^\star$	$\bar{n}_1$	$\bar{n}_2$	$\bar{n}_3$	$P(\mathbf{N}^\star)$	$J(\mathbf{N}^\star)$
.800	28.92	4.00	30.34	19.25	2.01	7.33	.8458	2329.51
.810	28.92	4.00	30.34	19.25	2.01	7.33	.8458	2329.51
.820	28.92	4.00	30.34	19.25	2.01	7.33	.8458	2329.51
.830	28.92	4.00	30.34	19.25	2.01	7.33	.8458	2329.51
.840	28.92	4.00	30.34	19.25	2.01	7.33	.8458	2329.51
.845	28.92	4.00	30.34	19.25	2.01	7.33	.8458	2329.51
.850	35.42	4.00	33.00	23.95	2.03	7.92	.8500	2327.69
.855	44.04	4.00	40.57	30.25	2.04	9.12	.8550	2318.85
.860	58.49	4.02	51.64	41.36	2.06	10.55	.8600	2295.17
.864	63.87	5.04	55.52	44.96	2.60	11.39	.8640	2262.48
.868	66.68	6.53	57.41	46.21	3.41	12.18	.8680	2222.46
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
.904	252.79	53.53	184.50	132.51	35.49	35.79	.9040	465.03
.906	325.19	68.29	227.72	156.50	47.38	39.86	.9060	−127.53
.908	514.80	100.10	322.37	205.40	75.07	44.81	.9080	−1441.55
.9088	792.88	140.14	499.81	247.51	112.45	47.10	.9088	−3177.73

$N_2^\star$, $N_3^\star$, and $J(\mathbf{N}^\star)$ are all constant (see the horizontal part of the corresponding graph).

The second observation we make from these results is the cases where $N_2^\star = N_{\min}$. Since $b_2(= 30)$ is much larger than b_1 and b_3, we expect that $N_2^\star = N_{\min}$ when $\hat{P}$ is small to avoid large buffer space cost, because otherwise $N_2^\star$ has to be larger than $N_{\min}$ to achieve the target production rate. Let us first check the maximum production rate that can be achieve on the line when $N_2 = N_{\min} = 4$. It is easy to compute that $P(\infty, 4, \infty) = .868$. Therefore, we know that for $\hat{P} > .868$, $N_2^\star$ has to be larger than $N_{\min}$. However, from Table 4.3 we notice that $N_2^\star$ starts deviating from $N_{\min}$ when $\hat{P}$ is .86. This is because, although it is possible to achieve $\hat{P} = .86$ with $N_2 = N_{\min}$, the extra cost in Buffers B_1 and B_3 makes the solution less profitable. We verify this as follows. Suppose we restrict $N_2^\star$ to be 4. Then, the optimal values of N_1 and N_3 for which $P(\mathbf{N}) = .86$ are 58.85 and 51.94, respectively. The profit is 2295.01, which is smaller than that for $\hat{P} = .86$ in table 4.3. This is because, although the cost of increasing the size of B_2 is high (since $b_2 = 30$), it costs more if we increase

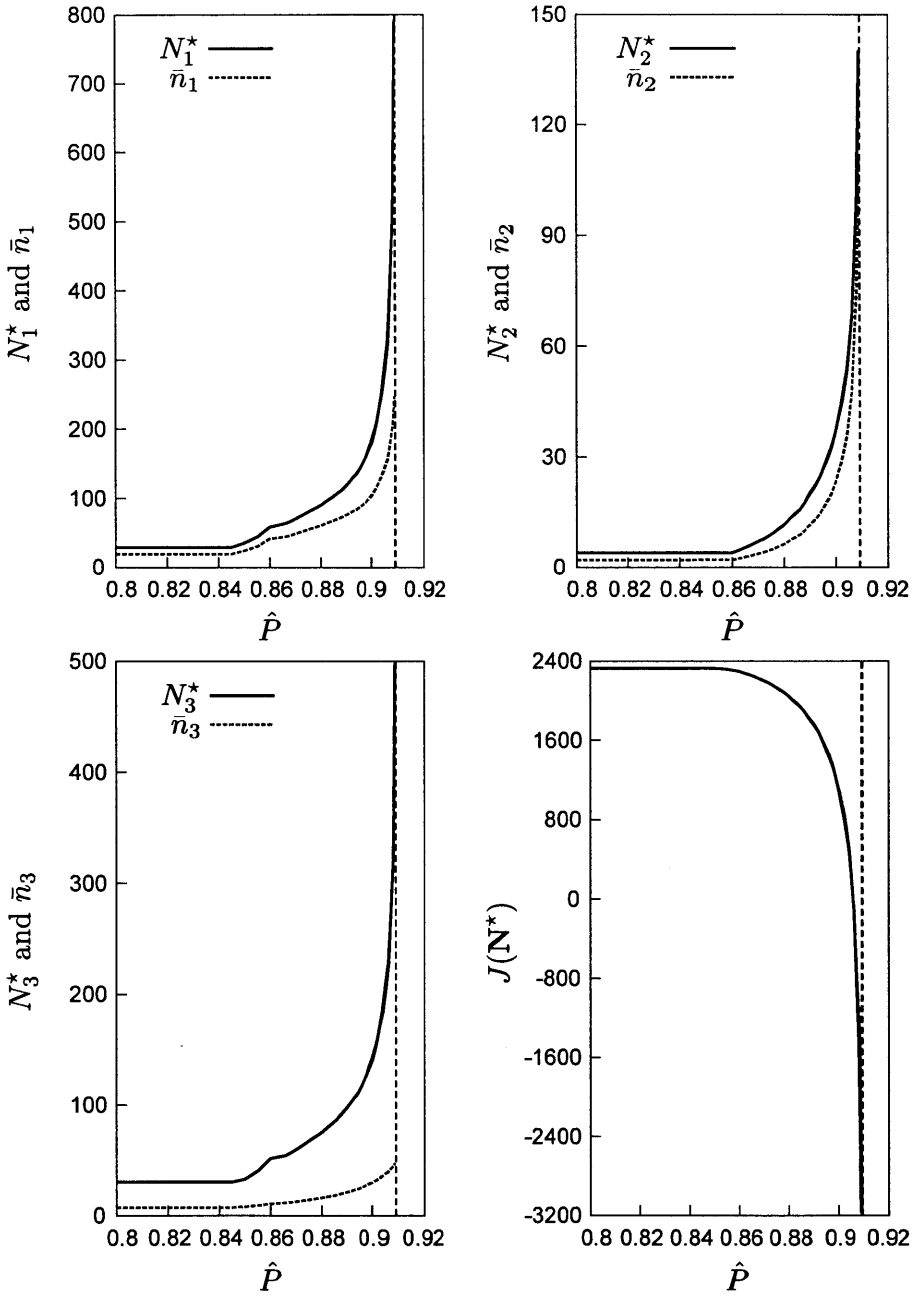


Figure 4-5: Impact of $\hat{P}$ on $N_i^\star$ and $J(\mathbf{N}^\star)$

the sizes of Buffers B_1 and B_3 instead B_2 to achieve the target production rate. The fact that $N_2^\star$ starts deviating from $N_{\min}$ when $\hat{P}$ is .86 also explains why there are two observable segments in the curves of $N_1^\star$ and $N_3^\star$ for $\hat{P} < .86$ and $\hat{P} \geq .86$.

Finally, Figure 4-5 indicates that when $\hat{P}$ goes to .909, which is the maximum production rate the line may have, $N_1^\star$, $N_2^\star$, and $N_3^\star$ will go to ∞. This is because the buffers have to be sufficiently large enough to eliminate any propagation of failures of one machine on the others to avoid lost of production rate. As a result, the profit of the line will go to $-\infty$ as the revenue of the line is upper bounded by $A\hat{P}$ while the cost of the line is not bounded.

4.3.2 Experiments on Short Lines

The proposed algorithm optimizes short lines very quickly, which are shown by the following two experiments. We optimize a five-machine line and a six-machine line, and compare the optimal solutions of the algorithm with solutions gained from searching the $P = \hat{P}(N_1, \cdots, N_{k-1})$ surface in $(N_1, \cdots, N_{k-1})$ space[7].

Experiment on a five-machine four-buffer line

The machine parameters are listed in Table 4.4. The required production rate $\hat{P}$ is .88. It is easy to check that the isolated production rate of the bottleneck of the line, Machine M_4, is greater than $\hat{P}$, so the problem is feasible. The profit is calculated as:

$$J = 2500P(\mathbf{N}) - \sum_{i=1}^{4} N_i - \sum_{i=1}^{4} \bar{n}_i.$$

Table 4.4: Machine parameters of the five-machine line experiment

machine	M_1	M_2	M_3	M_4	M_5
r_i	.11	.12	.10	.09	.10
p_i	.008	.01	.01	.01	.01
P_i	.932	.923	.909	.900	.909

To verify the optimal solution, we conduct a search on the $\hat{P}$ surface in $(N_1, \cdots, N_4)$ space. We search on the surface around the optimal solution of our algorithm. Exper-

[7]For a brief introduction about the $\hat{P}$ surface search, see Appendix F. For problems where the production rate constraint is inactive, we search the $P \geq \hat{P}$ surface.

imental results including the optimal solutions both from our algorithm and $\hat{P}$ surface search, and the error, are listed in Table 4.5. We see that the optimal solutions from the algorithm and from the $\hat{P}$ surface search are exactly the same. Computer time for this experiment is 2.05 seconds. The number of the two-machine-line evaluations is 77682. (The reason we provide this number is that it is not affected by the capability and performance of the computer that runs the algorithm.) Hence, our algorithm offers accurate results very quickly.

Table 4.5: Results of the five-machine line experiment

	$\hat{P}$ Surface Search	The algorithm	error
$P(\mathbf{N}^\star)$	.8800	.8800	0%
$N_1^\star$	29	29	0%
$N_2^\star$	58	58	0%
$N_3^\star$	93	93	0%
$N_4^\star$	88	88	0%
$\bar{n}_1(\mathbf{N}^\star)$	19.1842	19.1842	0%
$\bar{n}_2(\mathbf{N}^\star)$	34.0069	34.0069	0%
$\bar{n}_3(\mathbf{N}^\star)$	48.6107	48.6107	0%
$\bar{n}_4(\mathbf{N}^\star)$	32.1166	32.1166	0%
$J(\mathbf{N}^\star)$ ($/time unit)	1798.08	1798.08	0%

The optimal solution reveals that, since Machine M_4 is the bottleneck, the optimal size of Buffer B_3 is greater than the optimal sizes of other three buffers to absorb the large variability of M_4. Next we change b_3, the cost coefficient associated with buffer size of B_3, to 2. This means that the line pays more for the buffer size of B_3. Thus, we expect the optimal solution for the new line to have a smaller size for B_3 while greater sizes for the other three buffers to guarantee the performance of the line in terms of achieving the target production rate. Experimental results confirm our expectation (See Table 4.6). We see that the optimal size of B_3 is reduced from 93 to 79, and the optimal sizes of other three buffers increase. The maximum error is 2.02% and appears in $N_4^\star$. However, the profit error is .02% which is very small. Computer time for the revised experiment is 2.19 seconds. The number of the two-machine-line evaluations is 72264. To further study this phenomenon, we conduct more experiments for this

line by varying the cost coefficient of B_3 from 0 to 14 with a step size of 0.2, and report results in Figure 4-6[8]. Figure 4-6 indicates that as the cost coefficient of B_3 becomes larger and larger, the optimal value of B_3 becomes smaller to limit the cost spent on it. Meanwhile the optimal values of the other three buffers get larger so the line maintains the required production rate.

Table 4.6: Results of the modified five-machine line experiment

	$\hat{P}$ Surface Search	The algorithm	error
$P(\mathbf{N}^\star)$	.8800	.8800	.00%
$N_1^\star$	31	31	.00%
$N_2^\star$	65	65	.00%
$N_3^\star$	78	79	1.28%
$N_4^\star$	99	97	2.02%
$\bar{n}_1(\mathbf{N}^\star)$	20.7534	20.7515	.01%
$\bar{n}_2(\mathbf{N}^\star)$	39.9739	39.9653	.02%
$\bar{n}_3(\mathbf{N}^\star)$	41.0795	41.7460	1.62%
$\bar{n}_4(\mathbf{N}^\star)$	34.1755	33.8337	1.00%
$J(\mathbf{N}^\star)$ ($/time unit)	1713.02	1712.75	.02%

Experiment on a six-machine five-buffer line

The machine parameters and cost coefficients are provided in Table 4.7. The required production rate $\hat{P}$ is still .88, and it is easy to check that it is feasible for the line. The profit is calculated as

$$J = 3000P(\mathbf{N}) - \sum_{i=1}^{4} b_i N_i - \sum_{i=1}^{4} c_i \bar{n}_i.$$

Experimental results are presented in Table 4.8. The optimal buffer sizes from the algorithm and the $\hat{P}$ surface search are the same. Computer time for this experiment is 6.83 seconds. The number of the two-machine-line evaluations is 176216.

[8]Note that the optimal buffer sizes in Figure 4-6 are kept as non-integers.

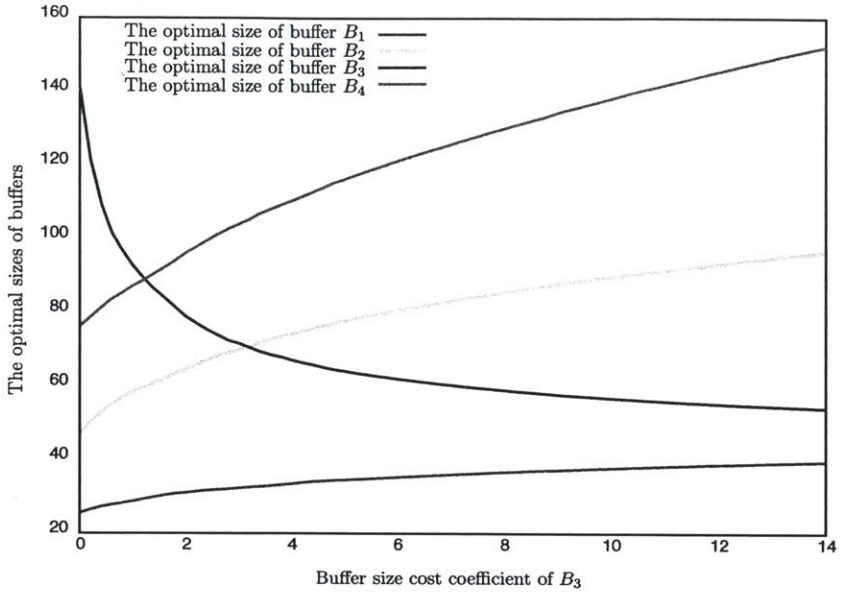


Figure 4-6: Optimal buffer spaces vs. cost coefficient of B_3

Table 4.7: Machine parameters of the six-machine line experiment

machine	M_1	M_2	M_3	M_4	M_5	M_6
r_i	.11	.12	.10	.09	.10	.11
p_i	.008	.01	.01	.01	.01	.009
P_i	.932	.923	.909	.900	.909	.924
buffer	B_1	B_2	B_3	B_4	B_5	
b_i	1.0	2.0	.5	.8	1.0	
c_i	1.0	1.0	2.0	1.0	1.5	

4.3.3 Experiments on Long Lines

Next we apply the algorithm to a 10-machine 9-buffer line. Machine parameters are shown in Table 4.9. The target production rate is $\hat{P} = .88$ for this line. In addition, we set all cost coefficients to 1 and therefore the profit of the line is

$$J = 5000P(\mathbf{N}) - \sum_{i=1}^{9} N_i - \sum_{i=1}^{9} \bar{n}_i.$$

Experimental results are provided in Table 4.10. We see that the optimal buffer

Table 4.8: Results of the six-machine line experiment

	$\hat{P}$ Surface Search	The algorithm	error
$P(\mathbf{N}^\star)$	.8800	.8800	0%
$N_1^\star$	33	33	0%
$N_2^\star$	46	46	0%
$N_3^\star$	104	104	0%
$N_4^\star$	113	113	0%
$N_5^\star$	57	57	0%
$\bar{n}_1(\mathbf{N}^\star)$	22.3513	22.3513	0%
$\bar{n}_2(\mathbf{N}^\star)$	26.2354	26.2354	0%
$\bar{n}_3(\mathbf{N}^\star)$	51.6319	51.6319	0%
$\bar{n}_4(\mathbf{N}^\star)$	43.0599	43.0599	0%
$\bar{n}_5(\mathbf{N}^\star)$	17.6553	17.6553	0%
$J(\mathbf{N}^\star)$ ($/time unit)	2094.22	2094.22	0%

Table 4.9: Machine parameters of the 10-machine line experiment

machine	M_1	M_2	M_3	M_4	M_5
r_i	.11	.12	.10	.09	.10
p_i	.008	.01	.01	.01	.01
P_i	.932	.923	.909	.900	.909
machine	M_6	M_7	M_8	M_9	M_{10}
r_i	.11	.10	.11	.12	.10
p_i	.01	.009	.01	.009	.008
P_i	.917	.917	.917	.930	.926

sizes from the algorithm and the surface search are the same. The algorithm provides accurate optimal solutions for long lines as well. (We provide more numerical experiments on randomly generated lines in Section 4.3.6.) Computer time for this experiment is 20.84 seconds. The number of the two-machine-line evaluations is 938944.

4.3.4 Computation Speed

In this section, we discuss the computation speed of the algorithm. Although we have shown that the algorithm offers the optimal solution for a 10-machine 9-buffer line

Table 4.10: Results of the 10-machine line experiment

	$\hat{P}$ Surface Search	The algorithm	error
$P(\mathbf{N}^\star)$	.8800	.8800	0%
$N_1^\star$	29	29	0%
$N_2^\star$	60	60	0%
$N_3^\star$	98	98	0%
$N_4^\star$	108	108	0%
$N_5^\star$	84	84	0%
$N_6^\star$	70	70	0%
$N_7^\star$	62	62	0%
$N_8^\star$	48	48	0%
$N_9^\star$	35	35	0%
$\bar{n}_1(\mathbf{N}^\star)$	19.1841	19.1841	0%
$\bar{n}_2(\mathbf{N}^\star)$	35.5039	35.5039	0%
$\bar{n}_3(\mathbf{N}^\star)$	52.8475	52.8475	0%
$\bar{n}_4(\mathbf{N}^\star)$	45.6174	45.6174	0%
$\bar{n}_5(\mathbf{N}^\star)$	34.4532	34.4532	0%
$\bar{n}_6(\mathbf{N}^\star)$	30.3590	30.3590	0%
$\bar{n}_7(\mathbf{N}^\star)$	27.2247	27.2247	0%
$\bar{n}_8(\mathbf{N}^\star)$	18.2801	18.2801	0%
$\bar{n}_9(\mathbf{N}^\star)$	12.3082	12.3082	0%
$J(\mathbf{N}^\star)$ ($/time unit)	3530.23	3530.23	0%

within one minute, it is important to observe the computation speed of algorithm for longer lines. Thus, we run the algorithm for a series of experiments for lines having identical machines. We vary the length of the line from 4 machines up to 30 machines. Machine parameters are $p_i = .01$ and $r_i = .1$. In all cases, we choose a feasible target production rate $\hat{P} = .88$, and the revenue coefficient $A = 500k$ for the line of length k. Furthermore, we initialize $A'_0 = A$ and $A'_1 = A + 1000$. (Note that all these lines require the one-dimensional search over A' to find the corresponding optimal solution that satisfies the respective production rate constraint.) The numbers of the two-machine-line evaluation for lines with different lengths are summarized in Table 4.11 and Figure 4-7, respectively. We fit a curve to those points in order to reveal the relation between the length of the production line and the computation effort of our algorithm. From those points in Figure 4-7, we guess that we can find an exponential

curve to fit them. Using the Curve Fitting Toolbox of Matlab[9], we find

$$y = 1.109 \times 10^6 e^{0.1851k},$$

where y denotes the number of the two-machine-line evaluations and k denotes the length of the line. The curve is shown in Figure 4-7 as well. It can be seen, from Table 4.11, that it takes the algorithm less than 10 minutes to find the optimal buffer allocation for the 15-machine line, and about 20 minutes to find the optimal buffer allocation for the 20-machine line. However, the exponential curve implies that our algorithm needs much more time for lines with more than thirty machines. In practice, it is possible that several machines are located adjacently in series and followed by a buffer. Therefore, while it is not rare for a transfer line to have more than 30 machines, the number of buffers is often much smaller. However, the exponential curve spurs us to study how to reduce the computation time for longer lines. A segmentation method that reduces computer time for long line optimization effectively is studied in Chapter 8. An additive property that provides us valuable insight in long line design is discussed in Chapter 9. We finally present the specific optimal solution for the 30-machine line. The optimal sizes of buffers are shown in Figure 4-8(a). The curve of $\bar{n}_i(\mathbf{N}^\star)/N_i^\star$ is illustrated in Figure 4-8(b).

[9]http://www.mathworks.com/products/curvefitting/.

Table 4.11: Numbers of the two-machine-line evaluations for lines with different lengths

Line length	$P(\mathbf{N}^{\star})$	# of two-mach line eval	Computer time (sec.)
4	.8800	24864	0.972
5	.8800	60210	1.843
6	.8800	116888	3.333
7	.8800	234330	6.345
8	.8800	459264	12.089
9	.8800	896546	23.661
10	.8799	1586672	42.306
11	.8799	2494944	65.675
12	.8800	3777060	99.125
13	.8800	6779256	176.954
14	.8800	10168032	265.304
15	.8799	15253940	398.723
20	.8799	51786204	1166.694
25	.8800	121869164	3089.636
30	.8799	283117352	7008.475

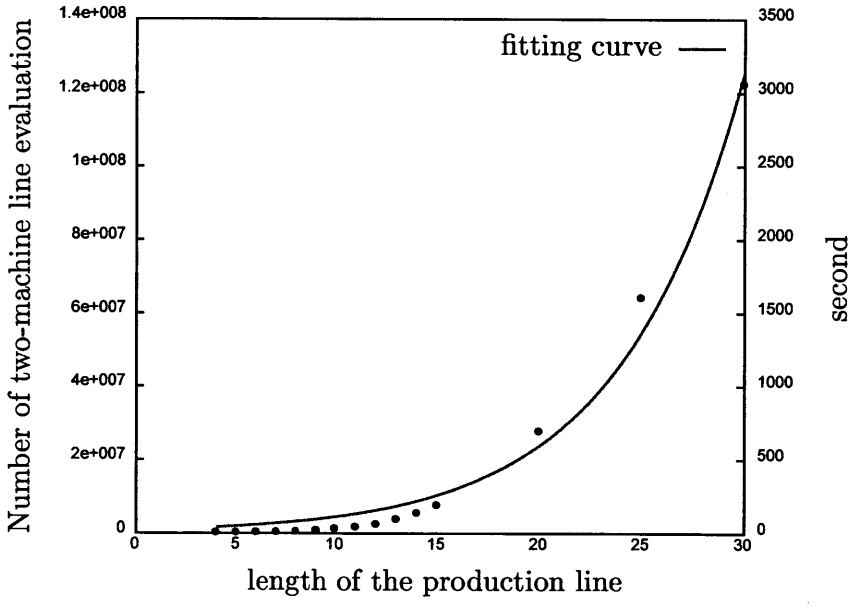
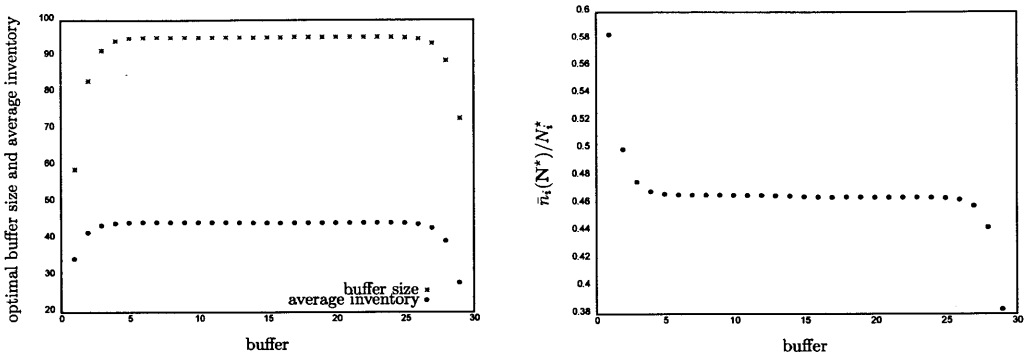


Figure 4-7: Number of the two-machine-line evaluations vs. The length of production lines and its fitting curve



(a) Optimal buffer allocation

(b) $\bar{n}_i/N_i$ curve

Figure 4-8: Results of the 30-machine line

4.3.5 Comparison with Literature

We mention that Schor (1995) and Gershwin and Schor (2000) develop an efficient buffer allocation algorithm that applies a primal-dual approach to minimize the total buffer space under a production rate constraint. Their primal problem is formulated as

$$\begin{aligned} \min_{\mathbf{N}} \quad & \sum_{i=1}^{k-1} N_i \\ \text{subject to} \quad & P(N_1, \cdots, N_{k-1}) \geq \hat{P}, \\ & N_i \geq N_{\min}, \forall i = 1, \cdots, k-1. \end{aligned} \tag{4.31}$$

It can be seen that (4.31) is a special case of our constrained problem (4.3) where $A = 0$, $b_i = 1$, and $c_i = 0, i = 1, \cdots, k-1$. Moreover, Gershwin and Goldis (1995) and Colledani et al. (2003) study problem (4.31) as well. Therefore, in this section, we compare our algorithm to the results reported by Schor (1995), Gershwin and Goldis (1995), and Colledani et al. (2003) for solving (4.31).

On the other hand, Colledani and Tolio (2005) develop a buffer allocation algorithm that solves problem (4.31) for the deterministic multiple failure mode production lines. In addition, Schor (1995) also studies our unconstrained problem (4.4) and reports some numerical examples for the continuous time continuous material

line model. Levantesi et al. (2001) and Tolio et al. (2009)[10] develop different algorithms that solve problem (4.31) for the continuous time continuous material line model. Therefore, we compare our algorithm with Colledani and Tolio (2005) for the deterministic multiple failure mode line model in Section 4.4.1, and with Schor (1995), Levantesi et al. (2001), and Tolio et al. (2009) for the continuous model in Section 4.4.2.

We first consider a balanced 10-machine line[11] with $r_i = .095$, $p_i = .007$, and $\hat{P} = .88$. The optimization results of (4.31) from our algorithm and Schor's algorithm are shown in Table 4.12. Note that, although the two algorithms find different optimal buffer allocations, the total numbers of buffer sizes are both 346, and both of the two buffer allocations satisfy the production rate constraint. It happens that for the problem under consideration, there are more than one optimal solution in terms of the total buffer size[12]. Among those feasible solutions that have the same total buffer size, we choose the one that enables the line to have the maximum production rate as our optimal solution.

Table 4.12: Comparison of algorithms, Experiment 1

	$N_1^\star$	$N_2^\star$	$N_3^\star$	$N_4^\star$	$N_5^\star$	$N_6^\star$	$N_7^\star$	$N_8^\star$	$N_9^\star$	$\sum N_i^\star$	$P(\mathbf{N}^\star)$
Schor (1995)	27	38	42	44	44	44	42	38	27	346	.88009
our algorithm	26	39	42	44	44	44	42	39	26	346	.88010

In addition to the 10-machine line above, Schor (1995) studies an example of a 12-machine line constructed by Park (1993), which also solves problem (4.31). The parameters of the line are listed in Table 4.13. Two target production rates are considered for this line and they are $\hat{P} = .85$ and $\hat{P} = .895$, respectively. The results of the experiment is summarized in Table 4.14.

It can be seen from Table 4.14 that the algorithm of Park (1993) fails to provide the optimal solution when $\hat{P} = .895$ since the total buffer size of Park (1993) is about

[10]This is a short version of Borgh (2010).

[11]See Section 6.2.2 of Schor (1995).

[12]It is helpful to point out clearly that although there are multiple solutions when buffer sizes are restricted to integers, there is a single solution for continuous buffer sizes.

Table 4.13: Machine parameters of the 12-machine line experiment of Park (1993)

machine	M_1	M_2	M_3	M_4	M_5	M_6
r_i	.35	.15	.40	.40	.30	.20
p_i	.037	.015	.02	.03	.03	.01
P_i	.904	.909	.952	.930	.909	.952
machine	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}
r_i	.30	.30	.40	.40	.30	.25
p_i	.02	.02	.02	.03	.03	.01
P_i	.938	.938	.952	.930	.909	.962

Table 4.14: Comparison of algorithms, Experiment 2

case	Park (1993)		Gershwin and Goldis (1995)		Schor (1995)		our algorithm	
	$P(\mathbf{N}^\star)$	$\sum N_i^\star$	$P(\mathbf{N}^\star)$	$\sum N_i^\star$	$P(\mathbf{N}^\star)$	$\sum N_i^\star$	$P(\mathbf{N}^\star)$	$\sum N_i^\star$
$\hat{P} = .85$	.8505	93	.8507	87	.8507	87	.8507	87
$\hat{P} = .895$	.8950	390	.8950	242	.8950	243	.8950	242

1.6 times larger than the other three algorithms. For the case where $\hat{P} = .85$, the algorithm of Park (1993) is about 1.07 times larger than the other three algorithms. Therefore, it provides near optimal solution in this case. The optimal solutions from the other three algorithms are very close for both cases. Schor (1995) does not report the specific buffer allocations for these two cases, but Park (1993) does. However, since the solution of Park (1993) deviates from the other three algorithms, we choose not to report his buffer allocation here. Instead, the buffer distributions for both cases from our algorithm are summarized in Table 4.15.

Table 4.15: Optimal buffer distribution for the 12-machine line of Park (1993)

case	$N_1^\star$	$N_2^\star$	$N_3^\star$	$N_4^\star$	$N_5^\star$	$N_6^\star$	$N_7^\star$	$N_8^\star$	$N_9^\star$	$N_{10}^\star$	$N_{11}^\star$
$\hat{P} = .85$	9	9	9	10	9	8	7	7	7	8	4
$\hat{P} = .895$	57	27	21	25	23	16	15	12	14	20	12

Colledani et al. (2003) study six 10-machine lines for problem (4.31) and compare

their results with Gershwin and Goldis (1995). Here we apply our algorithm to those six lines and compare the results with Colledani et al. (2003) and Gershwin and Goldis (1995). The parameters of these six lines are listed in Table 4.16. (Note that the first line is the same as the 10-machine line studied by Schor 1995, and Line E is a reverse case of Line F.) The target production rate for all these lines is $\hat{P} = .88$. The comparison of the three algorithms on these six lines is summarized in Table 4.17, in which we refer to the algorithm of Colledani et al. (2003) as CMGT and the algorithm of Gershwin and Goldis (1995) as GG. (Again, among all feasible solutions that have the same total buffer size, we choose the one that enables the line to have the maximum production rate as our optimal solution.) It can be seen that our algorithm is accurate as compared to the other two algorithms. Note that Line E is a reverse case of Line F. In all three algorithms, the optimal buffer distribution of Line E is the reverse of the optimal buffer distribution of Line F because costs and constraints are symmetric (and the average inventory $\bar{n}_i$ is not in the cost function).

4.3.6 More Numerical Experiments

Finally, for the deterministic single failure mode production line model, we provide more numerical experiments for 600 randomly generated production lines. These lines are generated according to the method of Gershwin (2011). This method allows us to generate only the relevant and practically important cases without having to generate and then discard any irrelevant cases. Therefore it is very efficient. In particular, we study 200 four-machine lines, 200 six-machine lines, and 200 eight-machine lines. In all these lines, the isolated production rate $P_i = r_i/(r_i + p_i)$ of a given machine is between .909 and .952 with r_i and p_i generated randomly. In addition, the buffer cost coefficients b_i and c_i for any buffer are also generated randomly. The target production rate is $\hat{P} = .88$ for the four-machine lines, and .86 for six-machine and eight-machine lines. The revenue coefficient A is 2000 for the four-machine and six-machine lines, and 4000 for the eight-machine lines. We compare the results from the algorithm with $\hat{P}$ surface search and compute three types of errors. They are the profit error, the production rate error, and the maximum buffer size error. We use

Table 4.16: Parameters of the six lines of Colledani et al. (2003)

	Line A	Line B	Line C	Line D	Line E	Line F
p_1	.007	.007	.007	.007	.010	.001
r_1	.095	.095	.094	.095	.092	.092
p_2	.007	.008	.008	.010	.009	.002
r_2	.095	.094	.095	.090	.092	.092
p_3	.007	.006	.003	.003	.008	.003
r_3	.095	.093	.045	.091	.092	.092
p_4	.007	.007	.004	.005	.007	.004
r_4	.095	.094	.078	.099	.092	.092
p_5	.007	.005	.006	.001	.006	.005
r_5	.095	.095	.069	.095	.092	.092
p_6	.007	.006	.007	.009	.005	.006
r_6	.095	.093	.094	.092	.092	.092
p_7	.007	.009	.008	.009	.004	.007
r_7	.095	.095	.095	.097	.092	.092
p_8	.007	.008	.003	.003	.003	.008
r_8	.095	.094	.045	.096	.092	.092
p_9	.007	.007	.004	.008	.002	.009
r_9	.095	.096	.078	.092	.092	.092
p_{10}	.007	.008	.006	.007	.001	.010
r_{10}	.095	.095	.069	.094	.092	.092

subscripts *alg* and *ss* to distinguish the optimal buffer allocations associated with the algorithm and the surface search, respectively. The three types of errors are defined as

$$J_{\text{err}} = \left| \frac{J(\mathbf{N}^{\star}_{\text{ss}}) - J(\mathbf{N}^{\star}_{\text{alg}})}{J(\mathbf{N}^{\star}_{\text{ss}})} \right| \times 100\%,$$

$$P_{\text{err}} = \left| \frac{P(\mathbf{N}^{\star}_{\text{ss}}) - P(\mathbf{N}^{\star}_{\text{alg}})}{P(\mathbf{N}^{\star}_{\text{ss}})} \right| \times 100\%,$$

and, finally

$$N_{\text{err}} = \max_{i=1,\cdots,k-1} \left\{ \left| \frac{\mathbf{N}^{\star}_{\text{ss}}(B_i) - \mathbf{N}^{\star}_{\text{alg}}(B_i)}{\mathbf{N}^{\star}_{\text{ss}}(B_i)} \right| \times 100\% \right\}.$$

The three types of errors for the 200 four-machine lines are illustrated in Figure 4-9. In particular, for each type error in Figure 4-9, we rank the three types of errors in their corresponding ascending orders respectively. (Therefore, the ith case in the

Table 4.17: Comparison of algorithms on 10-machine lines of Colledani et al. (2003)

Line	method	$N_1^\star$	$N_2^\star$	$N_3^\star$	$N_4^\star$	$N_5^\star$	$N_6^\star$	$N_7^\star$	$N_8^\star$	$N_9^\star$	$\sum N_i^\star$
A	CMGT	27	38	42	44	44	44	42	38	27	346
	GG	26	39	42	44	44	44	42	39	26	346
	our algorithm	26	39	42	44	44	44	42	39	26	346
B	CMGT	30	39	39	37	37	49	58	48	34	371
	GG	29	40	39	37	37	49	58	48	34	371
	our algorithm	29	40	39	37	38	49	58	47	34	371
C	CMGT	35	48	48	52	60	58	57	41	34	433
	GG	34	49	47	53	60	59	57	40	34	433
	our algorithm	35	49	47	52	60	59	58	40	33	433
D	CMGT	40	37	28	25	30	55	39	34	29	317
	GG	41	38	27	25	29	56	38	34	29	317
	our algorithm	41	38	27	25	30	55	39	34	29	318
E	CMGT	73	70	56	42	31	22	14	4	1	313
	GG	73	70	56	42	31	22	13	4	4	315
	our algorithm	72	71	56	42	31	22	13	4	4	315
F	CMGT	1	4	14	22	31	42	56	70	73	313
	GG	4	4	13	22	31	42	56	70	73	315
	our algorithm	4	4	13	22	31	42	56	71	72	315

profit error graph, for instance, may not necessary be the same as the ith case in the production rate error graph.) The average error of each type is also provided. In particular, in 94 out of the 200 cases, the optimal buffer sizes from the algorithm and the surface search are the same, and therefore the three types of error in these 94 cases are 0. In addition, the average profit error, the average production rate error, and the average maximum buffer error of these 200 cases are .0077%, .0063%, and 3.02%, respectively.

The three types of errors for the 200 six-machine lines are illustrated in Figure 4-10. In particular, in 108 out of the 200 cases, the optimal buffer sizes from the algorithm and the surface search are the same, and therefore the three types of errors in these 108 cases are 0. In addition, the average profit error, the average production rate error, and the average maximum buffer error of these 200 cases are .0019%, .0053%, and 2.85%, respectively.

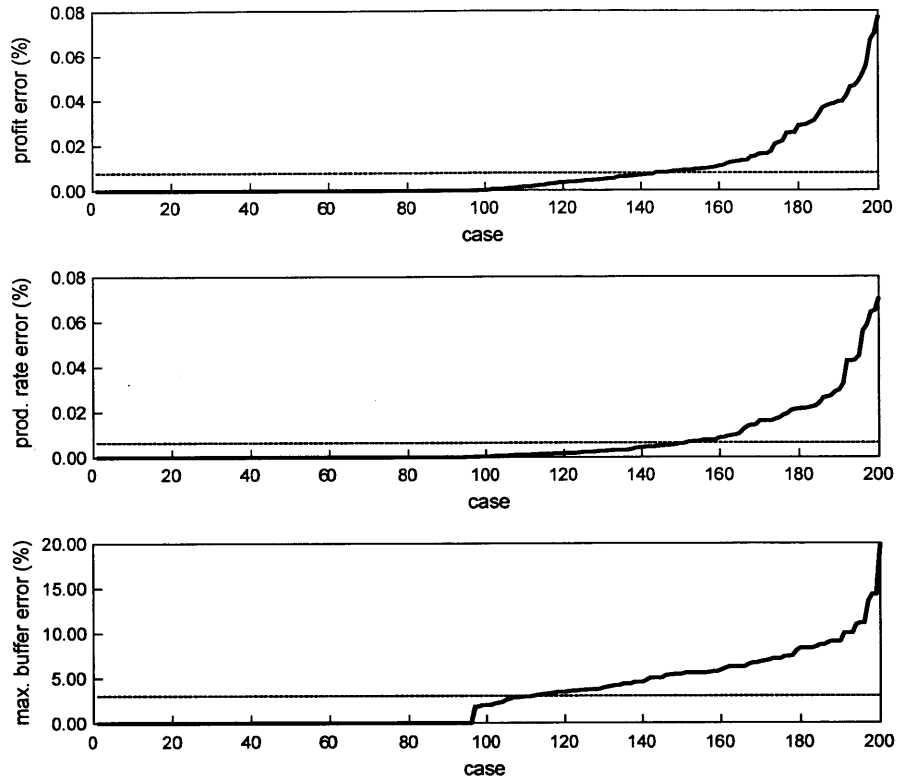


Figure 4-9: Results of two hundred randomly generated deterministic single failure mode four-machine lines

The three types of errors for the 200 eight-machine lines are illustrated in Figure 4-11. In 94 out of the 200 cases, the optimal buffer sizes from the algorithm and the surface search are the same, and therefore the three types of errors in these 94 cases are 0. In addition, the average profit error, the average production rate error, and the average maximum buffer error of these 200 cases are .0006%, .0064%, and 2.35%, respectively.

In these 600 examples, although buffer size errors can be large, the errors in J and P are always small. These numerical results show the accuracy and reliability of the proposed algorithm for production line profit maximization.

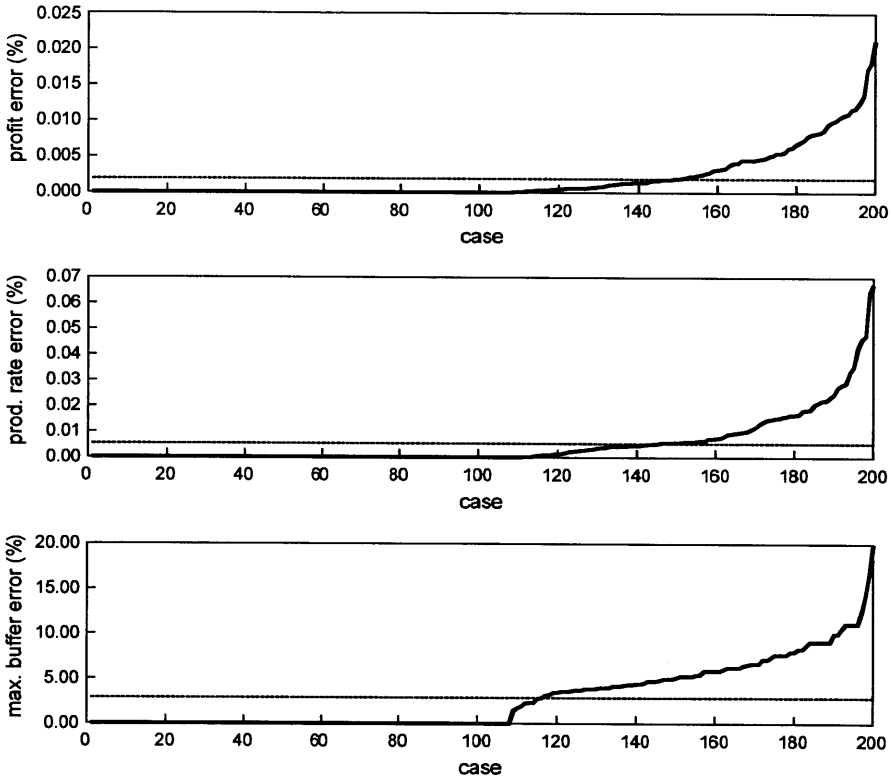


Figure 4-10: Results of two hundred randomly generated deterministic single failure mode six-machine lines

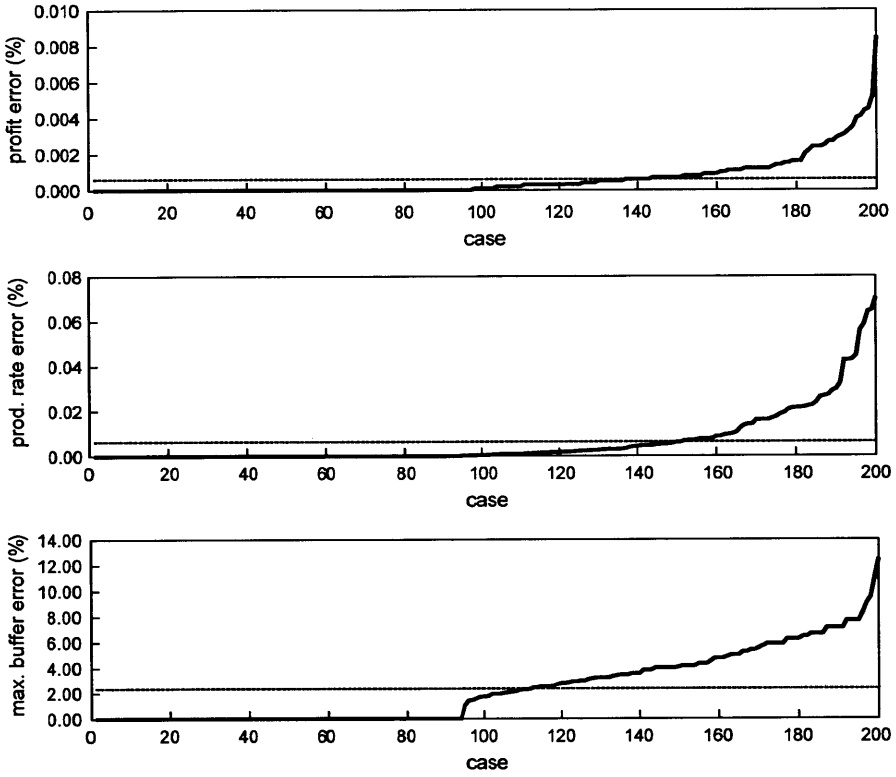


Figure 4-11: Results of two hundred randomly generated deterministic single failure mode eight-machine lines

4.4 Numerical Results for the Other Two Line Models

In this section, we extend the proposed algorithm for production line profit maximization to the other line models considered in this thesis. The two models are the deterministic multiple failure mode line model of Tolio and Matta (1998) and the continuous multiple failure mode line model of Levantesi et al. (2003).

4.4.1 The Deterministic Multiple Failure Mode Line Model

Tolio and Matta (1998) develop a decomposition method for evaluation of deterministic production lines with multiple machine failure modes. This model differs from the previous deterministic line model in that it allows machines to have more than one failure mode. The failure and repair probabilities of the jth failure mode of Machine M_i are denoted by p_{ij} and r_{ij}, respectively. The decomposition method makes use of the evaluation of two-machine lines with multiple failure modes developed by Tolio and Gershwin (1996) or a later version Tolio et al. (2002). As a reminder, we treat $\mathbf{N}$ as continuous variables and conduct a gradient method to solve the unconstraint problem (4.4). Therefore, we provide a continuous variable version of the two-machine line evaluation in Appendix C, which enables us to evaluate the production rate and the average inventory with non-integer buffer sizes.

For the deterministic multiple failure mode production line model of Tolio and Matta (1998), Colledani and Tolio (2005) develop an algorithm that solves problem (4.31). In particular, they provide a real case study to show how their algorithm is used to support the reconfiguration of a real system that produces armature spiders for electrical engines (see Colledani and Tolio 2005 for details). Here, we compare our algorithm with theirs by abstracting the machine parameters from that case without describing the specific manufacturing process. Machine parameters are listed in Table 4.18. The target production rate of the line is $\hat{P} = .68$. The results are summarized in Table 4.19, which indicates that both algorithms find the same optimal solution.

Table 4.18: Machine parameters of the system of Colledani and Tolio (2005)

machine	M_1	M_2	M_3	M_4	M_5
r_{i1}	.288	.09	.074	.079	.379
p_{i1}	.008	.003	.02	.013	.003
r_{i2}	.225	.24	.021	.9	.06
p_{i2}	.012	.008	.002	.114	.001

Table 4.19: Comparison of algorithms on the system of Colledani and Tolio (2005)

method	$N_1^\star$	$N_2^\star$	$N_3^\star$	$N_4^\star$	$\sum N_i^\star$	$P(\mathbf{N}^\star)$
Colledani and Tolio (2005)	5	13	38	3	59	.6804
our algorithm	5	13	38	3	59	.6804

In addition to comparing our algorithm with Colledani and Tolio (2005), we provide numerical experiments for 200 randomly generated five-machine production lines. These lines are generated according to the method of Gershwin (2011). In all these lines, each machine has two failure modes, where $r_{ij}/(r_{ij} + p_{ij})$ is between .909 and .952. The target production rate is $\hat{P} = .8$, while the revenue coefficient A is 3000.

The three types of errors for the 200 multiple failure five-machine lines are illustrated in Figure 4-12. Again, for each type error in Figure 4-12, we rank the three types of errors according to their corresponding ascending orders respectively. In 81 out of the 200 cases, the optimal buffer sizes from the algorithm and the surface search are the same, and therefore the three types of error in these 81 cases are 0. In addition, the average profit error, the average production rate error, and the average maximum buffer size error of these 200 cases are .003%, .021%, and 2.65%, respectively.

4.4.2 The Continuous Multiple Failure Mode Line Model

Levantesi et al. (2003) develop a decomposition method for evaluation of continuous production lines with multiple machine failure modes. In this model, machines produce at constant rates when not under repair or idle and are allowed to have different

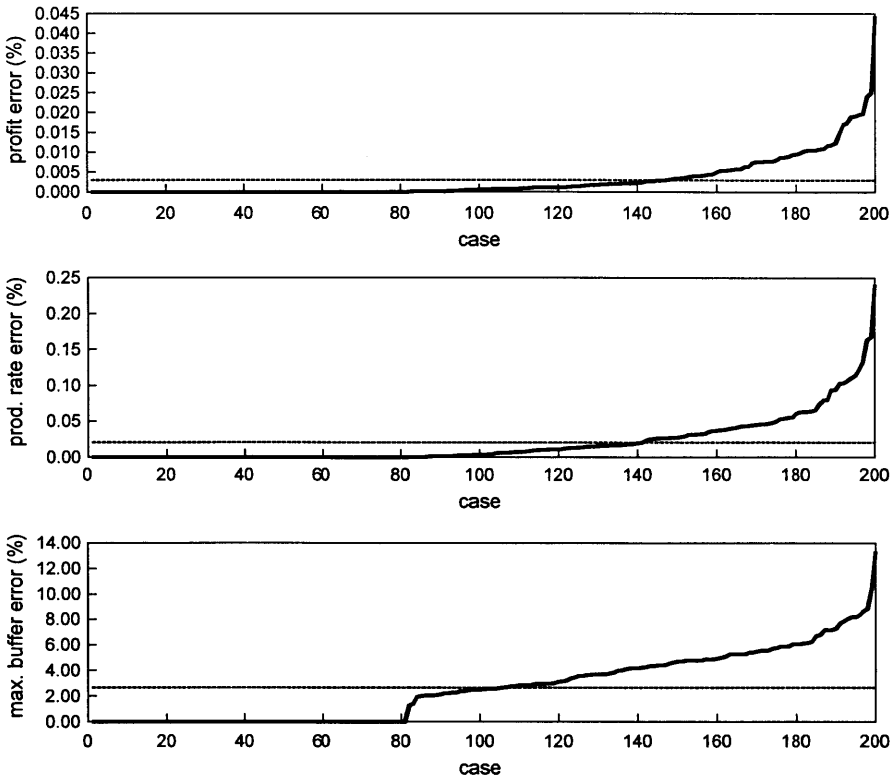


Figure 4-12: Results of two hundred randomly generated deterministic multiple failure mode five-machine lines

processing rates and more than one failure mode. In addition, the repair and failure times of any machine are exponentially distributed. The processing rate of Machine M_i is denoted by μ_i. In addition, since machines are allowed to have multiple failure mode, the failure and repair rates of the jth failure mode of Machine M_i are denoted by p_{ij} and r_{ij}, respectively. It is helpful to point out that if Machine M_i only has a single failure mode, then its isolated production rate P_i can be computed by $\mu_i r_{i1}/(r_{i1} + p_{i1})$.

The decomposition method of Levantesi et al. (2003) makes use of the evaluation of continuous two-machine lines with multiple failure modes developed by Levantesi et al. (1999a). In particular, Levantesi et al. (1999a) discuss in detail the steps to analyze, establish, and solve the model. They provide a general form of the probability density functions for all internal states and also solve the steady-state probabilities of the boundary states in the case that $\mu_u > \mu_d$, where μ_u and μ_d are the processing

rates of the upstream and the downstream machines, respectively. However, they do not discuss the solutions for the cases where $\mu_u < \mu_d$ or $\mu_u = \mu_d$, although the case that $\mu_u < \mu_d$ can be solved easily by reversing the line in which $\mu_u > \mu_d$. In addition, in Levantesi et al. (1999a), both the production rate and the average inventory are given in integral forms, which cannot be used directly for programming. Therefore, we provide the analytical solution for the case $\mu_u = \mu_d$ in Appendix D. Some discussion from the perspective of algorithm realization, including the analytical forms of the production rate and the average inventory, is also provided. The material covered in Appendix D is considered as a good complement of Levantesi et al. (1999a).

We provide numerical experiments for the continuous line model in this section. As stated in Section 4.3.5, we will be able to compare our algorithm with Schor (1995), Levantesi et al. (2001), and Tolio et al. (2009). In particular, Schor (1995) studies the continuous line model with the ADDX algorithm of Burman (1995) that is based on the continuous two-machine model of Gershwin and Schick (1980). On the other hand, Levantesi et al. (2001), Tolio et al. (2009), and our algorithm apply the decomposition of Levantesi et al. (2003) that is based on Levantesi et al. (1999a) for the continuous line model. Therefore, we shall expect slight differences between the solutions of Schor's algorithm and our algorithm as the two underlying analytical approaches for the continuous line model are different.

Table 4.20: Parameters of three five-machine lines with $r_{i1} = .1$ and $p_{i1} = .01$ of Schor (1995)

line	μ_1	μ_2	μ_3	μ_4	μ_5	$b_i, \forall i$	$c_i, \forall i$	A
1	1.03	1.01	1.02	1.00	1.04	1	1	1000
2	1.00	1.01	1.02	1.03	1.04	1	1	1000
3	1.00	1.01	1.02	1.03	1.04	0	1	1000

First, we consider three five-machine lines studied by Schor (1995) for our unconstrained problem (4.4). The line parameters are listed in Table 4.20. All machines have a single failure mode. The results for these three lines from Schor (1995) and our algorithm are summarized in Table 4.21. (Note that, although we are considering the

continuous line model, the buffer sizes are still integers. The model differs from the discrete model as it models parts as continuous flows.) Considering the different underlying approaches for the continuous line model in Schor (1995) and our algorithm, which result in (slightly) different evaluation results of both the production rate and the average buffer levels for the same buffer allocation, Table 4.21 shows that both algorithms are accurate.

Table 4.21: Result comparison of the three five-machine lines of Schor (1995)

line	method	$N_1^\star$	$N_2^\star$	$N_3^\star$	$N_4^\star$	$J(\mathbf{N}^\star)$
1	Schor (1995)	4	13	15	10	737.16
	our algorithm	4	13	14	10	735.04
2	Schor (1995)	6	14	15	8	740.78
	our algorithm	5	13	14	9	736.52
3	Schor (1995)	16	39	47	47	813.24
	our algorithm	14	38	48	47	808.31

Next, we consider four three-machine lines for problem (4.31) to compare our algorithm with the algorithm of Tolio et al. (2009). The parameters of these lines are listed in Table 4.22. The results are summarized in Table 4.23[13]. We see from Table 4.23 that, although there is small discrepancy in the optimal solutions from the algorithm of Tolio et al. (2009) and our algorithm, both algorithms are accurate in terms of the total buffer size $N_1^\star + N_2^\star$.

Next, we consider a three-machine line and a four-machine line for problem (4.31) to compare our algorithm with Levantesi et al. (2001). The two lines are studied in Levantesi et al. (2001). In addition, Levantesi et al. (2001) compare the results of their algorithm against exhaustive research.

The parameters of the three-machine line are listed in Table 4.24. All machines have a single failure mode. In particularly, we consider four target production rates. The results are summarized in Table 4.25[14]. We see from Table 4.25 that both

[13]The results of these experiments from the algorithm of Tolio et al. (2009) are provided in a unpublished manuscript (Borgh 2009b), in which the buffer sizes are non-integers. Therefore, for comparison, we do not convert the optimal solutions of our algorithm to integers.

[14]The optimal buffer sizes of these experiments in Levantesi et al. (2001) are non-integers. Therefore, for comparison, we do not convert the optimal solutions of our algorithm to integers.

Table 4.22: Parameters of four three-machine lines

		Line 1	Line 2	Line 3	Line 4
M_1	r_{11}	.075	.077	.3	.35
	p_{11}	.007	.015	.02	.037
	μ_1	1.0	1.0	1.0	1.0
	P_1	.915	.837	.938	.904
M_2	r_{21}	.095	.95	.23	.15
	p_{21}	.008	.08	.01	.015
	μ_2	1.0	1.0	1.0	0.9
	P_2	.922	.922	.958	.818
M_3	r_{31}	.078	.47	.78	.4
	p_{31}	.004	.03	.06	.02
	μ_3	1.0	1.0	1.0	1.0
	P_3	.951	.940	.929	.952
$\hat{P}$		.89	.82	.90	.815

Table 4.23: Result comparison of four three-machine lines

line	method	$N_1^\star$	$N_2^\star$	$\sum N_i^\star$	$P(\mathbf{N}^\star)$
1	Tolio et al. (2009)	50.7182	31.0056	81.7237	.8900
	our algorithm	52.2267	29.4690	81.6957	.8900
2	Tolio et al. (2009)	13.6483	4.5634	18.2116	.8199
	our algorithm	13.0866	5.1403	18.2269	.8200
3	Tolio et al. (2009)	4.9826	5.8432	10.8257	.9000
	our algorithm	4.6476	6.1650	10.8126	.9000
4	Tolio et al. (2009)	17.0466	11.3032	28.3498	.8150
	our algorithm	18.0194	10.0016	28.0210	.8150

algorithms are accurate in terms of the optimal buffer allocation as well as the total buffer size $N_1^\star + N_2^\star$, as compared to exhaustive research.

Table 4.24: Parameters of the three-machine line of Levantesi et al. (2001)

	M_1	M_2	M_3
r_{i1}	.350	.150	.400
p_{i1}	.037	.015	.020
μ_i	1.0	1.0	1.0
P_i	.904	.909	.952

Table 4.25: Result comparison of the three-machine line of Levantesi et al. (2001)

case	$\hat{P}$	method	$N_1^\star$	$N_2^\star$	$\sum N_i^\star$	$P(\mathbf{N}^\star)$
1	.8700	Levantesi et al. (2001)	14.56	5.86	20.42	.8700
		our algorithm	14.53	5.86	20.40	.8700
		exhaustive research	13.97	6.50	20.47	.8700
2	.8800	Levantesi et al. (2001)	22.72	8.99	31.71	.8800
		our algorithm	22.73	8.91	31.63	.8800
		exhaustive research	22.44	9.20	31.64	.8800
3	.8900	Levantesi et al. (2001)	40.45	14.47	54.92	.8900
		our algorithm	41.23	13.74	54.97	.8900
		exhaustive research	39.69	15.30	54.99	.8900
4	.9000	Levantesi et al. (2001)	112.48	28.20	140.68	.9000
		our algorithm	112.40	28.20	140.60	.9000
		exhaustive research	113.41	27.29	140.70	.9000

The parameters of the four-machine line are listed in Table 4.26. Again, all machines have a single failure mode. In particularly, we consider five target production rates. The results are summarized in Table 4.27. We see from Table 4.27 that both algorithms are accurate in terms of the optimal buffer allocation as well as the total buffer size.

Table 4.26: Parameters of the four-machine line of Levantesi et al. (2001)

	M_1	M_2	M_3	M_4
r_{i1}	.091	.0526	.0833	.1429
p_{i1}	.050	.006	.0454	.0454
μ_i	1.0	1.0	1.0	1.0
P_i	.645	.898	.647	.759

Finally, for the continuous multiple failure mode production line model, we provide numerical experiments for 200 randomly generated five-machine production lines. These lines are generated according to the method of Gershwin (2011). In all these lines, each machine has two failure modes, where $r_{ij}/(r_{ij} + p_{ij})$ is between .909 and .952 and μ_i is between .95 and 1.05 for a given machine. The target production rate is $\hat{P} = .8$, while the revenue coefficient A is 3000.

Table 4.27: Result comparison of the four-machine line of Levantesi et al. (2001)

case	$\hat{P}$	method	$N_1^\star$	$N_2^\star$	$N_3^\star$	$\sum N_i^\star$	$P(\mathbf{N}^\star)$
1	.4950	Levantesi et al. (2001)	5.81	7.51	4.71	18.03	.4953
		our algorithm	5.74	7.47	4.69	17.90	.4950
		exhaustive research	4.70	8.20	5.10	18.00	.4950
2	.5300	Levantesi et al. (2001)	9.91	12.11	8.31	30.33	.5301
		our algorithm	9.79	12.07	8.35	30.21	.5300
		exhaustive research	9.20	12.70	8.40	30.30	.5300
3	.5650	Levantesi et al. (2001)	16.61	19.41	14.14	50.16	.5651
		our algorithm	16.41	19.37	14.20	49.98	.5650
		exhaustive research	16.00	19.00	15.20	50.20	.5650
4	.6000	Levantesi et al. (2001)	29.91	33.01	24.68	87.60	.6000
		our algorithm	29.26	33.13	24.91	87.30	.6000
		exhaustive research	29.20	33.20	25.10	87.50	.6000
5	.6400	Levantesi et al. (2001)	106.10	93.61	62.77	262.48	.6400
		our algorithm	91.32	94.36	69.40	255.08	.6400
		exhaustive research	100.00	89.00	69.90	258.90	.6400

The three types of errors for the 200 continuous multiple failure five-machine lines are illustrated in Figure 4-13. In 112 out of the 200 cases, the optimal buffer sizes from the algorithm and the surface search are the same, and therefore the three types of errors in these 112 cases are 0. In addition, the average profit error, the average production rate error, and the average maximum buffer size error of these 200 cases are .0020%, .0038%, and 1.95%, respectively.

4.5 Summary

In this chapter, we present an accurate, fast, and reliable algorithm for maximizing profits through buffer space optimization for production lines. In the cost function, we consider both buffer space cost and average inventory cost and assign different cost coefficients to different buffers. In addition, we include a production rate constraint in our problem. A nonlinear programming approach is adopted to solve the problem. The algorithm is proved theoretically by the KKT conditions of nonlinear programming.

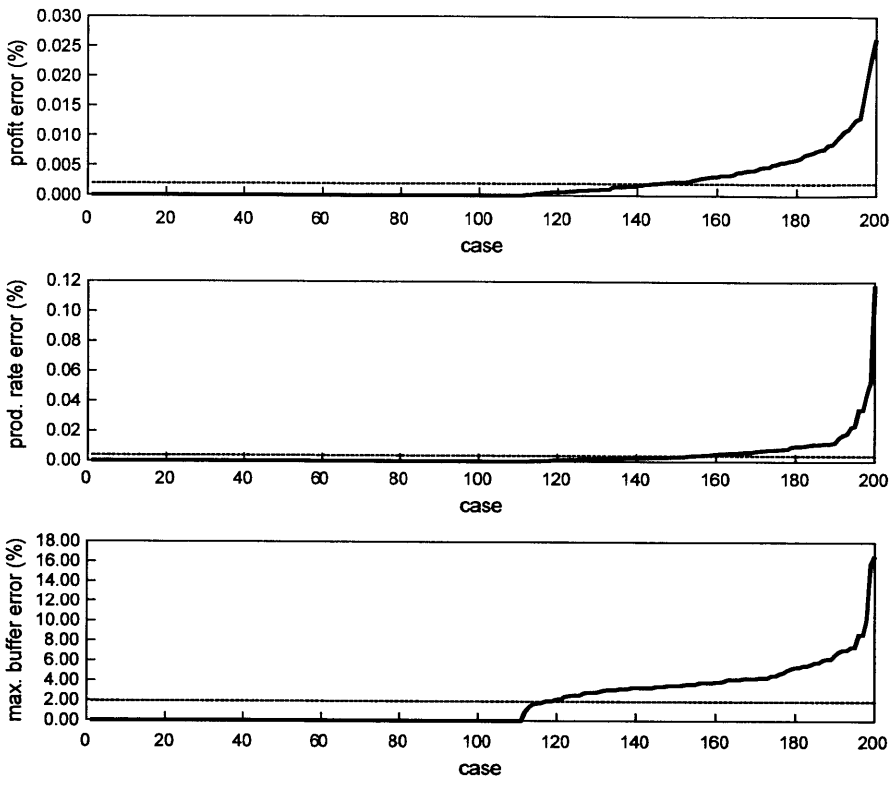


Figure 4-13: Results of two hundred randomly generated continuous multiple failure mode five-machine lines

The proposed algorithm can be applied to the three production line models under consideration. They are the deterministic single failure mode line model of Gershwin (1987a), (1994), the deterministic multiple failure mode model of Tolio and Matta (1998), and the continuous multiple failure mode model of Levantesi et al. (2003). To study the accuracy and efficiency of the algorithm, we provide numerical experiments on randomly generated lines. In addition, the algorithm is compared with existing algorithms for solving a special case (i.e., Problem (4.31)) of the constrained problem. In particular,

- for the deterministic single failure mode model, we compare our algorithm to Schor (1995), Gershwin and Goldis (1995), and Colledani et al. (2003) for solving (4.31),

- for the deterministic multiple failure mode model, we compare our algorithm

with Colledani and Tolio (2005) for solving (4.31),

- and finally, for the continuous multiple failure mode model, we compare our algorithm with Schor (1995), Levantesi et al. (2001), and Tolio et al. (2009) for solving (4.31).

All these numerical experiments studied in this chapter show the accuracy and efficiency of the proposed algorithm. The algorithm will be extended to single closed-loop systems in Chapter 6 and to production lines with an additional maximum part waiting time constraint in Chapter 7. Some valuable insights about optimal design of long lines are discussed in Chapters 8 and 9.

Chapter 5

Modification of Single Loop System Evaluation

5.1 Problem and Motivation

A *closed-loop production system*, or *loop*, is a system in which a constant amount of material flows through a single fixed cycle of work stations and storage buffers (Gershwin and Werner 2007). This type of system appears frequently in factories. Manufacturing processes which utilize pallets or fixtures can be viewed as loops since the number of pallets/fixtures that are in the system remains constant. Similarly, control policies such as CONstant Work-In-Process, or CONWIP, (Spearman et al. 1990) and Kanban (Monden 1998) create conceptual loops by imposing a limitation on the number of parts that can be in the system at any given time. The difference between CONWIP and Kanban is that CONWIP pulls a job into the beginning of the line and the job goes with a card through all workstations, while Kanban provides tighter control over the material flow through individual workstations (Hopp and Spearman 2000).

Figure 5-1 shows a k-machine k-buffer loop system. Assume that there are a constant number of pallets traveling in the system. In addition, Machine M_1 is the first machine of the system, while Machine M_k is the last machine of the system. Whenever a new part tries to enter the system at M_1, we need to check first if M_1 is

blocked or not. But just as importantly, we have to check if there are pallets available in Buffer B_k. If M_1 is not blocked and B_k is not empty, then a new part is allowed to enter the system at M_1 and it will travel together with the pallet assigned to it (from B_k) through the entire system. After that part is produced by M_k, it leaves the system while the pallet associated with it goes to B_k again waiting for future parts. This is how such a closed-loop system differs from a traditional series transfer line. In other words, whether a new part can enter the system or not depends on whether there are free pallets available. If all pallets are occupied by parts being operated at machines in the system, then B_k will be empty and no more parts will be allowed to enter the system. This is also how such a system or a CONWIP policy controls the total number of parts in the system. Consequently, a loop system or the CONWIP policy are ways of reducing work-in-process inventory.

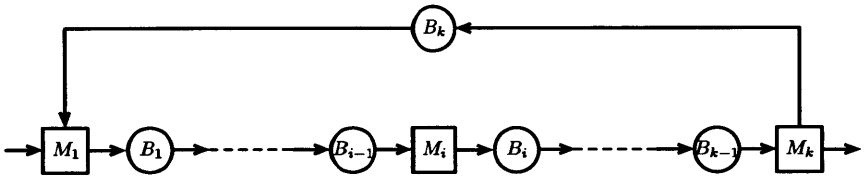


Figure 5-1: An example of a closed-loop system

Loop systems and CONWIP policies have many applications. Ip et al. (2007) compare the single loop and multiple loop CONWIP production control systems for a lamp assembly production line producing different kinds of products with discrete distribution processing time and demand. Resano Lázaro and Luis Pérez (2008) and Resano Lázaro and Luis Pérez (2009) study networks of closed loops in automobile assembly lines. Li et al. (2010) apply multi-CONWIP in semiconductor assembly and test factory. Rodzewicz et al. (2010) introduce the CONWIP concept to ship repair through the completion of a discrete event simulation. In addition, the concept of CONWIP has been applied to supply chain management as well (see Ovalle and Marquez 2003). Takahashi et al. (2005) apply Kanban, CONWIP and synchronized CONWIP to supply chains to determine the superior system.

Given the importance of loop systems, we want to study how the production rate

and the average inventories of all buffers of a loop system change as functions of the buffer sizes as well as the *loop invariant* (the constant number of pallets, or another quantity such as the number of *tokens* or *production authorization cards*, allowed in the system at any given time). However, we would like to indicate clearly that it is the total number of parts in the system, which is upper bounded by the loop invariant, that should be used to compute the production rate as well as average inventories. Suppose we consider a loop system with a constant number of pallets. The number of parts in the system may not equal the number of pallets all the time, since it is possible for the system to have free pallets in Buffer B_k while M_1 is occupied occasionally. Therefore, we assume that the total number of pallets in the system is constant and therefore it equals to the loop invariant[1]. As a result, in what follows, we use the word *part* to cover all cases (e.g., *pallet*, *token*, and *card*). It is desirable to find the optimal combination of buffer allocation and the loop invariant that satisfies the production rate target at the minimum costs in terms of buffer space and average inventory. Therefore, we want to extend the buffer allocation optimization algorithm developed in Chapter 4 for tandem lines to closed-loop systems. There are a number of studies regarding the evaluation of such systems, however little work has been dedicated to the optimization of loop systems. Evaluation results provide average production rate as well as the average inventory level of each buffer in the system, which serve as prerequisites of the optimization. On the other hand, optimization depends highly on the accuracy of evaluation results given a set of machine and buffer parameters, as well as the smoothness of the evaluation results as a result of continuous changes in the input system parameters.

Onvural and Perros (1987) study closed cyclic queueing networks and demonstrate that the production rate of a closed-loop system is a function of the number of parts in the system. Tolio and Gershwin (1998) present a decomposition approach for estimating the production rate of a closed queueing network with exponential servers, finite buffer capacity and a blocking after service discipline. Each subsystem is analyzed as an $M/M/1/C_i + 1$ queue with state-dependent arrival and service rates. Frein et al.

[1]This is a common assumption in the evaluation of loop systems.

(1996) propose the first approximate analytical method for evaluating the performance of closed-loop systems with unreliable machines and finite buffers. However, it does not treat the correlation that exists among the numbers of parts in the buffers. As a result, the method is only accurate for large loops with populations that are neither too large or too small. Maggio (2000) and Maggio et al. (2009) present a new decomposition method based on Tolio decomposition (Tolio and Matta 1998). This new decomposition method considers the correlation among the numbers of parts in the buffers, therefore it provides more accurate results. However, due to its complexity, it is not practical for systems with more than three machines. Werner (2001) and Gershwin and Werner (2007) simplify and extend the decomposition method mentioned above, and developed an algorithm[2] that can evaluate loops with any number of machines efficiently and accurately. Zhang (2006) extends Werner's algorithm to the evaluation of multiple loop systems. We will comment more on both Werner's and Zhang's algorithms in Section 5.2. One paper that deals with the optimization of the profit of loop systems is Helber et al. (2009). It adopts a linear programming algorithm to evaluation closed-loop systems and then studies the profit of the system as a function of the CONWIP level. However, it does not consider buffer spaces as decision variables. For other works, see Akyildiz (1988), Lim and Meerkov (1993), Bonvik et al. (1997), Bonvik et al. (2000), Balsamo et al. (2001), Kim et al. (2002), Bozer and Hsieh (2005), Biller et al. (2009), and Mhada and Malhamé (2011).

The purpose of this chapter is to discuss how to improve the evaluation accuracy of single closed-loop systems towards the ultimate goal of optimization. We will further extend the optimization algorithm of Chapter 4 to closed-loop systems in Chapter 6. The rest of this chapter is organized as follows. We first comment on Werner's algorithm for closed-loop system evaluation and briefly introduce Zhang's algorithm for multiple loop system evaluation in Section 5.2. By providing numerical evidence, we explain the necessity for improvement of loop evaluation for the purpose of optimization. In Section 5.3, we extend the evaluation of closed-loop systems to single *open-loop systems*, in which the total number of parts either within the

[2]For the rest of this chapter, we refer to this algorithm as Werner's algorithm.

entire system *or* within a portion of the system is controlled by the loop invariant. Next, two necessary modifications on the existing evaluation algorithm are analyzed in great detail in Section 5.4, followed by numerical experiments in Section 5.5. The improvement of loop evaluation will be demonstrated by those experiments.

5.2 Related Algorithms for Loop Evaluation and Necessity for Improvement

5.2.1 Review of Werner's Algorithm

Gershwin and Werner (2007) and Zhang (2006) develop efficient and accurate evaluation algorithms for single closed-loop systems and multiple loop systems, respectively. However, since our ultimate goal is to extend the optimization algorithm for transfer lines (studied in Chapter 4) to single loop systems, we care about not only the accuracy of evaluation, but also the smoothness of the evaluation with respect to changes in the input parameters. As we will show later in this section, both Werner's and Zhang's algorithms, although accurate, exhibit undesirable discontinuities of evaluation results.

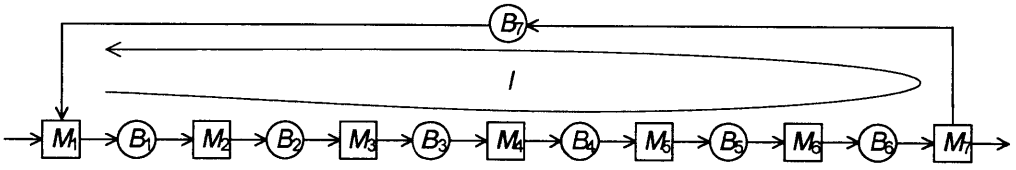


Figure 5-2: A closed-loop system

Figure 5-2 provides an example of the kind of closed-loop system that Werner studied. In this example, all seven machines and seven buffers of the line are controlled by the loop. As explained in Section 5.1, the loop invariant (denoted by I in the remainder of this chapter) is the constant number of parts that are allowed in the system at any given time. Therefore, we have

$$I = n_1(t) + n_2(t) + \cdots + n_7(t)$$

where $n_i(t), i = 1, \cdots, 7$ is the inventory level of Buffer B_i at time t. The loop invariant limits the total number of parts in the system, affects the behavior of blocking and starvation, and therefore controls the production rate and buffer levels. Gershwin and Werner (2007) developed a decomposition approach by considering the relationship among the numbers of parts in the buffers. It provides evaluation results in terms of the production rate of the system as well as the average inventory of each buffer efficiently for single closed-loop systems of any size.

We provide a brief review of the decomposition approach of Werner's algorithm. Decomposition (Gershwin 1987a, Tolio and Matta 1998, and other relevant literatures in Section 1.2.2) approximates complex systems as a set of two-machine one-buffer building blocks. Since there are analytical solutions for two-machine one-buffer building blocks based on Markov chain models, once we find parameters for those two-machine one-buffer building blocks, we will be able to evaluate the original system.

Different decomposition approaches for transfer lines are studied in detail in all those relevant literatures mentioned in Section 1.2.2. Gershwin (1991) and Gershwin and Burman (2000) applied the decomposition approach to analyze assembly and disassembly systems. Gershwin and Werner (2007) and Zhang (2006) adopted Tolio's decomposition (Tolio and Matta 1998) for loop system evaluation. Using Figure 5-3, we illustrate the idea of decomposition very briefly. For details, refer to the references mentioned above.

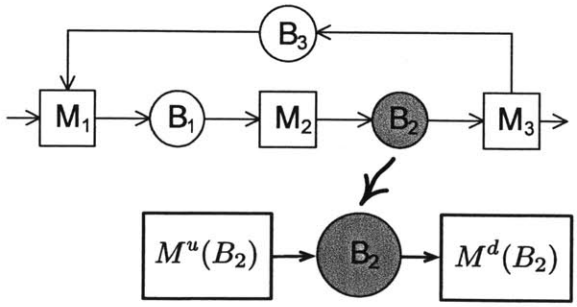
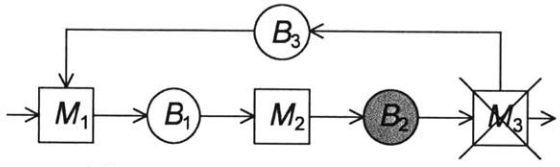


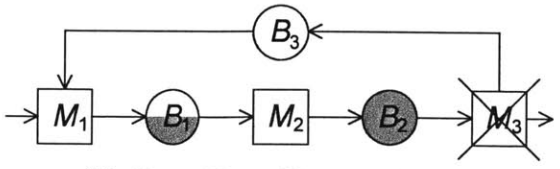
Figure 5-3: The decomposition approach of loop evaluation

Figure 5-3 shows a closed-loop system that has three machines and three buffers.

Consider the material inflow to and outflow from Buffer B_2. We study it by imagining that this buffer is in a two-machine one-buffer line, where $M^u(B_2)$ denotes the upstream pseudo-machine and $M^d(B_2)$ denotes the downstream pseudo-machine. The key to the decomposition approach is to choose parameters for both upstream and downstream pseudo-machines such that the material flow behavior through B_2 in the two-machine one-buffer line is approximately the same as that in the original loop. The upstream pseudo-machine, for instance, has one up state and several down states. When it is up, it produces a part in each time unit if it does not fail. It can fail in failure mode i with probability $p^{u_i}(B_2)$. If it is in down state i, it can get repaired with probability $r^{u_i}(B_2)$ in each time unit. We need to determine all failure modes for both upstream and downstream pseudo-machines. Gershwin and Werner (2007) indicates that all machine failures in the original system which could cause B_2 to be empty should be categorized as the failure modes of the upstream pseudo-machine $M^u(B_2)$. This is because if the upstream pseudo-machine fails for a long time, B_2 can become empty. Similarly, all machine failures in the original system which could cause B_2 to be full are categorized as the failure modes of the downstream pseudo-machine $M^d(B_2)$. In other words, B_2 will be starved due to the failure modes of its upstream pseudo-machine and be blocked due to the failure modes of its downstream pseudo-machine.



(a) $N_1 = N_2 = N_3 = 10, I = 10$



(b) $N_1 = N_2 = N_3 = 10, I = 14$

Figure 5-4: Demonstration of the threshold in loop evaluation

Consider the three-machine three-buffer closed-loop system in Figure 5-4(a), where the sizes of the three buffers are $N_1 = N_2 = N_3 = 10$ and the loop invariant $I = 10$. Suppose that Machine M_3 fails for a long time. As a result, parts will accumulate at Buffer B_2 and B_2 will be full, while buffers B_1 and B_3 will be empty. Therefore, according to the analysis above, the failure of M_3 can be considered as the failure of the upstream pseudo-machine $M^u(B_1)$ of B_1.

However, there is difficulty in evaluating a two-machine one-buffer building block because of the presence of buffer thresholds (Gershwin and Werner 2007 and Zhang 2006). Consider the loop system in Figure 5-4(b), where the loop invariant $I = 14$. Suppose we look at the two-machine one-buffer building block $M^u(B_1) - B_1 - M^d(B_1)$ with respect to B_1. For the discussion below, let n_i be the buffer level of B_i. Consider the two cases below:

- if $n_1 \leq 4$ and $M^d(B_1)$ is down, the failure of $M^d(B_1)$ can be due to either M_2 or M_3. This is because, since the loop invariant $I = 14$ and $n_1 \leq 4$, then $n_2 + n_3 \geq 10$. So it is possible for B_2 to be full (i.e., $n_2 = N_2 = 10$) if M_3 fails. A full B_2 then causes M_2 to be blocked. In this case, the failure of $M^d(B_1)$ is due to the failure of M_3. On the other hand, the failure of M_2 can cause $M^d(B_1)$ to be down as well. Therefore, in this case, the failure of $M^d(B_1)$ can be due to either M_2 or M_3 and its repair probability is either r_2 or r_3.

- if $n_1 > 4$ and $M^d(B_1)$ is down, the failure of $M^d(B_1)$ must be due to M_2 and not M_3. This is because in this case $n_1 > 4$. Since $I = 14$, then $n_2 + n_3 < 10$. It is not possible for B_2 to be full if M_3 fails. If it did, M_2 would not be blocked and therefore $M^d(B_1)$ could not be down. Therefore the failure of $M^d(B_1)$ can only be due to M_2 and its repair probability is r_2.

According to the analysis above, we see that it is possible for the repair probability of $M^d(B_1)$ to be a function of the buffer level n_1. This is undesirable as it makes the evaluation of two-machine one-buffer building blocks very complicated. Therefore, 4 (which is determined by the loop invariant and the size of each buffer in the loop) is the threshold of B_1 and we need to eliminate the thresholds for Buffer B_1 as well

as other buffers in the system. To resolve this issue, perfectly reliable machines are introduced in Werner's algorithm.

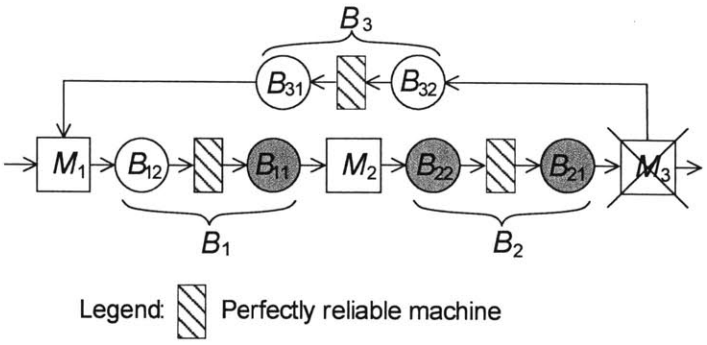


Figure 5-5: A modified closed-loop system after elimination of buffer thresholds

For the loop shown in Figure 5-4(b), we have indicated that the failure of M_3 creates threshold in B_1. With a similar analysis, it can be seen that the failures of M_1 and M_2 create thresholds in buffers B_2 and B_3, respectively. To eliminate these thresholds, we break up buffers B_1, B_2, and B_3 by inserting a (hypothetically) perfectly reliable machine in each of them, and then analyze the modified loop (Figure 5-5). For instance, we replace Buffer B_1 by a upstream Buffer B_{12}, a perfectly reliable machine, and a downstream buffer B_{11}. The size of B_{12} is 6(= N_1−threshold) while the size of B_{11} is 4 (=threshold). B_2 and B_3 are modified accordingly. After eliminating the threshold, we have six instead of three buffers. More importantly, any machine failure can cause a given buffer to be either full or empty, but not partially full. Therefore, we will be able to find the parameters for pseudo-machines of all buffers in the modified loop. Then the decomposition approach developed by Tolio and Matta (1998) is adopted in Werner's algorithm to evaluate the loop. In the decomposition, the analytical solutions developed by Tolio et al. (2002) is used to evaluate two-machine one-buffer building blocks.

5.2.2 Single Open-Loop Systems

One simple extension of a single closed loop system is a single open-loop system (see Figure 5-6 for an example). Such a system allows portion of the system to be controlled by the loop invariant. For instance, if a portion of the production line is in a clean room environment, it is desirable to control the total inventory in that portion of the line due to the expensive inventory holding cost as well as buffer space cost in the clean room environment. On the other hand, if the fixtures or pallets structure only applies to part of a transfer line, we shall also expect that part of the line to be controlled by a constant work-in-process inventory. Therefore, as compared to a single closed-loop system, a single open-loop system is a more general case. A single closed-loop system can be considered as a special case of a single open-loop system where all machines and buffers of the system are controlled by the loop invariant.

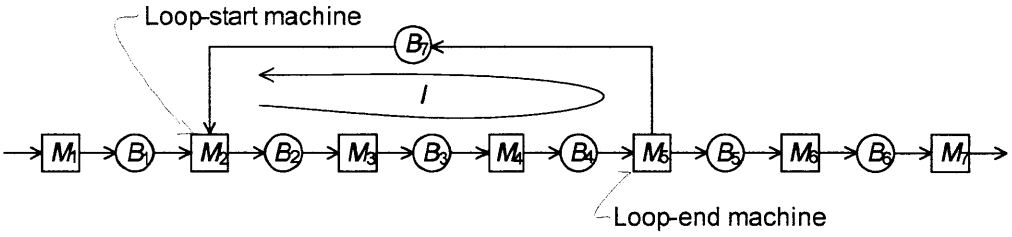


Figure 5-6: A single open-loop system

In particular, in Figure 5-6, the loop structure is formed by connecting Machines M_5 and M_2 by B_7. In this case, M_1 is the upstream machine of the entire loop, while M_6 and M_7 are the downstream machines of the entire loop. For convenience, for the rest of this chapter, we refer to first machine involved in the loop structure as the *loop-start machine*, while the last machine involved in the loop structure as the *loop-end machine*. Other machines inside in the loop are called *inner loop machines*. In Figure 5-6, M_2 is the loop-start machine and M_5 is the loop-end machine. Machines M_3 and M_4 are inner loop machines. The total number of parts in the buffers within the loop must remain constant. Therefore

$$I = n_2(t) + n_3(t) + n_4(t) + n_7(t).$$

When Werner developed his evaluation algorithm, he focused on the closed-loop systems. As a result, his algorithm cannot be used directly to evaluate single open-loop systems. However, we have to point out that decomposition is the key to analyze both single closed-loop and open-loop systems. Therefore, Werner's algorithm can be extended easily to analyze the more general system. On the other hand, Zhang (2006) extended Werner's algorithm to production systems with multiple loop structures. Consequently, Zhang's algorithm is able to evaluate single open-loop systems. In Zhang (2006), he introduced a complicated induction algorithm based on graph theory to conduct blocking and starvation analysis, which is a prerequisite for the decomposition approach. That induction algorithm is effective to deal with multiple coupled loop structures. However, in our case where we only have one open loop, there is no need for induction. Thus, the blocking and starvation analysis for single open-loop systems can be significantly simplified. As a result, the resulting decomposition algorithm after the blocking and starvation analysis can be also simplified. We discuss the evaluation of single open-loop systems in Section 5.3.

5.2.3 The Discontinuities of Evaluation Results of Werner's Algorithm

In this section, we provide an example to show the discontinuities of evaluation results of Werner's algorithm. Figure 5-7 shows the evaluation result for a three-machine three-buffer closed-loop system. The three machines are identical with parameters $p_i = .01$ and $r_i = .1$, where p_i and r_i are the failure and repair probabilities of M_i in each time unit, respectively. In addition, the buffer sizes are $N_1 = N_2 = N_3 = 10$. We vary the loop invariant I from 4 to 26, and study the production rate of the system as a function of I. In Figure 5-7, the loop invariant I is on the horizontal axis, while the production rate is on the vertical axis. We compare the evaluation result from Werner's algorithm with simulation. The length of the simulation is 5,100,000 time steps with the first 100,000 time steps being the warm up period. We run the simulation 20 times. The standard deviation of the production rate is about 7×10^{-4}.

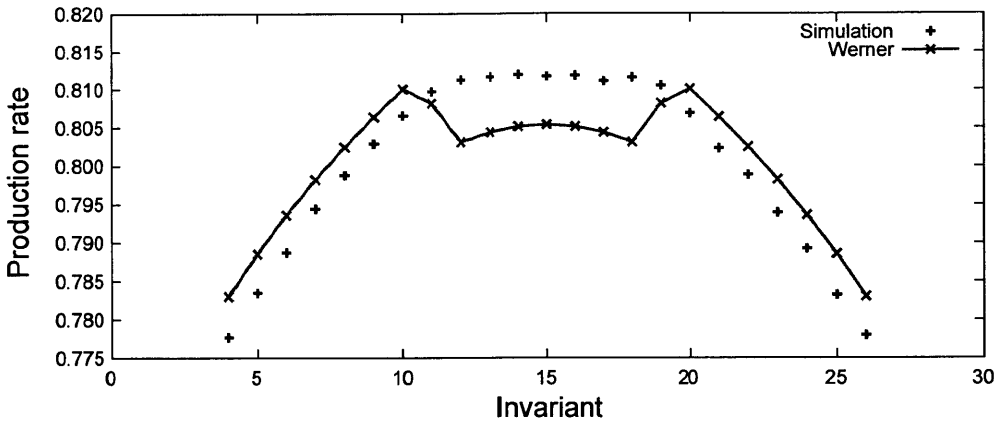


Figure 5-7: Evaluation results of Werner's algorithm – the Batman effect

Figure 5-7 demonstrates that:

1. Werner's algorithm is very accurate because the production rate difference between simulation and the decomposition method for any given invariant I is very small.

2. However, there are apparent discontinuities of the production rate[3] as I changes from 10 to 11, 11 to 12, 18 to 19, and 19 to 20.

From an optimization standpoint, these discontinuities are indeed undesirable as they will lead to inaccurate optimization results, especially when the optimization technique requires the use of gradient. These discontinuities will lead to incorrect search directions in the optimization algorithm. Consequently, in order to optimize single loop systems, we need to further improve the evaluation accuracy and eliminate the Batman effect.

In what follows, we first explain how we use the decomposition approach to analyze single open-loop systems in Section 5.3. Two potential problems in Werner's algorithm (as well as Zhang's algorithm) are identified and resolved in Sections 5.4.1 and 5.4.2, respectively. We will further explain how the Batman effect occurs due to those two problems.

[3]Due to the shape of the curve, we call it the "Batman" effect.

5.3 Evaluation of Single Open-Loop Systems

We use the decomposition approach to analyze single open-loop systems. As explained in Section 5.2.1, the critical step in the decomposition approach is to assign failure modes to the upstream and downstream pseudo-machines of each two-machine one-buffer building block. This is realized by using blocking and starvation analysis.

5.3.1 Blocking and Starvation Analysis

When we evaluate the performance of a manufacturing system by decomposition, the blocking and starvation properties of the system provide essential information for setting up the parameters of the pseudo-machines in a set of two-machine one-buffer building blocks (Zhang 2006). Recall that if a machine failure can cause the buffer to be empty, then it will be categorized as a failure mode of the upstream pseudo-machine in the two-machine one-buffer building block that contains that buffer. On the other hand, if a failure in the system can cause the buffer to be full, then it will be categorized as a failure mode of the downstream pseudo-machine. Zhang (2006) introduced a *machine failure — buffer level matrix* that summarizes the blocking and starvation analysis results. We use this concept in our single open-loop system evaluation. For instance, let us consider the system shown in Figure 5-8. It is a five-machine production system with a loop including Machines M_2, M_3, M_4, and Buffers B_2, B_3, B_5. The size of each buffer is $N_i = 20, \forall i$. The loop invariant I is 27. The machine failure — buffer level matrix is therefore

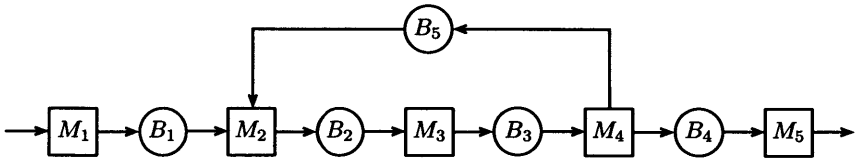


Figure 5-8: A five-machine single open-loop system

$$
\begin{array}{c} \\ M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \end{array}
\begin{array}{c}
\begin{array}{ccccc} B_1 & B_2 & B_3 & B_4 & B_5 \end{array} \\
\left[\begin{array}{ccccc}
0 & 0 & 7 & 0 & 20 \\
20 & 0 & 7 & 0 & 20 \\
20 & 20 & 0 & 0 & 7 \\
20 & 7 & 20 & 0 & 0 \\
20 & 7 & 20 & 20 & 0
\end{array}\right]
\end{array}
$$

In this matrix, buffers are placed in columns and machines are placed in rows. For the discussion below, let $X_{i,j}$ be the matrix element of row i and column j. $X_{1,1} = X_{1,2} = X_{1,4} = 0$, for instance, indicates that when Machine M_1 fails for a long time, then buffers B_1, B_2 and B_4 will be empty. In addition, B_3 will be partially full with a work-in-process inventory of seven parts, while B_5 will be full. (As a check, the total number of parts in buffers B_2, B_3, and B_5 is 27, which equals the loop invariant.) A similar explanation applies to each of other elements of the matrix. Furthermore, let us look at the first column, which indicates the limiting propagation state of Buffer B_1 given failures of different machines. It says B_1 could be empty due to failures of M_1, and it could be full due to failures of Machines M_2, M_3, M_4 and M_5. Therefore, we conclude that for the two-machine one-buffer building block that contains B_1, failures of M_1 are associated with the failures of its upstream pseudo-machine, while failures of M_2 to M_5 are associated with the failures of its downstream pseudo-machine. Therefore, the machine failure — buffer level matrix provides us essential information about how we can set up the two-machine one-buffer building block for each buffer in the original system, and how we can identify the potential failure modes of the upstream and downstream pseudo-machines[4]. As a result, the first requirement of the decomposition approach is to derive the machine failure — buffer level matrix for a given single open-loop system.

[4]In Columns 2, 3 and 5 in the example, the corresponding buffers may not be totally full due to some machine failures. For example, if M_4 or M_5 fails, B_2 will not be full. In this case, we cannot categorize failure modes of M_4 and M_5 to the building block associated with B_2 without necessary modifications. In particular, the number 7 is a threshold for B_2 and therefore we need to modify the matrix by eliminating all thresholds in all buffers. We discuss this in Section 5.3.3.

5.3.2 Five Types of Machine Failures

In order to construct the machine failure — buffer level matrix for a given single open-loop system, it is helpful to notice that there are (at most) five types of machine failures: upstream machine failure, loop-start machine failure, inner loop machine failure, loop-end machine failure, and downstream machine failure (see Figure 5-9). In Figure 5-9, M_1 is the upstream machine of the loop, M_6 and M_7 are the downstream machines of the loop, M_2 is the loop-start machine, M_5 is the loop-end machine, and M_3 and M_4 are inner loop machines. The failures of each type machine are considered as each type failure. For example, the failures of the loop-start machine are called loop-start machine failures.

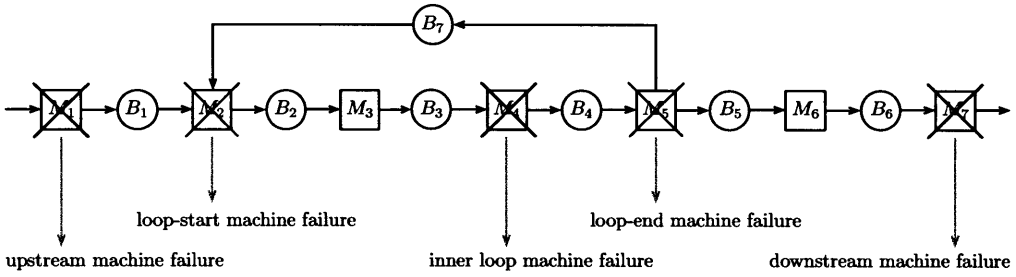


Figure 5-9: Five types of machine failures

It is helpful to point out that even for a given type of machine failure, there can be several possible scenarios of the limiting propagation state of buffers that is determined by the specific loop invariant as well as the buffer sizes. To illustrate this point, we construct four loop systems. Then, we discuss the five types of machine failures separately. We go through each type of machine failure to build up the machine failure — buffer level matrix for each of these four systems. The four loop systems under consideration are shown in Figure 5-10.

It can be seen that the four loop systems have the identical structure, where the total number of parts in Buffers B_3, B_4, B_5, and B_8 equals to the loop invariant I. In addition, all buffers in the four systems have sizes of 10 parts. The loop invariants in these four systems are 5, 15, 25, and 35, respectively. We do not specify the machine parameters as the limiting propagation state of buffers is independent of them.

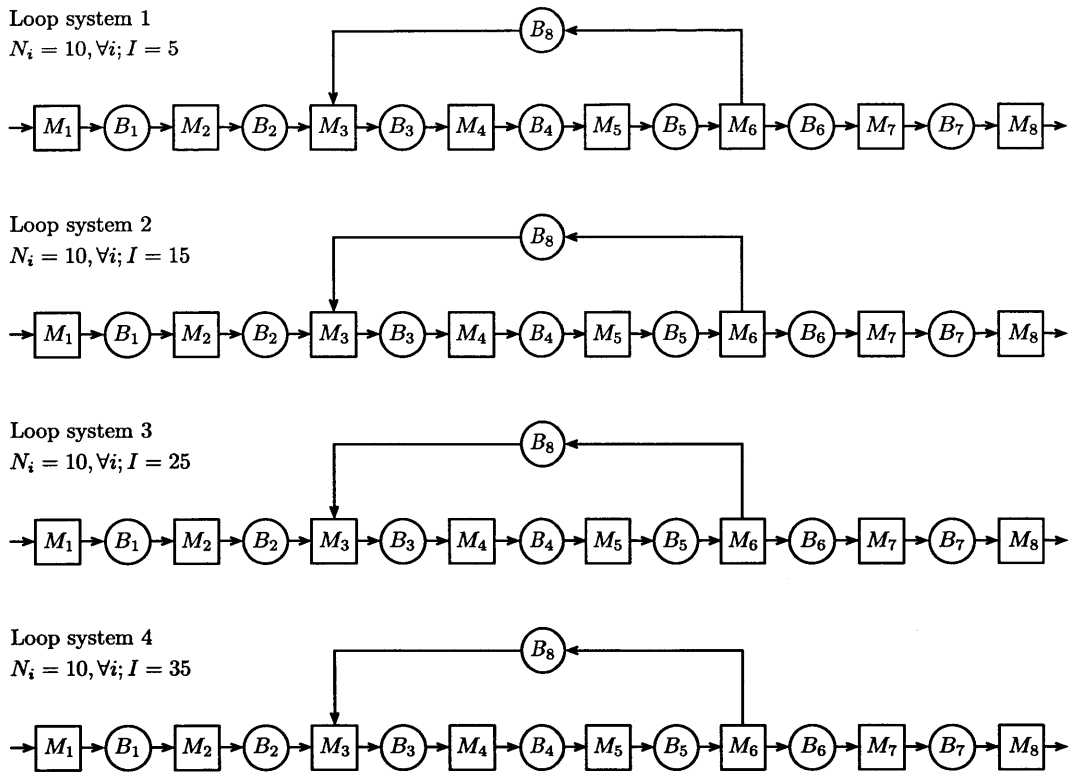
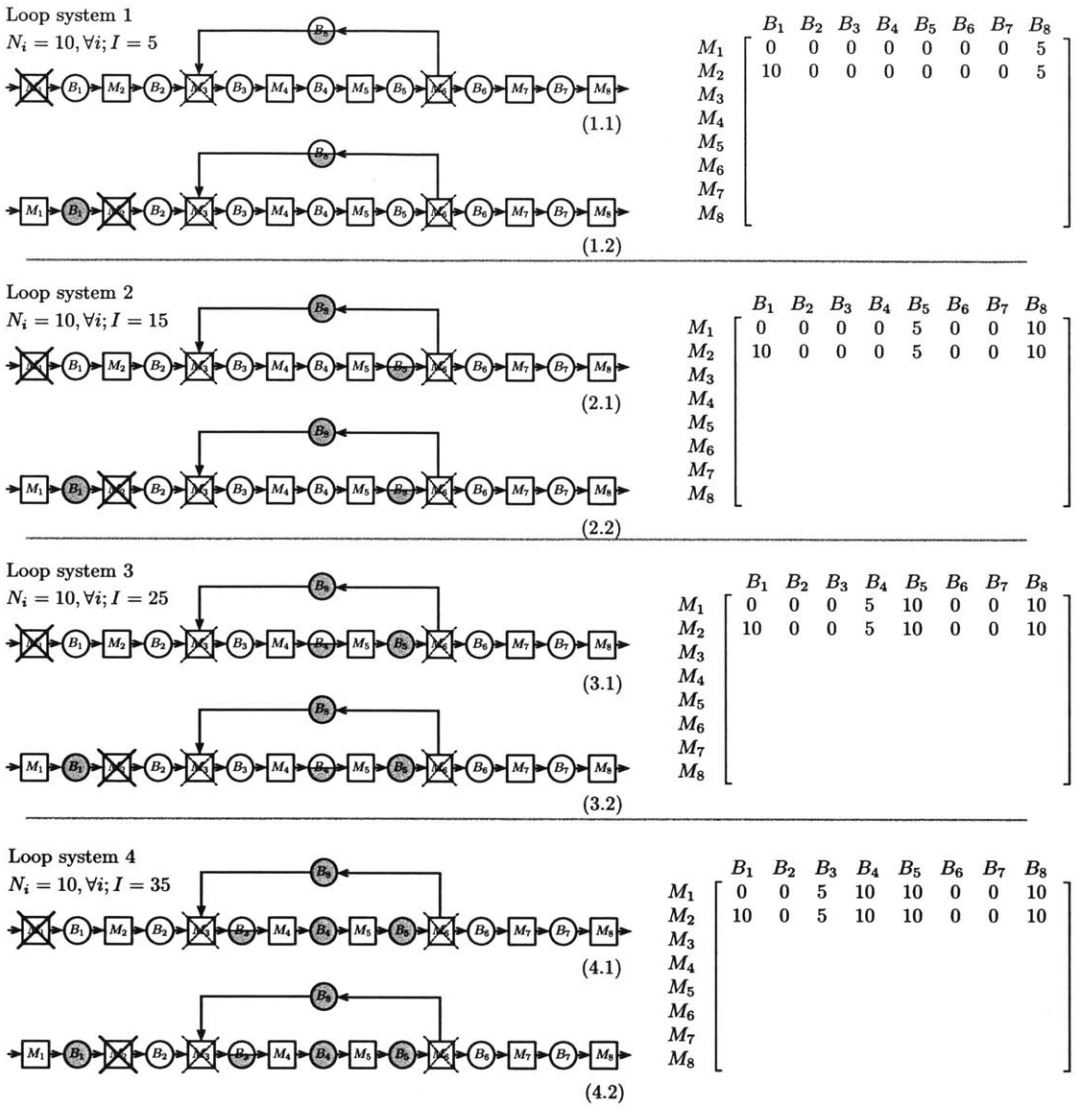


Figure 5-10: Four single open-loop systems

1. Upstream machine failure

We first examine the case where one of the upstream machines of the loop fails. Note that the failures of Machines M_1 and M_2 in each of those four systems belong to upstream machine failures. In other words, for each system, we study the limiting propagation states of buffers given that Machines M_1 and M_2 fail, respectively. Therefore, for each system, the first two rows of the machine failure — buffer level matrix can be filled. They are provided in Figure 5-11.

We first discuss the inventory levels of buffers inside the loop (i.e., Buffers B_3, B_4, B_5, and B_8). In order to determine them, we study the status of the loop-start machine, because it can be starved due to upstream machine failures. We realize that the buffer upstream of the loop-start machine (i.e., B_2) is always empty given an upstream machine failure, and therefore the loop-start machine will be starved. Since it is starved, the loop-start machine (i.e., M_3) looks to the rest of the system as if it



	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	0	0	0	0	5
M_2	10	0	0	0	0	0	0	5
M_3								
M_4								
M_5								
M_6								
M_7								
M_8								

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	0	5	0	0	10
M_2	10	0	0	0	5	0	0	10
M_3								
M_4								
M_5								
M_6								
M_7								
M_8								

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	5	10	0	0	10
M_2	10	0	0	5	10	0	0	10
M_3								
M_4								
M_5								
M_6								
M_7								
M_8								

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	5	10	10	0	0	10
M_2	10	0	5	10	10	0	0	10
M_3								
M_4								
M_5								
M_6								
M_7								
M_8								

Figure 5-11: Upstream machine failure examples

failed. A dashed cross is used in Figure 5-11 to distinguish the starvation of M_2 from the actual failure of a upstream machine. Material flow within the loop will move along the direction of the loop. Therefore, parts start accumulating at the buffer upstream of the loop-start machine (i.e., B_8). Consequently, the inventory levels of all buffers inside the loop will be determined as though the loop-start machine fails. Moreover,

1. For Loop system 1 in Figure 5-11, the loop invariant I is smaller than the size

of Buffer B_8. Therefore, all parts in the loop will accumulate in B_8 and B_8 will be partially full. As a result, other buffers within the loop, including the buffer upstream of the loop-end machine, will be empty. The empty buffer upstream of the loop-end machine will starve the loop-end machine (i.e., M_6). Since it is starved, M_6 looks to the buffers downstream of the loop (i.e., B_7 and B_8) as if it failed.

2. For Loop systems 2, 3, and 4 in Figure 5-11, the loop invariant I is larger than the size of Buffer B_8, and therefore B_8 will be full. Note that B_8 is the buffer downstream of the loop-end machine. Therefore, the loop-end machine (i.e., M_6) will be blocked. Since it is blocked, M_6 looks to the buffers downstream of the loop (i.e., B_7 and B_8) as if it failed.

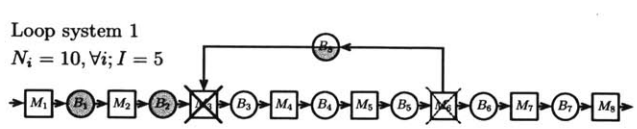
The inventory levels of buffers outside the loop (i.e., Buffers B_1, B_2, B_6, and B_7) are easy to determine. In particular, the buffers upstream of the failed machine will be full, while the buffers between the failure machine and the loop-start machine will be empty, since all remaining parts in these buffers will be processed by the loop through the loop-start machine until it gets starved. On the other hand, because the loop-end machine will look to all buffers downstream of the loop as if it failed, these buffers will be empty. For example, in Scenarios (1.2), (2.2), (3.2), and (4.2) of Figure 5-11 where Machine M_2 fails, B_1 is full while B_2, B_6, and B_7 are empty.

The following conclusions about the limiting propagation state of buffers can be drawn from the discussion above. Given a upstream machine failure:

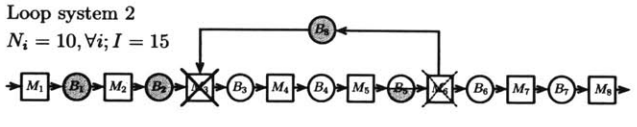
- Buffers upstream of the failed machine will be full,
- Buffers between the failed machine and the loop-start machine will be empty,
- Buffers downstream of the loop will be empty,
- Inventory levels of buffers within the loop are determined by the size of each buffer and the loop invariant as though the loop-start machine failed.

2. Loop-start machine failure

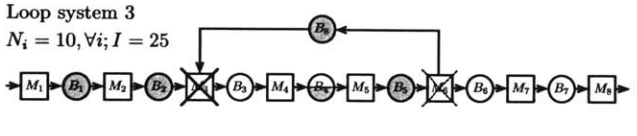
Next, we consider the case where the loop-start machine fails. Note that the failures of Machine M_3 in each of those four systems belong to loop-start machine failures. Therefore, for each system, the third row of the machine failure — buffer level matrix can be filled. They are provided in Figure 5-12.



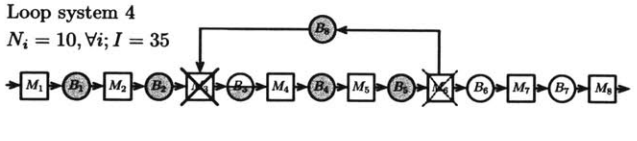
	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	0	0	0	0	5
M_2	10	0	0	0	0	0	0	5
M_3	10	10	0	0	0	0	0	5
M_4								
M_5								
M_6								
M_7								
M_8								



	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	0	5	0	0	10
M_2	10	0	0	0	5	0	0	10
M_3	10	10	0	0	5	0	0	10
M_4								
M_5								
M_6								
M_7								
M_8								



	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	5	10	0	0	10
M_2	10	0	0	5	10	0	0	10
M_3	10	10	0	5	10	0	0	10
M_4								
M_5								
M_6								
M_7								
M_8								



	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	5	10	10	0	0	10
M_2	10	0	5	10	10	0	0	10
M_3	10	10	5	10	10	0	0	10
M_4								
M_5								
M_6								
M_7								
M_8								

Figure 5-12: Loop-start machine failure examples

Again, we first discuss the inventory levels of buffers inside the loop (i.e., Buffers B_3, B_4, B_5, and B_8). In this case, the loop-start machine fails and material flow within the loop moves along the direction of the loop. Therefore, parts start accumulating

at upstream buffer of the loop-start machine (i.e., B_8). The inventory levels of all buffers inside the loop will be determined as the loop-start machine fails. Moreover,

1. For Loop system 1 in Figure 5-12, the loop invariant I is smaller than the size of B_8. Therefore, all parts in the loop will accumulate in B_8 and B_8 will be partially full. As a result, the buffer upstream of the loop-end machine will be empty and the loop-end machine (i.e., M_6) will be starved. Since it is starved, M_6 looks to the buffers downstream of the loop (i.e., B_7 and B_8) as if it failed.

2. For Loop systems 2, 3, and 4 in Figure 5-12, the loop invariant I is larger than the size of B_8, and therefore B_8 will be full. Therefore, the loop-end machine (i.e., M_6) will be blocked. Since it is blocked, M_6 looks to the buffers downstream of the loop (i.e., B_7 and B_8) as if it failed.

The inventory levels of buffers outside the loop (i.e., buffers B_1, B_2, B_6, and B_7) are again easy to determine. In particular, all buffers upstream of the failed machine (the loop-start machine) will be full. On the other hand, because the loop-end machine will look to all buffers downstream of the loop as if it failed, these buffers will be empty. For example, for all four systems in Figure 5-12, B_1 and B_2 are full while B_6 and B_7 are empty.

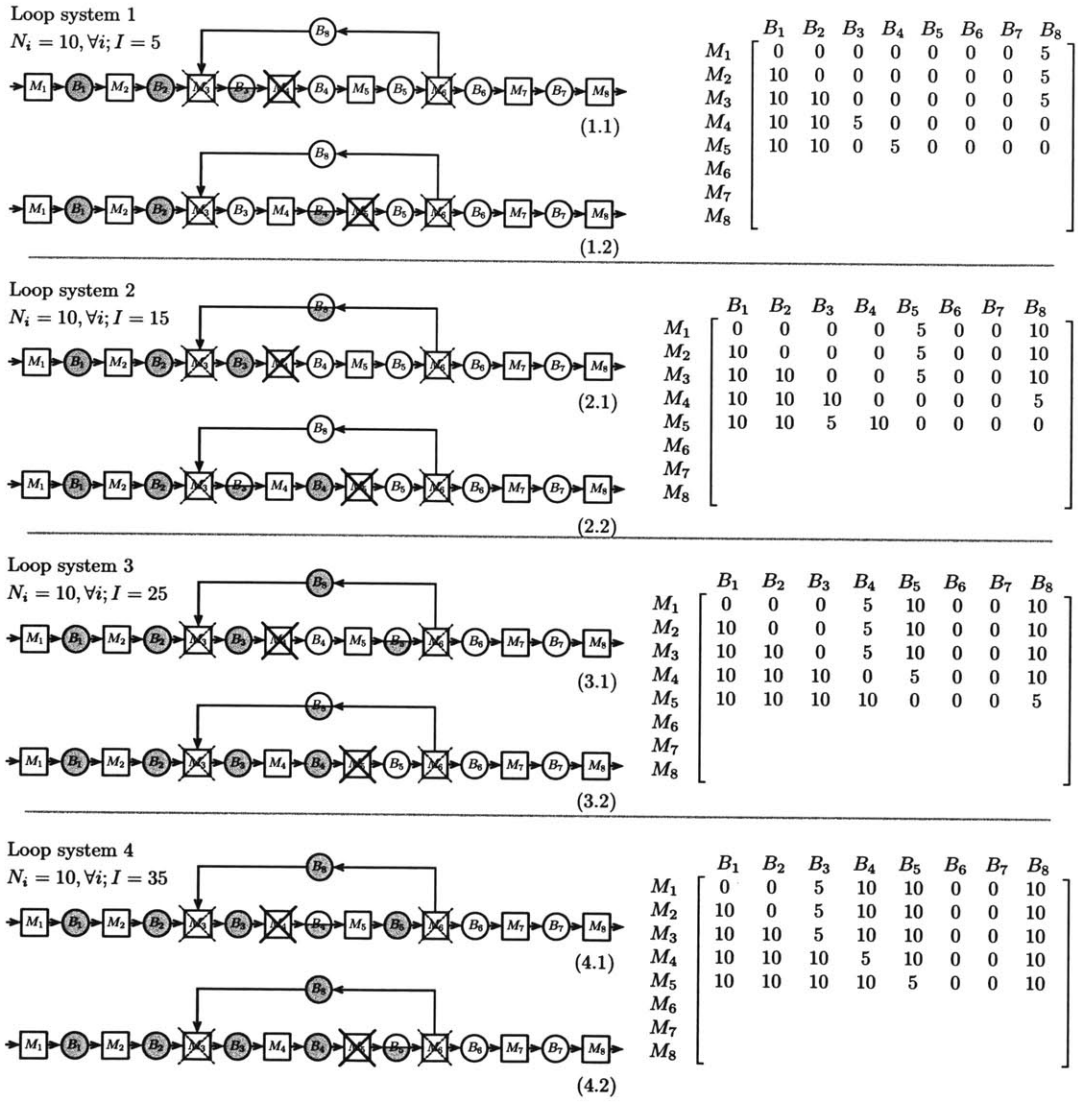
The following conclusions about the limiting propagation state of buffers, given a loop-start machine failure, can be drawn:

- Buffers upstream of the loop will be full,
- Buffers downstream of the loop will be empty,
- Inventory levels of buffers inside the loop are determined by the size of each buffer within the loop and the loop invariant for the case where the loop-start machine fails.

3. Inner loop machine failure

We study the case where one of the inner loop machines fails here. Note that the failures of Machines M_4 and M_5 in each of those four systems belong to inner loop

machine failures. In other words, for each system, we study the limiting propagation states of buffers given that Machines M_4 and M_5 fail, respectively. Therefore, for each system, the fourth row and the fifth row of the machine failure — buffer level matrix can be filled. They are provided in Figure 5-13.



	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	0	0	0	0	5
M_2	10	0	0	0	0	0	0	5
M_3	10	10	0	0	0	0	0	5
M_4	10	10	5	0	0	0	0	0
M_5	10	10	0	5	0	0	0	0
M_6								
M_7								
M_8								

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	0	5	0	0	10
M_2	10	0	0	0	5	0	0	10
M_3	10	10	0	0	5	0	0	10
M_4	10	10	10	0	0	0	0	5
M_5	10	10	5	10	0	0	0	0
M_6								
M_7								
M_8								

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	5	10	0	0	10
M_2	10	0	0	5	10	0	0	10
M_3	10	10	0	5	10	0	0	10
M_4	10	10	10	0	5	0	0	10
M_5	10	10	10	10	0	0	0	5
M_6								
M_7								
M_8								

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	5	10	10	0	0	10
M_2	10	0	5	10	10	0	0	10
M_3	10	10	5	10	10	0	0	10
M_4	10	10	10	5	10	0	0	10
M_5	10	10	10	10	5	0	0	10
M_6								
M_7								
M_8								

Figure 5-13: Inner loop machine failure examples

We first discuss the inventory levels of buffers inside the loop (i.e., Buffers B_3, B_4, B_5, and B_8). Material flow within the loop will move along the direction of the loop. As a result, since a certain inner loop machine fails, parts start accumulating at

buffer upstream of that specific machine. The inventory levels of all buffers inside the loop will be determined by the size of each buffer inside the loop, the loop invariant, and the position of the particular failed inner loop machine.

To determine the inventory levels of buffers outside the loop (i.e., Buffers B_1, B_2, B_6, and B_7), we need to study the status of the loop-start machine and the loop-end machine, given a inner loop machine failure. We consider the loop-start machine first. There are two possibilities:

1. For Scenarios (1.1), (1.2), and (2.2) in Figure 5-11, the buffer upstream of the loop-start machine (i.e., B_8) is empty. The empty B_8 will starve the loop-start machine (i.e., M_3). Since it is starved, M_3 looks to the buffers upstream of the loop (i.e., B_1 and B_2) as if it failed.

2. For all other scenarios in Figure 5-11, the buffer downstream of the loop-start machine is full. Therefore, the loop-start machine (i.e., M_3) will be blocked. Since it is blocked, M_3 looks to the buffers upstream of the loop (i.e., B_1 and B_2) as if it failed.

Similarly, for the loop-end machine, there are three possibilities:

1. For Scenarios (1.1), (1.2), (2.1), (2.2), and (3.2) in Figure 5-11, the buffer upstream of the loop-end machine (i.e., B_5) is empty. The empty B_5 will starve the loop-end machine (i.e., M_6). Since it is starved, M_6 looks to the buffers downstream of the loop (i.e., B_7 and B_8) as if it failed.

2. For all other scenarios in Figure 5-11, the buffer downstream of the loop-end machine (i.e., B_8) is full. Therefore, the loop-end machine (i.e., M_6) will be blocked. Since it is blocked, M_6 looks to the buffers downstream of the loop (i.e., B_7 and B_8) as if it failed.

According to the analysis above, we know that the loop-start and loop-end machines will look to the buffers upstream of the loop and the buffers downstream of the loop as if they failed, respectively. As a result, all buffers upstream of the loop will be full, while all buffers downstream of the loop will be empty. For instance, in

all four systems of Figure 5-11, B_1 and B_2 are full while B_6 and B_7 are empty. The following conclusions about the limiting propagation state of buffers, given a inner loop machine failure, can be drawn:

- Buffers upstream the loop will be full,
- Buffers downstream the loop will be empty,
- Inventory levels of buffers in the loop are determined by the size of each buffer within the loop, the loop invariant, and the position of the particular failed machine.

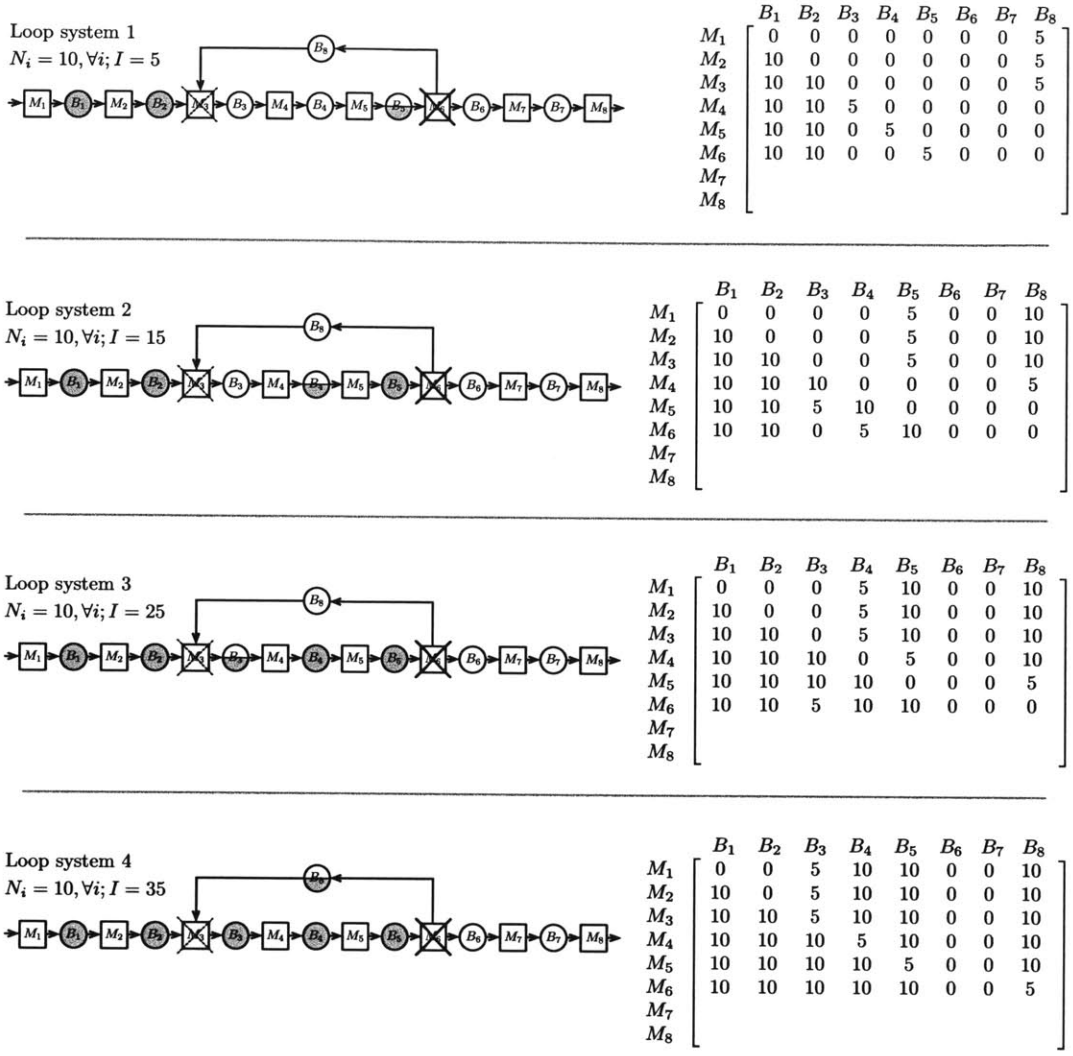
4. Loop-end machine failure

Next, we study the case where the loop-end machine fails. Note that the failures of Machine M_6 in each of those four systems belong to loop-end machine failures. Therefore, for each system, the sixth row of the machine failure — buffer level matrix can be filled. They are provided in Figure 5-14.

Again, we first discuss the inventory levels of buffers inside the loop (i.e., Buffers B_3, B_4, B_5, and B_8). In this case, the loop-end machine fails and material flow within the loop moves along the direction of the loop. Therefore, parts start accumulating at the upstream buffer of the loop-end machine (i.e., B_6). The inventory levels of all buffers inside the loop will be determined as the loop-end machine fails. Moreover,

1. For Loop systems 1, 2, and 3 in Figure 5-14, the buffer upstream of the loop-start machine will be empty because $I \leq N_3 + N_4 + N_5$. As a result, the loop-start machine (i.e., M_3) will be starved. Since it is starved, M_3 looks to the buffers upstream of the loop (i.e., B_1 and B_2) as if it failed.

2. For Loop system 4 in Figure 5-14, the buffer downstream of the loop-start machine will be full because $I > N_3 + N_4 + N_5$. Therefore, the loop-start machine (i.e., M_3) will be blocked. Since it is blocked, M_3 looks to the buffers upstream of the loop (i.e., B_1 and B_2) as if it failed.



	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	0	0	0	0	5
M_2	10	0	0	0	0	0	0	5
M_3	10	10	0	0	0	0	0	5
M_4	10	10	5	0	0	0	0	0
M_5	10	10	0	5	0	0	0	0
M_6	10	10	0	0	5	0	0	0
M_7								
M_8								

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	0	5	0	0	10
M_2	10	0	0	0	5	0	0	10
M_3	10	10	0	0	5	0	0	10
M_4	10	10	10	0	0	0	0	5
M_5	10	10	5	10	0	0	0	0
M_6	10	10	0	5	10	0	0	0
M_7								
M_8								

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	5	10	0	0	10
M_2	10	0	0	5	10	0	0	10
M_3	10	10	0	5	10	0	0	10
M_4	10	10	10	0	5	0	0	10
M_5	10	10	10	10	0	0	0	5
M_6	10	10	5	10	10	0	0	0
M_7								
M_8								

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	5	10	10	0	0	10
M_2	10	0	5	10	10	0	0	10
M_3	10	10	5	10	10	0	0	10
M_4	10	10	10	5	10	0	0	10
M_5	10	10	10	10	5	0	0	10
M_6	10	10	10	10	10	0	0	5
M_7								
M_8								

Figure 5-14: Loop-end machine failure examples

The inventory levels of buffers outside the loop (i.e., Buffers B_1, B_2, B_6, and B_7) are easy to determined. In particular, all buffers downstream of the failure machine (the loop-end machine) will be empty. On the other hand, because the loop-start machine will look to all buffers upstream of the loop as if it failed, these buffers will be full. For example, for all four systems in Figure 5-14, B_1 and B_2 are full while B_6 and B_7 are empty.

The following conclusions about the limiting propagation state of buffers, given a loop-end machine failure, can be drawn:

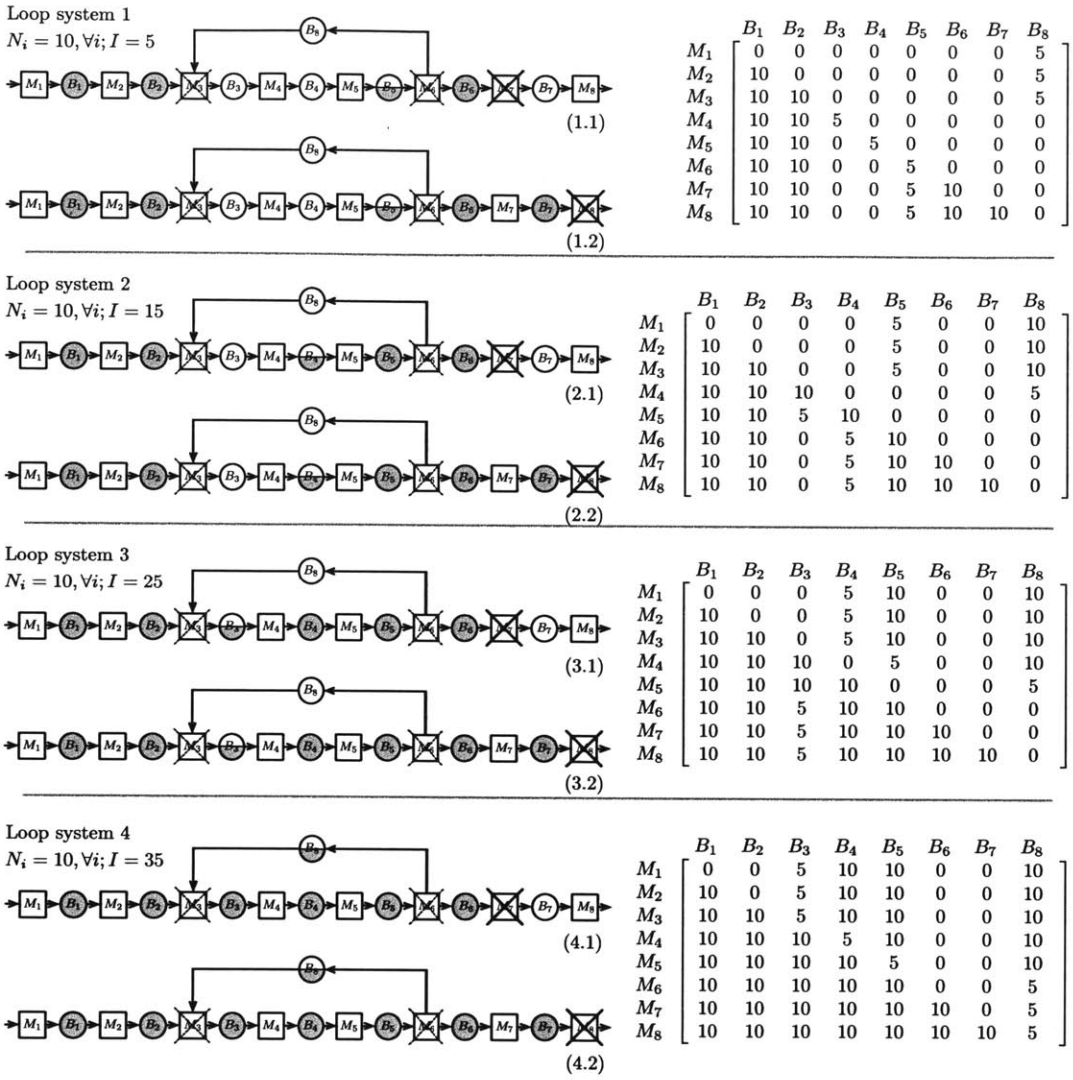
- Buffers upstream the loop will be full,
- Buffers downstream the loop will be empty,
- Inventory levels of buffers inside the loop are determined by the size of each buffer within the loop and the loop invariant for the case where the loop-end machine fails.

5. Downstream machine failure

Finally, we analyze the case where one of the machines downstream the loop fails. Note that the failures of Machines M_7 and M_8 in each of those four systems belong to downstream machine failures. In other words, for each system, we study the limiting propagation states of buffers given that M_7 and M_8 fail, respectively. Therefore, for each system, the last two rows of the machine failure — buffer level matrix can be filled and we can finally finish building up the machine failure — buffer level matrix for each of the four systems. They are provided in Figure 5-15.

We first discuss the inventory levels of buffers inside the loop (i.e., Buffers B_3, B_4, B_5, and B_8). In order to determine them, we study the status of the loop-end machine, because it can be blocked due to downstream machine failures. We realize that the buffer downstream of the loop-end machine (i.e., B_6) is always full given a downstream machine failure, and therefore the loop-end machine will be blocked. Since it is blocked, the loop-end machine (i.e., M_6) looks to the rest of the system as if it failed. Material flow within the loop will move along the direction of the loop. Therefore, parts start accumulating at the buffer upstream of the loop-end machine (i.e., B_5). The inventory levels of all buffers inside the loop will be determined as through the loop-end machine fails. Moreover,

1. For Loop systems 1, 2, and 3 in Figure 5-14, the buffer upstream of the loop-start machine will be empty because $I \leq N_3 + N_4 + N_5$. As a result, the loop-start machine (i.e., M_3) will be starved. Since it is starved, M_3 looks to the buffers upstream of the loop (i.e., B_1 and B_2) as if it failed.



	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	0	0	0	0	5
M_2	10	0	0	0	0	0	0	5
M_3	10	10	0	0	0	0	0	5
M_4	10	10	5	0	0	0	0	0
M_5	10	10	0	5	0	0	0	0
M_6	10	10	0	0	5	0	0	0
M_7	10	10	0	0	5	10	0	0
M_8	10	10	0	0	5	10	10	0

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	0	5	0	0	10
M_2	10	0	0	0	5	0	0	10
M_3	10	10	0	0	5	0	0	10
M_4	10	10	10	0	0	0	0	5
M_5	10	10	5	10	0	0	0	0
M_6	10	10	0	5	10	0	0	0
M_7	10	10	0	5	10	10	0	0
M_8	10	10	0	5	10	10	10	0

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	0	5	10	0	0	10
M_2	10	0	0	5	10	0	0	10
M_3	10	10	0	5	10	0	0	10
M_4	10	10	10	0	5	0	0	10
M_5	10	10	10	10	0	0	0	5
M_6	10	10	5	10	10	0	0	0
M_7	10	10	5	10	10	10	0	0
M_8	10	10	5	10	10	10	10	0

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
M_1	0	0	5	10	10	0	0	10
M_2	10	0	5	10	10	0	0	10
M_3	10	10	5	10	10	0	0	10
M_4	10	10	10	5	10	0	0	10
M_5	10	10	10	10	5	0	0	10
M_6	10	10	10	10	10	0	0	5
M_7	10	10	10	10	10	10	0	5
M_8	10	10	10	10	10	10	10	5

Figure 5-15: Downstream machine failure examples

2. For Loop system 4 in Figure 5-14, the buffer downstream of the loop-start machine will be full because $I > N_3 + N_4 + N_5$. Therefore, the loop-start machine (i.e., M_3) will be blocked. Since it is blocked, M_3 looks to the buffers upstream of the loop (i.e., B_1 and B_2) as if it failed.

The inventory levels of buffers outside the loop (i.e., Buffers B_1, B_2, B_6, and B_7) are easy to determined. In particular, the buffers downstream of the failure machine will be empty, while the buffers between the loop-end machine and the failure machine will be full. On the other hand, because the loop-start machine will look to all buffers

upstream of the loop as if it failed, these buffers will be empty. For example, in Scenarios (1.1), (2.1), (3.1), and (4.1) of Figure 5-15 where M_7 fails, B_1, B_2, and B_6 are full while B_7 is empty.

The following conclusions about the limiting propagation state of buffers can be drawn from the discussion above. Given a downstream machine failure:

- Buffers downstream of the failed machine will be empty,
- Buffers between the loop-end machine and the failed machine will be full,
- Buffers upstream of the loop will be full,
- Inventory levels of buffers within the loop are determined by the size of each buffer and the loop invariant as though the loop-end machine failed.

5.3.3 Thresholds

Based on the analysis in the previous section, we can derive the machine failure — buffer level matrix for any given single open-loop system. However, in some cases, the matrix cannot be used directly to construct two-machine one-buffer building blocks for buffers in the system due to the presence of buffer thresholds, which are first introduced by Maggio (2000) and then studied by Werner (2001), Gershwin and Werner (2007), and Zhang (2006). To explain this, we consider the system shown in Figure 5-8 again. Recall that its machine failure — buffer level matrix is

	B_1	B_2	B_3	B_4	B_5
M_1	0	0	7	0	20
M_2	20	0	7	0	20
M_3	20	20	0	0	7
M_4	20	7	20	0	0
M_5	20	7	20	20	0

Taking B_2 as an example, Machines M_4 and M_5 cause B_2 to be partially full with seven parts, but not totally full. As explained in Section 5.2.1, 7 is the threshold of

Buffer B_2. It is helpful to mention that, for single open-loop systems, the thresholds only appear in buffers inside the loop because thresholds are a result of the loop invariant and the sizes of buffers inside the loop. For buffers outside the loop, there are no thresholds. In other words, they will be either full or empty due to the failure mode of a given machine. Moreover, for a given buffer inside the loop, it is possible for it to have more than one threshold. See examples studied in Zhang (2006).

Since thresholds are undesirable, they need to be eliminated. We have mentioned in Section 5.2.1 that thresholds can be eliminated by inserting perfectly reliable machines (Gershwin and Werner 2007). For the system shown in Figure 5-8, we break up buffers B_2, B_3, and B_5 by inserting a (hypothetically) perfectly reliable machine in each of them. After introducing the perfectly reliable machines and eliminating the thresholds, we derive the modified loop (Figure 5-16). In particular, we replace B_2 by a upstream buffer B_{22}, a perfectly reliable machine, and a downstream buffer B_{21}. The size of B_{22} is $13(= N_2 - I)$ while the size of B_{21} is $7(= I)$. Similarly, B_3 is replaced by B_{32}, a perfectly reliable machine, and B_{31}. B_5 is replaced by B_{52}, a perfectly reliable machine, and B_{51}. Then, for all original buffers that do not have thresholds and those newly derived buffers in the modified loop, we apply the blocking and starvation analysis for those five types of machine failures mentioned in Section 5.3.1 and realize that perfectly reliable machines do not fail. From this, we derived the modified machine failure — buffer level matrix.

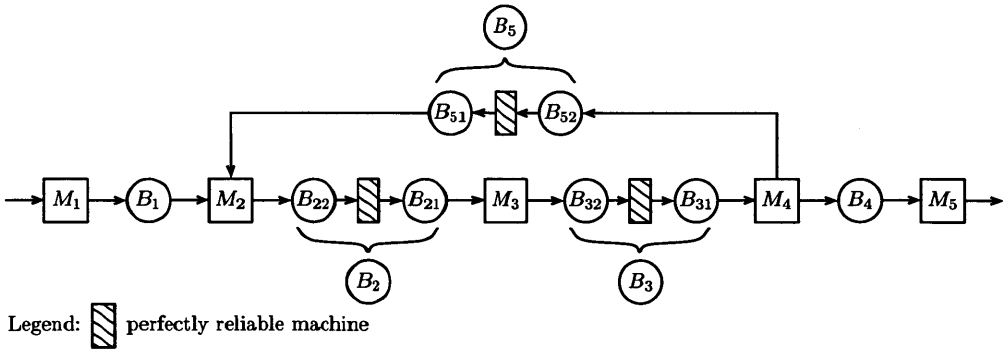


Figure 5-16: The modified loop system of Figure 5-8

$$
\begin{array}{c}
 & \begin{array}{cccccccc} B_1 & B_{22} & B_{21} & B_{32} & B_{31} & B_4 & B_{72} & B_{71} \end{array} \\
\begin{array}{c} M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \end{array} &
\left[\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 7 & 0 & 13 & 7 \\
20 & 0 & 0 & 0 & 7 & 0 & 13 & 7 \\
20 & 13 & 7 & 0 & 0 & 0 & 0 & 7 \\
20 & 0 & 7 & 13 & 7 & 0 & 0 & 0 \\
20 & 0 & 7 & 13 & 7 & 20 & 0 & 0
\end{array}\right]
\end{array}
$$

In the modified matrix, we see that there are eight buffers instead of the original five buffers. However, the most important fact is that for all these eight buffers, they are all either full or empty given the failure of a certain machine. In other words, a very long failure of each machine causes each buffer to be either full or empty. Therefore, we are able to categorize the failure modes of a given machine to the hypothetical two-machine one-buffer building block that contains that buffer. Since, in the modified system, there are eight buffers (rather than five buffers in the original system), there are eight (rather than five) two-machine one-buffer building blocks. This modified machine failure — buffer level matrix enables us to construct two-machine one-buffer building blocks for all buffers in the modified loop system.

5.3.4 Decomposition

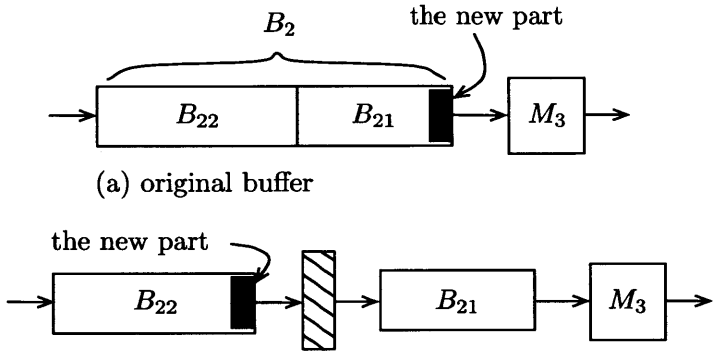
The purpose of all the analysis in Sections 5.3.1, 5.3.2, and 5.3.3 is to address how to construct the upstream pseudo-machine $M^u(B_j)$ and the downstream pseudo-machine $M^d(B_j)$ of the building block that contains Buffer B_j. After we construct the two-machine one-buffer building blocks for all buffers in the modified system, we apply the decomposition method developed by Tolio and Matta (1998) to analyze the single open-loop system. For more details about the decomposition algorithm, refer to Zhang (2006). In the decomposition algorithm, the analytical solutions of Tolio et al. (2002) are used to evaluate each of those two-machine one-buffer building blocks. With those two-machine one-buffer building blocks, the decomposition algorithm determines the parameters for all building blocks and then finds the production rate as well as the average inventory of each buffer of the system.

5.4 Modifications of Loop Evaluation

As mentioned in Section 5.2.3, there are two issues with Werner's algorithm that lead to the Batman effect. We explain and resolve them in this section.

5.4.1 New Model of the Perfectly Reliable Machine without Delay

The first issue occurs when we eliminate buffer thresholds by inserting perfectly reliable machines. Let us study Figure 5-16 again.



(a) original buffer

(b) modified buffer after inserting a perfectly reliable machine

Figure 5-17: Different behavior of Buffer B_2 with and without a perfectly reliable machine

Note that perfectly reliable machines are inserted to eliminate thresholds in buffers B_2, B_3, and B_5. We have to keep in mind that there are no such machines in reality. We explain the problem resulting from the insertion of the perfectly reliable machine by studying B_2 as an example (Figure 5-17). Consider the original buffer without a perfectly reliable machine first (Figure 5-17(a)). Suppose that at some time Buffer B_2 is empty and therefore both hypothetical buffers B_{21} and B_{22} are empty. If a part arrives, the part goes to the hypothetical Buffer B_{21} directly and therefore the downstream machine M_3 will not be starved. However, in the modified buffer case (Figure 5-17(b)), if we model the newly inserted machines conventionally, the part goes to B_{22} first and stays there for one time step, after which it enters B_{21}. Consequently,

M_3 will be starved during that time step and this reduces the production rate of the system. To summarize, in reality (i.e., the original buffer without a perfectly reliable machine), if a part arrives at Buffer B_2, it goes directly to the hypothetical Buffer B_{21} if it is not full. It will stay at B_{22} if and only if B_{21} is full. Thus, if B_{21} is empty, B_{22} must be empty and therefore B_2 is empty. However, if we modify the buffer by inserting a perfectly reliable machine and we model the newly inserted machines conventionally, that part will arrive at B_{22} first and not go to B_{21} until the next time unit. This is because, in the two-machine one-buffer building block of B_{22} in the decomposition, it takes its downstream machine one time unit to process a part to its downstream buffer (B_{21}). Therefore, it is possible that B_{21} is empty while B_{22} is not, which should never occur. The argument above reveals that each perfectly reliable machine (if modeled conventionally) could add a small amount of time delay because material needs to transverse that machine. Zhang (2006) comments that the delay is nearly negligible. From a standpoint of evaluation accuracy, the time delay is indeed negligible. However, since our ultimate goal is the optimization of single open-loop systems, we observe that the time delay is one of the key reasons and difficulties in optimization that lead to the discontinuity in loop evaluation and the Batman effect.

Recall that, in the Batman effect (Figure 5-7), the production rate of the system, derived by Werner's algorithm, when $I = 11$ is smaller than that when $I = 10$, which differs from the simulation result. This is due to the time delay mentioned above. When $I \leq 10$, there is no need to insert perfectly reliable machines. However, when $I > 10$, perfectly reliable machines are inserted to eliminate buffer thresholds. The time delay brought by this effect reduces the production rate of the system, and leads to the discontinuity[5]. Therefore, it is necessary to modify the evaluation algorithm to resolve this issue.

It is important to point out that it is the the evaluation of two-machine one-buffer building blocks in the decomposition approach that requires modifications because of the time delay. In other words, we have to consider how to enable the upstream

[5]The fact that the production rate when $I = 12$ is less than the production rate when $I = 11$ is due to another issue that we address in Section 5.4.2.

and/or the downstream pseudo-machines to have no delay when required. Consider Figure 5-18 that shows the the building block that contains B_{21} of the system shown in Figure 5-16.

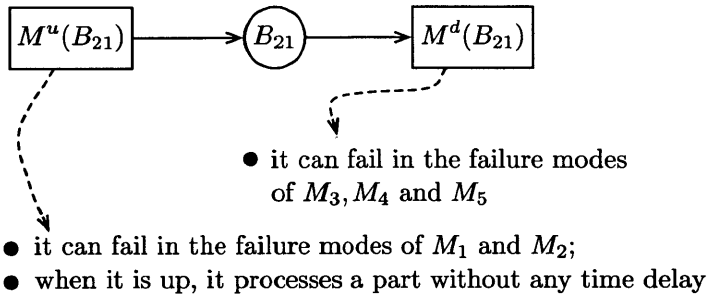


Figure 5-18: A two-machine one-buffer building block whose upstream machine has the no-delay property when it is up

First, we would like to indicate clearly that although the upstream machine of B_{21} in Figure 5-16 is a perfectly reliable machine, the upstream pseudo-machine of B_{21} in Figure 5-18 is **NOT** perfect and has a set of failure modes. According to the modified machine failure — buffer level matrix, machines M_1 and M_2 in the original system can cause B_{21} to be empty. Therefore, the upstream pseudo-machine $M^u(B_{21})$ in Figure 5-18 can fail in the failure modes of M_1 and M_2. Similarly, the downstream pseudo-machine $M^d(B_{21})$ in Figure 5-18 can fail in the failure modes of M_3, M_4 and M_5. However, the key feature for $M^u(B_{21})$ in Figure 5-18 is that whenever it is up, it produces a part without any time delay, because of the perfectly reliable machine upstream of B_{21} in Figure 5-16. In the following, we refer to such a upstream machine as a *no-delay machine*. As a comparison, we refer to a machine that does not exhibit the no-delay property (when it is up) as an *ordinary machine*. We have to modify the existing analytical solutions of Tolio et al. (2002) to cope with the no-delay property of the upstream machine, the downstream machine, or both machines in a two-machine one-buffer building block when necessary, because it takes both machines one time unit to produce a part given no failure occurs in the current model.

Model assumptions

In order to modify the analytical solutions of Tolio et al. (2002), we make the following assumptions about a no-delay machine:

1. If a no-delay machine is not in a failed state, blocked, or starved, it can produce a part **at any instant** and **without any delay** during a time unit.

2. A no-delay machine can **NOT** produce more than one part in any time unit.

Because of these two assumptions, we only need to modify the boundary conditions of the existing Markov chain model for two-machine one-buffer building blocks studied in Tolio et al. (2002). The boundary states refer to the states where the inventory level is 0, 1, $N-1$, or N. This is because if the inventory level n satisfies $2 \leq n \leq N-2$, then it makes no changes to the Markov chain model because of the second assumption. The upstream machine can add at most one part to the buffer in each time unit, and the downstream machine can remove at most one part from the buffer in each time unit. Because $2 \leq n \leq N-2$, the buffer will not be empty or full after that time unit, and therefore the upstream machine will not be blocked and the downstream machine will not be starved. In other words, the no-delay property brings no impact to the buffer level, the upstream machine, or the downstream machine. However, as we will show shortly, if the system is in a boundary state, the no-delay property of a machine can have impact to both the buffer level and the machine states. Consequently, we only need to modify boundary conditions.

In addition, because of the two assumptions, it makes no difference if the upstream machine is the only no-delay machine, the downstream machine is the only no-delay machine, or both upstream and downstream machines are no-delay machines. The three cases above will have exactly the same boundary conditions after modification. In what follows, we first explain intuitively how the boundary conditions change because of no-delay machines. After that, mathematical models and analytical solutions are provided.

Ordinary machines, full buffer case

We first study the full buffer case (i.e., the upper boundary condition) with two ordinary machines. Assume that the buffer is full and the downstream machine is down at the end of time unit t.

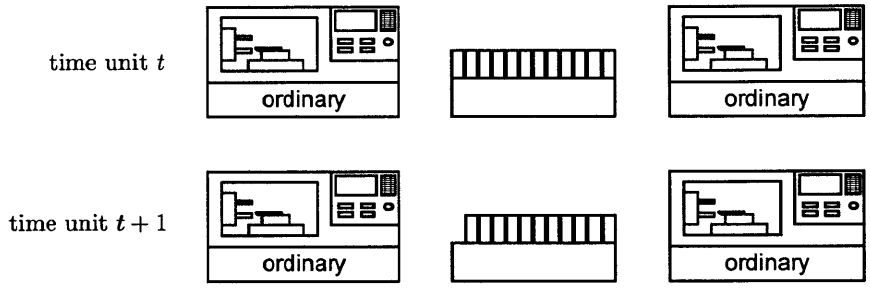


Figure 5-19: Ordinary machine, full buffer case

According to the model convention of Tolio et al. (2002), machines change states at the beginning of a time unit, while the buffer level changes at the end of a time unit. Therefore, if the downstream machine gets repaired, it will become up at the beginning of time unit $t+1$. Then it will work on the first part in the buffer during time unit $t+1$. The buffer level becomes $N-1$ at the end of time unit $t+1$, and the upstream machine is blocked during the entire time unit $t+1$.

No-delay upstream machine – ordinary downstream machine, full buffer case

Now, assume that the upstream machine is a no-delay machine and the downstream machine is an ordinary machine. The buffer is full and the downstream machine is down at the end of time unit t.

If the downstream machine gets repaired, it will be up at the beginning of time unit $t+1$. Then it will work on the first part in the buffer during time unit $t+1$. At the end of time unit $t+1$, the buffer level first becomes $N-1$ since the downstream machine processes a part. However, due to its no-delay property, the upstream machine will add a part (the solid one) to the buffer simultaneously at the end of time unit $t+1$, if it does not fail. The two effects cancel out and the buffer level remains N.

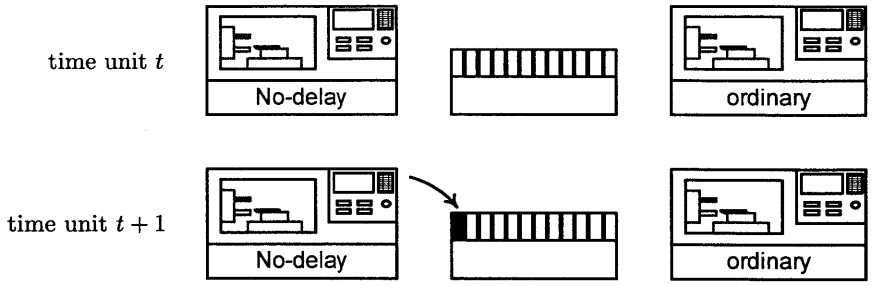


Figure 5-20: No-delay upstream machine – ordinary downstream machine, full buffer case

Ordinary upstream machine – no-delay downstream machine, full buffer case

Next, assume that the downstream machine is a no-delay machine and the upstream machine is an ordinary machine. The buffer is full and the downstream machine is down at the end of time unit t.

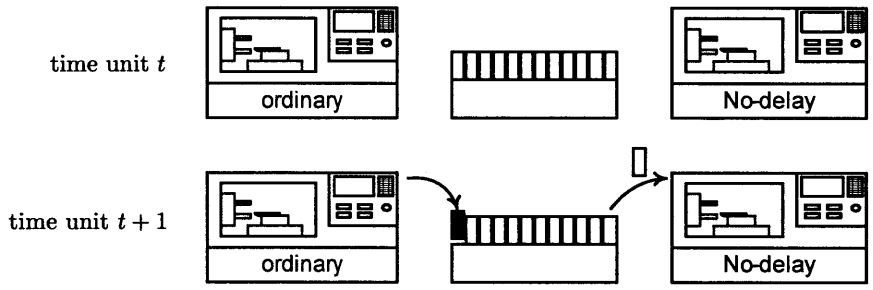


Figure 5-21: Ordinary upstream machine – no-delay downstream machine, full buffer case

If the downstream machine gets repaired, it will be up at the beginning of time unit $t+1$. Then it will work on the first part in the buffer and remove it from the buffer without any delay. As a result, the buffer level goes to $N-1$ at the beginning of time unit $t+1$. Therefore, the upstream machine is no longer blocked. If it does not fail, it adds a part into the buffer at the end of time unit $t+1$. So, the two effects cancel out and the buffer level remains N.

Ordinary machines, empty buffer case

Now, let us study the empty buffer case (i.e., the lower boundary condition) with two ordinary machines. Assume that the buffer is empty and the upstream machine is down at the end of time unit t.

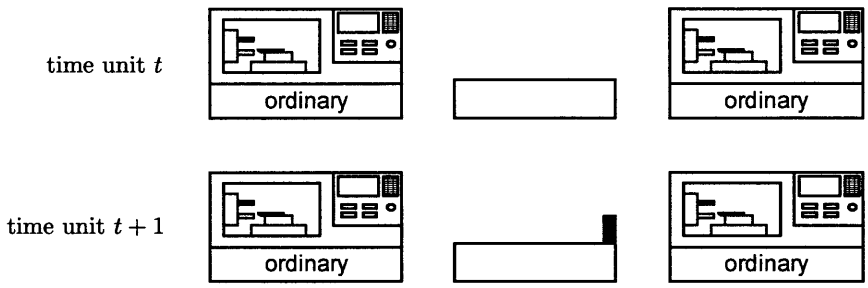


Figure 5-22: Ordinary machine, empty buffer case

If the upstream machine gets repaired, it will become up at the beginning of time unit $t+1$. Then it will add a part to the buffer, and the buffer level becomes 1 at the end of time unit $t+1$, while the downstream machine is starved during the entire time unit $t+1$.

No-delay upstream machine – ordinary downstream machine, empty buffer case

Assume that the upstream machine is a no-delay machine and the downstream machine is an ordinary machine. The buffer is empty and the upstream machine is down at the end of time unit t.

If the upstream machine gets repaired, it will be up at the beginning of time unit $t+1$. Then it will add a part to the buffer and the inventory level is 1 at the beginning of time unit $t+1$ due to the no-delay property of the upstream machine. As a result, the downstream machine will not be starved. If it does not fail, it will work on and remove that part from the buffer at the end of time unit $t+1$. So, the two effects cancel out and the buffer remains empty.

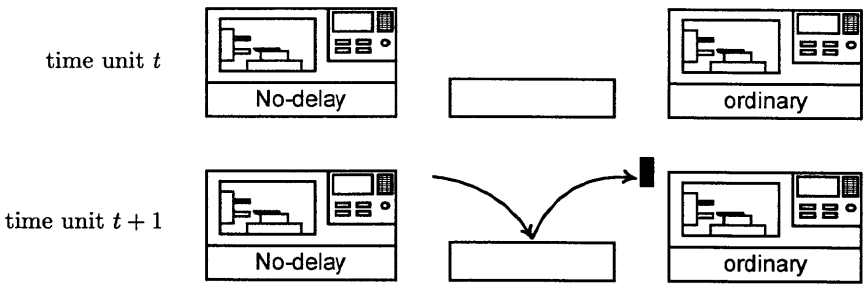


Figure 5-23: No-delay upstream machine – ordinary downstream machine, empty buffer case

Ordinary upstream machine – no-delay downstream machine, empty buffer case

Next, assume that the downstream machine is a no-delay machine but the upstream machine is an ordinary machine. The buffer is empty and the upstream machine is down at the end of time unit t.

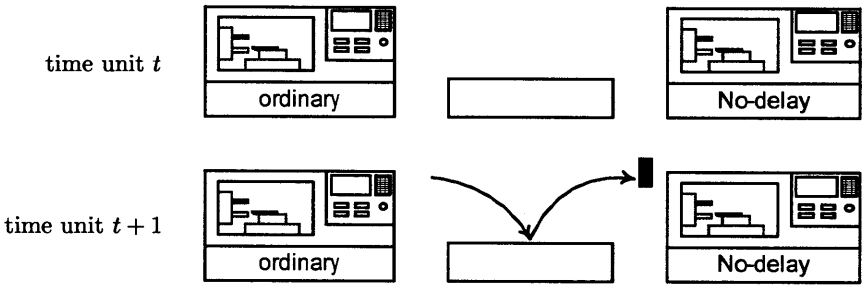


Figure 5-24: Ordinary upstream machine – no-delay downstream machine, empty buffer case

If the upstream machine gets repaired, it will be up at the beginning of time unit $t+1$. Then it will add a part to the buffer and the inventory level first becomes 1 at the end of time unit $t+1$. However, due to its no-delay property, the downstream machine processes that part immediately. So, the two effects cancel out and the buffer remains empty.

As implied by the six examples above, we see that by introducing no-delay machines, both upper and lower boundary conditions in the two-machine one-buffer building blocks that have such machines should be modified. In addition, the two

no-delay full buffer cases indicate that no matter whether the upstream or the downstream machine is the no-delay machine, the upper boundary conditions have the same changes. Similarly, the two no-delay empty buffer cases indicate that the lower boundary conditions have the same changes as well regardless of which machine is a no-delay machine. As a final reminder, we have indicated that, because of the two model assumptions about no-delay machines, it makes no difference if the upstream machine is the only no-delay machine, the downstream machine is the only no-delay machine, or both upstream and downstream machines are no-delay machines. These three cases have exactly the same boundary conditions.

Modifications to mathematical models

Here we explain how we modify the boundary conditions in the Markovian two-machine one-buffer building block model of Tolio et al. (2002). In essence, we will modify the transition equations that contain upper and lower boundary states. In Tolio et al. (2002), the state of the system is defined as (n, α_1, α_2), where n is the buffer level ($0 \leq n \leq N$), α_1 is the state of the upstream machine, and α_2 is the state of the downstream machine. If the upstream machine is operational, $\alpha_1 = 1$. Otherwise $\alpha_1 = u_i$ for some $i = 1, \cdots, s$ where u_i represents the failure mode of the machine. Similarly, α_2 can assume the values 1, $d_1, \cdots, d_t$. The steady state probability of the system being in state (n, α_1, α_2) is indicated by $\mathbf{p}(n, \alpha_1, \alpha_2)$. In particular, if the upstream machine is operational, it can fail in mode u_i with probability p^{u_i} while attempting to perform an operation. When the upstream machine is failed in mode u_i, it can get repaired during a time unit with probability r^{u_i}. Similarly p^{d_j} and r^{d_j} represent respectively failure and repair probabilities for the failure modes of the downstream machine. The total failure probability P^U of the upstream machine, i.e., the probability of failure regardless of the mode in which the machine fails, is given by $P^U = \sum_{i=1}^{s} p^{u_i}$. Similarly, the total failure probability P^D of the downstream machine is $P^D = \sum_{i=1}^{t} p^{d_j}$. P^U and P^D must satisfy $P^U \leq 1$ and $P^D \leq 1$.

The boundary states include $(0, \alpha_1, \alpha_2)$, $(1, \alpha_1, \alpha_2)$, $(N-1, \alpha_1, \alpha_2)$, and (N, α_1, α_2). We modify the Markov chain model of the two-machine one-buffer building block

with the additional transitions among boundary states because of the new no-delay features. Then it can be seen that states $(0,1,d_j),(0,u_i,d_j),(N,u_i,1),(N,u_i,d_j), i = 1,\cdots,s, j = 1,\cdots,j$ are transient because they cannot be visited from recurrent states. Therefore, the long term steady state probabilities of these states are 0. Next, we derive the new steady state probabilities of other recurrent boundary states.

New steady state transition equations of lower boundary states

New transition equations for those recurrent lower boundary states are provided as follows.

$$\mathbf{p}(0,1,1) = \mathbf{p}(0,1,1)(1-P^U)(1-P^D) + \sum_{i=1}^{s} \mathbf{p}(0,u_i,1) r^{u_i}(1-P^D), \tag{5.1}$$

$$\begin{aligned} \mathbf{p}(0,u_i,1) = {} & \mathbf{p}(0,1,1)p^{u_i} + \mathbf{p}(0,u_i,1)(1-r^{u_i}) + \mathbf{p}(1,1,1)p^{u_i}(1-P^D) \\ & + \sum_{j=1}^{t} \mathbf{p}(1,1,d_j)p^{u_i} r^{d_j} + \sum_{j=1}^{t} \mathbf{p}(1,u_i,d_j)(1-r^{u_i}) r^{d_j} \\ & + \mathbf{p}(1,u_i,1)(1-r^{u_i})(1-P^D), \end{aligned} \tag{5.2}$$

$$\begin{aligned} \mathbf{p}(1,1,1) = {} & \mathbf{p}(1,1,1)(1-P^U)(1-P^D) + \sum_{j=1}^{t} \mathbf{p}(1,1,d_j)(1-P^U) r^{d_j} \\ & + \sum_{i=1}^{s}\sum_{j=1}^{t} \mathbf{p}(1,u_i,d_j) r^{u_i} r^{d_j} + \sum_{i=1}^{s} \mathbf{p}(1,u_i,1) r^{u_i}(1-P^D), \end{aligned} \tag{5.3}$$

$$\mathbf{p}(1,1,d_j) = \mathbf{p}(0,1,1)(1-P^U)p^{d_j} + \sum_{i=1}^{s} \mathbf{p}(0,u_i,1) r^{u_i} p^{d_j}, \tag{5.4}$$

$$\begin{aligned} \mathbf{p}(1,u_i,d_j) = {} & \mathbf{p}(1,1,1)p^{u_i}p^{d_j} + \mathbf{p}(1,1,d_j)p^{u_i}(1-r^{d_j}) \\ & + \mathbf{p}(1,u_i,d_j)(1-r^{u_i})(1-r^{d_j}) + \mathbf{p}(1,u_i,1)(1-r^{u_i})p^{d_j}, \end{aligned} \tag{5.5}$$

$$\begin{aligned}\mathbf{p}(1,u_i,1) &= \mathbf{p}(2,1,1)p^{u_i}(1-P^D)+\sum_{j=1}^{t}\mathbf{p}(2,1,d_j)p^{u_i}r^{d_j} \\ &\quad +\sum_{j=1}^{t}\mathbf{p}(2,u_i,d_j)(1-r^{u_i})r^{d_j}+\mathbf{p}(2,u_i,1)(1-r^{u_i})(1-P^D).\end{aligned} \tag{5.6}$$

We use State (1,1,1) as an example to explain the difference between the transition equations of the modified model and those of the original model. Equations (5.7) and (5.3) are the transition equations for (1,1,1) in the original model and the modified model, respectively.

$$\begin{aligned}\mathbf{p}(1,1,1) &= \mathbf{p}(1,1,1)(1-P^U)(1-P^D)+\sum_{i=1}^{s}\mathbf{p}(1,u_i,1)r^{u_i}(1-P^D) \\ &\quad +\sum_{i=1}^{s}\sum_{j=1}^{t}\mathbf{p}(1,u_i,d_j)r^{u_i}r^{d_j}+\sum_{i=1}^{s}\mathbf{p}(0,u_i,1)r^{u_i}.\end{aligned} \tag{5.7}$$

The first three terms on the right hand sides of both equations are the same. However, the fourth terms are different. In the model with ordinary machines, (1,1,1) can be reached from (0,u_i,1) with probability $r^{u_i}, i = 1,\cdots,s$ if M_u is repaired and adds a part into the buffer, while M_d is starved during that time unit. However, in the modified model with no-delay machine(s), (1,1,1) cannot be reached from (0,u_i,1). But it can be reached from state (1,1,d_j) with probability $(1-P^U)r^{d_j}, j = 1\cdots,t$. Note that (1,1,$d_j$) is a transient state in the original model. However, it becomes recurrent in the modified model. Similar analysis applies to other boundary states.

We realize that Equation (5.4) is the simplified form of

$$\begin{aligned}\mathbf{p}(1,1,d_j) &= \mathbf{p}(0,1,1)(1-P^U)p^{d_j}+\mathbf{p}(0,1,d_j)(1-P^U)(1-r^{d_j}) \\ &\quad +\sum_{i=1}^{s}\mathbf{p}(0,u_i,d_j)r^{u_i}(1-r^{d_j})+\sum_{i=1}^{s}\mathbf{p}(0,u_i,1)r^{u_i}p^{d_j}\end{aligned} \tag{5.8}$$

because $\mathbf{p}(0,1,d_j)=\mathbf{p}(0,u_i,d_j)=0, i=1,\cdots,s, j=1,\cdots,t$. Therefore, the forms of Equations (5.3), (5.5), (5.6), and (5.8) are the same as the forms of transition equations for the internal states in Tolio et al. (2002). As a result, we conclude that

under new lower boundary transition equations, states $(1,1,1)$, $(1,u_i,1)$, $(1,1,d_j)$, and $(1,u_i,d_j)$ can be treated as internal states. Consequently, their probabilities can be expressed as the internal form. Thus, according to Tolio et al. (2002), we have

$$\begin{aligned}
\mathbf{p}(1,1,1) &= \sum_{m=1}^{R} C_m X_m, \\
\mathbf{p}(1,1,d_j) &= \sum_{m=1}^{R} C_m X_m D_{j,m}, \\
\mathbf{p}(1,u_i,1) &= \sum_{m=1}^{R} C_m X_m U_{i,m}, \\
\mathbf{p}(1,u_i,d_j) &= \sum_{m=1}^{R} C_m X_m U_{i,m} D_{j,m},
\end{aligned} \tag{5.9}$$

where $R = s + t$, $C_m, m = 1, \cdots, R$ are normalization constants, and $X_m, U_{i,m}$ and $D_{j,m}$ $i = 1, \cdots, s, j =, 1 \cdots, t, m = 1, \cdots, R$ are defined in Tolio et al. (2002). Hence, the remaining unknown probabilities are $\mathbf{p}(0,1,1)$ and $\mathbf{p}(0,u_i,1), i = 1, \cdots, s$. We derive them now. Comparing (5.1) and (5.4) we have

$$\frac{1}{p^{d_j}}\mathbf{p}(1,1,d_j) = \frac{1}{1-P^D}\mathbf{p}(0,1,1). \tag{5.10}$$

Thus,

$$\mathbf{p}(0,1,1) = \frac{1-P^D}{p^{d_j}}\mathbf{p}(1,1,d_j) = \frac{1-P^D}{p^{d_j}}\sum_{m=1}^{R} C_m X_m D_{j,m}. \tag{5.11}$$

Equation (5.2) can be written as

$$\begin{aligned}
r^{u_i}\mathbf{p}(0,u_i,1) &= \mathbf{p}(0,1,1)p^{u_i} + \mathbf{p}(1,1,1)p^{u_i}(1-P^D) + \sum_{j=1}^{t}\mathbf{p}(1,1,d_j)p^{u_i}r^{d_j} \\
&\quad + \sum_{j=1}^{t}\mathbf{p}(1,u_i,d_j)(1-r^{u_i})r^{d_j} + \mathbf{p}(1,u_i,1)(1-r^{u_i})(1-P^D).
\end{aligned} \tag{5.12}$$

Plugging in the expressions for the terms on the right hand side, we find

$$
\begin{aligned}
r^{u_i}\mathbf{p}(0,u_i,1) &= \frac{p^{u_i}}{p^{d_j}}(1-P^D)\sum_{m=1}^{R} C_m X_m D_{j,m} + \sum_{m=1}^{R} C_m X_m p^{u_i}(1-P^D) \\
&\quad + p^{u_i}\sum_{m=1}^{R} C_m X_m \sum_{j=1}^{t} D_{j,m} r^{d_j} + \sum_{j=1}^{t}\sum_{m=1}^{R} C_m X_m U_{i,m} D_{j,m} r^{d_j}(1-r^{u_i}) \\
&\quad + \sum_{m=1}^{R} C_m X_m U_{i,m}(1-r^{u_i})(1-P^D) \\
&= \frac{p^{u_i}}{p^{d_j}}(1-P^D)\sum_{m=1}^{R} C_m X_m D_{j,m} + p^{u_i}\sum_{m=1}^{R} C_m X_m \left[\sum_{j=1}^{t} D_{j,m} r^{d_j} + (1-P^D)\right] \\
&\quad + (1-r^{u_i})\sum_{m=1}^{R} C_m X_m U_{i,m}\left[\sum_{j=1}^{t} D_{j,m} r^{d_j} + (1-P^D)\right].
\end{aligned} \tag{5.13}
$$

Note that, according to Tolio et al. (2002),

$$
\sum_{j=1}^{t} D_{j,m} r^{d_j} + (1-P^D) = \frac{1}{X_m K_m}. \tag{5.14}
$$

Therefore, (5.13) becomes

$$
r^{u_i}\mathbf{p}(0,u_i,1) = \frac{p^{u_i}}{p^{d_j}}(1-P^D)\sum_{m=1}^{R} C_m X_m D_{j,m} + p^{u_i}\sum_{m=1}^{R}\frac{C_m}{K_m} + (1-r^{u_i})\sum_{m=1}^{R} C_m \frac{U_{i,m}}{K_m}. \tag{5.15}
$$

Hence,

$$
\mathbf{p}(0,u_i,1) = \frac{p^{u_i}(1-P^D)}{r^{u_i}p^{d_j}}\sum_{m=1}^{R} C_m X_m D_{j,m} + \frac{p^{u_i}}{r^{u_i}}\sum_{m=1}^{R}\frac{C_m}{K_m} + \frac{1-r^{u_i}}{r^{u_i}}\sum_{m=1}^{R} C_m \frac{U_{i,m}}{K_m}. \tag{5.16}
$$

Therefore (5.11) and (5.16) are the steady state probabilities of $(0,1,1)$ and $(0,u_i,1), i = 1,\cdots,s$. The next step in Tolio et al. (2002) is to construct a set of

equations that can be used to find the normalization constants $C_1, C_2, \cdots, C_R$. Since we have modified transition equations of those lower boundary states, we implicitly change the set of equations for $C_1, C_2, \cdots, C_R$. Either Equation (5.1) or (5.3) can be used to set up part of those set of equations, and they provide exactly the same results[6]. Assume that we use equation (5.1) and plug in the expressions of $\mathbf{p}(0,1,1)$ and $\mathbf{p}(0,u_i,1), i = 1, \cdots, s$, then

$$\begin{aligned}\frac{P^D + P^U - P^D P^U}{p^{d_j}} \sum_{m=1}^{R} C_m X_m D_{j,m} &= \frac{P^U(1-P^D)}{p^{d_j}} \sum_{m=1}^{R} C_m X_m D_{j,m} \\ &\quad + \sum_{i=1}^{s} p^{u_i} \sum_{m=1}^{R} \frac{C_m}{K_m} + \sum_{i=1}^{s} (1 - r^{u_i}) \sum_{m=1}^{R} C_m \frac{U_{i,m}}{K_m},\end{aligned} \tag{5.17}$$

or

$$\frac{P^D}{p^{d_j}} \sum_{m=1}^{R} C_m X_m D_{j,m} = P^U \sum_{m=1}^{R} \frac{C_m}{K_m} + \sum_{m=1}^{R} \frac{C_m}{K_m} \sum_{i=1}^{s} (1 - r^{u_i}) U_{i,m}, \quad j = 1, \cdots, t. \tag{5.18}$$

Therefore, (5.18) consists of t equations for solving $C_1, C_2, \cdots, C_R$. The s other equations for solving them can be found from the modified transition equations of the upper boundary states and we will address them shortly. On the other hand, if we instead use (5.3) to find $C_1, C_2, \cdots, C_R$, then

$$\begin{aligned}\left[P^U + P^D - P^U P^D\right] \sum_{m=1}^{R} C_m X_m &= \frac{1 - P^U}{p^{d_j}} \sum_{m=1}^{R} C_m X_m D_{j,m} \sum_{k=1}^{t} p^{d_k} r^{d_k} \\ &\quad + \sum_{m=1}^{R} C_m X_m \left[\sum_{i=1}^{s} \sum_{k=1}^{t} U_{i,m} D_{k,m} r^{u_i} r^{d_k} + \sum_{i=1}^{s} U_{i,m} r^{u_i} (1 - P^D) \right]\end{aligned} \tag{5.19}$$

for $j = 1, 2, \cdots, t$. Thus, (5.19) contains t equations for solving $C_1, C_2, \cdots, C_R$. Solving $C_1, C_2, \cdots, C_R$ with either (5.18) or (5.19) giving exactly the same results.

[6] We have verified this with numerical experiments.

New steady state transaction equations of upper boundary states

New upper boundary transition equations are

$$\mathbf{p}(N,1,1) = \mathbf{p}(N,1,1)(1-P^U)(1-P^D) + \sum_{j=1}^{t} \mathbf{p}(N,1,d_j)(1-P^U)r^{d_j}, \quad (5.20)$$

$$\begin{aligned}\mathbf{p}(N,1,d_j) = {} & \mathbf{p}(N,1,1)p^{d_j} + \mathbf{p}(N,1,d_j)(1-r^{d_j}) \\ & + \mathbf{p}(N-1,1,1)(1-P^U)p^{d_j} + \mathbf{p}(N-1,1,d_j)(1-P^U)(1-r^{d_j}) \\ & + \sum_{i=1}^{s} \mathbf{p}(N-1,u_i,d_j)r^{u_i}(1-r^{d_j}) + \sum_{i=1}^{s} \mathbf{p}(N-1,u_i,1)r^{u_i}p^{d_j},\end{aligned} \quad (5.21)$$

$$\begin{aligned}\mathbf{p}(N-1,1,1) = {} & \mathbf{p}(N-1,1,1)(1-P^U)(1-P^D) + \sum_{j=1}^{t} \mathbf{p}(N-1,1,d_j)(1-P^U)r^{d_j} \\ & + \sum_{i=1}^{s}\sum_{j=1}^{t} \mathbf{p}(N-1,u_i,d_j)r^{u_i}r^{d_j} + \sum_{i=1}^{s} \mathbf{p}(N-1,u_i,1)r^{u_i}(1-P^D),\end{aligned} \quad (5.22)$$

$$\begin{aligned}\mathbf{p}(N-1,1,d_j) = {} & \mathbf{p}(N-2,1,1)(1-P^U)p^{d_j} + \mathbf{p}(N-2,1,d_j)(1-P^U)(1-r^{d_j}) \\ & + \sum_{i=1}^{s} \mathbf{p}(N-2,u_i,d_j)r^{u_i}(1-r^{d_j}) + \sum_{i=1}^{s} \mathbf{p}(N-2,u_i,1)r^{u_i}p^{d_j},\end{aligned} \quad (5.23)$$

$$\begin{aligned}\mathbf{p}(N-1,u_i,d_j) = {} & \mathbf{p}(N-1,1,1)p^{u_i}p^{d_j} + \mathbf{p}(N-1,1,d_j)p^{u_i}(1-r^{d_j}) \\ & + \mathbf{p}(N-1,u_i,d_j)(1-r^{u_i})(1-r^{d_j}) + \mathbf{p}(N-1,u_i,1)(1-r^{u_i})p^{d_j},\end{aligned} \quad (5.24)$$

$$\mathbf{p}(N-1,u_i,1) = \mathbf{p}(N,1,1)p^{u_i}(1-P^D) + \sum_{j=1}^{t} \mathbf{p}(N,1,d_j)p^{u_i}r^{d_j}. \quad (5.25)$$

We realize that Equation (5.25) is the simplified form of

$$\begin{aligned}\mathbf{p}(N-1,u_i,1) &= \mathbf{p}(N,1,1)p^{u_i}(1-P^D)+\sum_{j=1}^{t}\mathbf{p}(N,1,d_j)p^{u_i}r^{d_j} \\ &\quad +\sum_{j=1}^{t}\mathbf{p}(N,u_i,d_j)(1-r^{u_i})r^{d_j}+\mathbf{p}(N,u_i,1)(1-r^{u_i})(1-P^D)\end{aligned} \tag{5.26}$$

because as $\mathbf{p}(N,u_i,d_j)=\mathbf{p}(N,u_i,d_j)=0, i=1,\cdots,s, j=1,\cdots,t$. Therefore, the forms of Equations (5.22), (5.23), (5.24), and (5.26) are the same as the forms of transition equations for the internal states in Tolio et al. (2002). As a result, we conclude that under new upper boundary transition equations, states $(N-1,1,1)$, $(N-1,u_i,1)$, $(N-1,1,d_j)$, and $(N-1,u_i,d_j)$ should be considered as internal states. So, their probabilities can be expressed as the internal form. According to Tolio et al. (2002), we have

$$\begin{aligned}\mathbf{p}(N-1,1,1) &= \sum_{m=1}^{R}C_m X_m^{N-1}, \\ \mathbf{p}(N-1,1,d_j) &= \sum_{m=1}^{R}C_m X_m^{N-1}D_{j,m}, \\ \mathbf{p}(N-1,u_i,1) &= \sum_{m=1}^{R}C_m X_m^{N-1}U_{i,m}, \\ \mathbf{p}(N-1,u_i,d_j) &= \sum_{m=1}^{R}C_m X_m^{N-1}U_{i,m}D_{j,m}.\end{aligned} \tag{5.27}$$

Hence, the remaining unknown probabilities are $\mathbf{p}(N,1,1)$ and $\mathbf{p}(N,1,d_j), j=1,\cdots,t$. Comparing (5.20) and (5.25) we have

$$\mathbf{p}(N,1,1)=\frac{1-P^U}{p^{u_i}}\mathbf{p}(N-1,u_i,1). \tag{5.28}$$

Thus,

$$\mathbf{p}(N,1,1) = \frac{1-P^U}{p^{u_i}} \sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m}. \tag{5.29}$$

Equation (5.21) can be written as

$$\begin{aligned} r^{d_j}\mathbf{p}(N,1,d_j) &= \mathbf{p}(N,1,1)p^{d_j} + \mathbf{p}(N-1,1,1)(1-P^U)p^{d_j} \\ &\quad + \mathbf{p}(N-1,1,d_j)(1-P^U)(1-r^{d_j}) \\ &\quad + \sum_{i=1}^{s} \mathbf{p}(N-1,u_i,d_j) r^{u_i}(1-r^{d_j}) + \sum_{i=1}^{s} \mathbf{p}(N-1,u_i,1) r^{u_i} p^{d_j}. \end{aligned} \tag{5.30}$$

Plugging in the expressions for the terms on the right hand side, we find

$$\begin{aligned} r^{d_j}\mathbf{p}(N,1,d_j) &= \frac{p^{d_j}}{p^{u_i}}(1-P^U) \sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m} + \sum_{m=1}^{R} C_m X_m^{N-1} p^{d_j}(1-P^U) \\ &\quad + p^{d_j} \sum_{m=1}^{R} C_m X_m^{N-1} \sum_{i=1}^{s} U_{i,m} r^{u_i} + \sum_{i=1}^{s}\sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m} D_{j,m} r^{u_i}(1-r^{d_j}) \\ &\quad + \sum_{m=1}^{R} C_m X_m^{N-1} D_{j,m}(1-r^{d_j})(1-P^U) \\ &= \frac{p^{d_j}(1-P^U)}{p^{u_i}} \sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m} + p^{d_j} \sum_{m=1}^{R} C_m X_m^{N-1} \left[\sum_{i=s}^{t} U_{i,m} r^{u_i} + (1-P^U) \right] \\ &\quad + (1-r^{d_j}) \sum_{m=1}^{R} C_m X_m^{N-1} D_{j,m} \left[\sum_{i=1}^{s} U_{i,m} r^{u_i} + (1-P^U) \right]. \end{aligned} \tag{5.31}$$

Note that, according to Tolio et al. (2002),

$$\sum_{i=1}^{s} U_{i,m} r^{u_i} + (1-P^U) = X_m K_m. \tag{5.32}$$

Therefore, (5.31) becomes

$$\begin{aligned} r^{d_j}\mathbf{p}(N,1,d_j) &= \frac{p^{d_j}(1-P^U)}{p^{u_i}}\sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m} + p^{d_j}\sum_{m=1}^{R} C_m X_m^N K_m \\ &\quad +(1-r^{d_j})\sum_{m=1}^{R} C_m X_m^N D_{j,m} K_m. \end{aligned} \tag{5.33}$$

Thus,

$$\begin{aligned} \mathbf{p}(N,1,d_j) &= \frac{p^{d_j}(1-P^U)}{r^{d_j}p^{u_i}}\sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m} + \frac{p^{d_j}}{r^{d_j}}\sum_{m=1}^{R} C_m X_m^N K_m \\ &\quad +\frac{1-r^{d_j}}{r^{d_j}}\sum_{m=1}^{R} C_m X_m^N D_{j,m} K_m. \end{aligned} \tag{5.34}$$

As mentioned previously, we need another s equations from the modified upper boundary transition equations to find the normalization constants $C_1, C_2, \cdots, C_R$. To do this, either Equation (5.20) or (5.22) can be used, and they provide exactly the same results[7]. Assume that we use Equation (5.20) and plug in the expressions of $\mathbf{p}(N,1,1)$ and $\mathbf{p}(N,1,d_j), j=1,\cdots,t$, then

$$\begin{aligned} \frac{P^D+P^U-P^DP^U}{p^{u_i}}\sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m} &= \frac{P^D(1-P^U)}{p^{u_i}}\sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m} \\ &\quad +P^U\sum_{m=1}^{R} C_m X_m^N K_m \\ &\quad +\sum_{j=1}^{t}(1-r^{d_j})\sum_{m=1}^{R} C_m X_m^N D_{j,m} K_m, \end{aligned} \tag{5.35}$$

or

$$\frac{P^U}{p^{u_i}}\sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m} = P^D\sum_{m=1}^{R} C_m X_m^N K_m + \sum_{m=1}^{R} C_m X_m^N K_m \sum_{j=1}^{t} D_{j,m}(1-r^{d_j}), \quad i=1,\cdots,s. \tag{5.36}$$

[7]We have verified this with numerical experiments.

Therefore, (5.36) provides s equations for solving $C_1, C_2, \cdots, C_R$. Together with t other equations from either (5.18) or (5.19), we will have $s+t=R$ equations for R unknowns. On the other hand, if we instead use (5.22) to find $C_1, C_2, \cdots, C_R$, then

$$\begin{aligned}\left[P^U + P^D - P^U P^D\right] \sum_{m=1}^{R} C_m X_m^{N-1} &= \frac{1-P^D}{p^{u_i}} \sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m} \sum_{k=1}^{s} p^{u_k} r^{u_k} \\ &\quad + \sum_{m=1}^{R} C_m X_m^{N-1} \left[\sum_{k=1}^{s} \sum_{j=1}^{t} U_{k,m} D_{j,m} r^{u_k} r^{d_j} + \right. \\ &\quad \left. (1-P^U) \sum_{j=1}^{t} D_{j,m} r^{d_j} \right] \end{aligned} \tag{5.37}$$

for $i = 1, 2, \cdots, s$. Thus, (5.37) provides s equations for solving $C_1, C_2, \cdots, C_R$. We will still have $s+t$ equations for $C_1, C_2, \cdots, C_R$.

Summary of new boundary state probabilities

The new steady state probabilities of both lower and upper boundary states are summarized as follows. For $i = 1, \cdots, s$ and $j = 1, \cdots, t$,

$$\mathbf{p}(0,1,d_j) = 0,$$

$$\mathbf{p}(0,u_i,d_j) = 0,$$

$$\mathbf{p}(0,1,1) = \frac{1-P^D}{p^{d_j}} \sum_{m=1}^{R} C_m X_m D_{j,m},$$

$$\mathbf{p}(0,u_i,1) = \frac{p^{u_i}(1-P^D)}{r^{u_i} p^{d_j}} \sum_{m=1}^{R} C_m X_m D_{j,m} + \frac{p^{u_i}}{r^{u_i}} \sum_{m=1}^{R} \frac{C_m}{K_m} + \frac{1-r^{u_i}}{r^{u_i}} \sum_{m=1}^{R} C_m \frac{U_{i,m}}{K_m},$$

$$\mathbf{p}(1,1,1) = \sum_{m=1}^{R} C_m X_m,$$

$$\mathbf{p}(1,1,d_j) = \sum_{m=1}^{R} C_m X_m D_{j,m},$$

$$\mathbf{p}(1,u_i,1) = \sum_{m=1}^{R} C_m X_m U_{i,m},$$

$$\mathbf{p}(1,u_i,d_j) = \sum_{m=1}^{R} C_m X_m U_{i,m} D_{j,m},$$

$$\mathbf{p}(N-1,1,1) = \sum_{m=1}^{R} C_m X_m^{N-1},$$

$$\mathbf{p}(N-1,1,d_j) = \sum_{m=1}^{R} C_m X_m^{N-1} D_{j,m},$$

$$\mathbf{p}(N-1,u_i,1) = \sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m},$$

$$\mathbf{p}(N-1,u_i,d_j) = \sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m} D_{j,m},$$

$$\mathbf{p}(N,1,1) = \frac{1-P^U}{p^{u_i}} \sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m},$$

$$\mathbf{p}(N,1,d_j) = \frac{p^{d_j}(1-P^U)}{r^{d_j} p^{u_i}} \sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m} + \frac{p^{d_j}}{r^{d_j}} \sum_{m=1}^{R} C_m X_m^N K_m$$

$$+ \frac{1-r^{d_j}}{r^{d_j}} \sum_{m=1}^{R} C_m X_m^N D_{j,m} K_m,$$

$$\mathbf{p}(N,u_i,1) = 0,$$

$$\mathbf{p}(N,u_i,d_j) = 0.$$

Production rate of two-machine one-buffer building block with no-delay machine(s)

In addition to the modifications to steady state probabilities of boundary states, we need to modify the expression of the production rate as well. The no-delay properties of no-delay machines make the upstream machine less blocked and the downstream machine less starved. Therefore, the system has higher production rate. For the existing Tolio two-machine one-buffer model (Tolio et al. 2002) where both upstream and downstream machines are ordinary, the production rate of the line is calculated according to Gershwin (1994) as

$$E_1 = \mathbf{p}(\alpha_1(t+1) = 1, n(t) < N), \tag{5.38}$$

and

$$E_2 = \mathbf{p}(\alpha_2(t+1) = 1, n(t) > 0). \tag{5.39}$$

Because of the conservation of flow, $E_1 = E_2$. However, when we introduce no-delay machine(s), extra terms have to be considered. In the following, we derive the production rate of the system from the perspectives of the upstream machine and the downstream machine, respectively. In other words, we will derive both E_1 and E_2. Numerical experiments are provided to verify that $E_1 = E_2$.

Modifications of E_1 due to no-delay machine(s)

From a perspective of the upstream machine, the production rate can be calculated as

$$E_1 = \mathbf{p}(\alpha_1(t+1) = 1, n(t) < N) + \mathbf{p}(\alpha_1(t+1) = 1, \alpha_2(t+1) = 1, n(t) = N). \tag{5.40}$$

As compared to (5.38), the second term on the right hand side of (5.40) is an additional term. It is due to the no-delay property of the upstream machine, the downstream machine, or both. This is because as long as there is at least one no delay machine, the upstream machine can keep operating even when the buffer is full

due to the no-delay property. However, in the original model, the upstream machine is blocked given a full buffer.

Equation (5.40) is awkward because it involves states at two different time steps. Let us consider the two terms on the right hand side of (5.40) separately and transform them into two statements about the state of the system at a single time step.

The first term $\mathbf{p}(\alpha_1(t+1)=1, n(t)<N)$, denoted by FT_u, can be transformed according to the Total Probability Theorem (Bertsekas and Tsitsiklis 2008),

$$\begin{aligned} FT_u &= \mathbf{p}(\alpha_1(t+1)=1|\alpha_1(t)=1, n(t)<N)\mathbf{p}(\alpha_1(t)=1, n(t)<N) \\ &\quad + \sum_{i=1}^{s} \mathbf{p}(\alpha_1(t+1)=1|\alpha_1(t)=u_i, n(t)<N)\mathbf{p}(\alpha_1(t)=u_i, n(t)<N). \end{aligned} \tag{5.41}$$

Note that

$$\mathbf{p}(\alpha_1(t+1)=1|\alpha_1(t)=1, n(t)<N) = 1-\sum_{i=1}^{s} p^{u_i} = 1-P^U, \tag{5.42}$$

and

$$\mathbf{p}(\alpha_1(t+1)=1|\alpha_1(t)=u_i, n(t)<N) = r^{u_i}, i=1,\cdots,s. \tag{5.43}$$

Thus, (5.41) can be further simplified to

$$FT_u = (1-P^U)\mathbf{p}(\alpha_1(t)=1, n(t)<N) + \sum_{i=1}^{s} r^{u_i}\mathbf{p}(\alpha_1(t)=u_i, n(t)<N). \tag{5.44}$$

It can be shown from the Markov chain model that, for each $i=1,\cdots,s$,

$$\begin{aligned} r^{u_i}\mathbf{p}(\alpha_1(t)=u_i, n(t)<N) &= p^{u_i}\mathbf{p}(\alpha_1(t)=1, n(t)<N) \\ &\quad + p^{u_i}(1-P^D)\mathbf{p}(\alpha_1(t)=1, \alpha_2(t)=1, n(t)=N) \\ &\quad + \sum_{j=1}^{t} p^{u_i}r^{d_j}\mathbf{p}(\alpha_1(t)=1, \alpha_2(t)=d_j, n(t)=N). \end{aligned} \tag{5.45}$$

The left hand side is the probability that the system leaves the set of states $S_u = \{\alpha_1(t) = u_i \text{ and } n(t) < N\}$. This is because the only way the system can leave S_u is for the upstream machine to get repaired from the failure mode u_i. The right hand side is the probability that the system enters S_u. Substituting (5.45) into (5.44),

$$\begin{aligned} FT_u &= (1-P^U)\mathbf{p}(\alpha_1(t)=1, n(t)<N) + \sum_{i=1}^{s} p^{u_i}\mathbf{p}(\alpha_1(t)=1, n(t)<N) \\ &\quad + \sum_{i=1}^{s} p^{u_i}(1-P^D)\mathbf{p}(\alpha_1(t)=1, \alpha_2(t)=1, n(t)=N) \\ &\quad + \sum_{i=1}^{s}\sum_{j=1}^{t} p^{u_i} r^{d_j}\mathbf{p}(\alpha_1(t)=1, \alpha_2(t)=d_j, n(t)=N) \\ &= \mathbf{p}(\alpha_1(t)=1, n(t)<N) + P^U(1-P^D)\mathbf{p}(\alpha_1(t)=1, \alpha_2(t)=1, n(t)=N) \\ &\quad + \sum_{j=1}^{t} P^U r^{d_j}\mathbf{p}(\alpha_1(t)=1, \alpha_2(t)=d_j, n(t)=N). \end{aligned} \tag{5.46}$$

Dropping the t arguments, we finally find

$$FT_u = \sum_{n=0}^{N-1}\left[\mathbf{p}(n,1,1) + \sum_{j=1}^{t}\mathbf{p}(n,1,d_j)\right] + P^U(1-P^D)\mathbf{p}(N,1,1) + \sum_{j=1}^{t} P^U r^{d_j}\mathbf{p}(N,1,d_j). \tag{5.47}$$

Next, let us consider the second term $\mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1, n(t)=N)$, denoted by ST_u. Again, we need to convert this expression into a statement about the state of the system at a single time step. We apply the Total Probability Theorem again and find

$$
\begin{aligned}
ST_u &= \sum_{i=1}^{s} \mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=u_i, \alpha_2(t)=1, n(t)=N) \\
&\quad \times \mathbf{p}(\alpha_1(t)=u_i, \alpha_2(t)=1, n(t)=N) \\
&\quad + \sum_{i=1}^{s}\sum_{j=1}^{t} \mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=u_i, \alpha_2(t)=d_j, n(t)=N) \\
&\quad \times \mathbf{p}(\alpha_1(t)=u_i, \alpha_2(t)=d_j, n(t)=N) \\
&\quad + \sum_{j=1}^{t} \mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=1, \alpha_2(t)=d_j, n(t)=N) \\
&\quad \times \mathbf{p}(\alpha_1(t)=1, \alpha_2(t)=d_j, n(t)=N) \\
&\quad + \mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=1, \alpha_2(t)=1, n(t)=N) \\
&\quad \times \mathbf{p}(\alpha_1(t)=1, \alpha_2(t)=1, n(t)=N).
\end{aligned} \tag{5.48}
$$

Note that, for $i = 1, \cdots, s$ and $j = 1, \cdots, t$,

$$
\mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=u_i, \alpha_2(t)=1, n(t)=N) = r^{u_i}(1-P^D), \tag{5.49}
$$

$$
\mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=u_i, \alpha_2(t)=d_j, n(t)=N) = r^{u_i} r^{d_j}, \tag{5.50}
$$

$$
\mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=1, \alpha_2(t)=d_j, n(t)=N) = (1-P^U) r^{d_j}, \tag{5.51}
$$

and

$$
\mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=1, \alpha_2(t)=1, n(t)=N) = (1-P^U)(1-P^D). \tag{5.52}
$$

Substituting (5.49) to (5.52) into (5.48),

$$\begin{aligned} ST_u &= \sum_{i=1}^{s} r^{u_i}(1-P^D)\mathbf{p}(\alpha_1(t)=u_i,\alpha_2(t)=1,n(t)=N) \\ &+\sum_{i=1}^{s}\sum_{j=1}^{t} r^{u_i}r^{d_j}\mathbf{p}(\alpha_1(t)=u_i,\alpha_2(t)=d_j,n(t)=N) \\ &+\sum_{j=1}^{t}(1-P^U)r^{d_j}\mathbf{p}(\alpha_1(t)=1,\alpha_2(t)=d_j,n(t)=N) \\ &+(1-P^U)(1-P^D)\mathbf{p}(\alpha_1(t)=1,\alpha_2(t)=1,n(t)=N). \end{aligned} \tag{5.53}$$

Dropping the t arguments leaves

$$\begin{aligned} ST_u &= \sum_{i=1}^{s} r^{u_i}(1-P^D)\mathbf{p}(N,u_i,1)+\sum_{i=1}^{s}\sum_{j=1}^{t} r^{u_i}r^{d_j}\mathbf{p}(N,u_i,d_j) \\ &+\sum_{j=1}^{t}(1-P^U)r^{d_j}\mathbf{p}(N,1,d_j)+(1-P^U)(1-P^D)\mathbf{p}(N,1,1). \end{aligned} \tag{5.54}$$

Combining (5.47) and (5.54),

$$\begin{aligned} E_1 &= \mathbf{p}(\alpha_1(t+1)=1,n(t)<N)+\mathbf{p}(\alpha_1(t+1)=1,\alpha_2(t+1)=1,n(t)=N) \\ &= \sum_{n=0}^{N-1}\left[\mathbf{p}(n,1,1)+\sum_{j=1}^{t}\mathbf{p}(n,1,d_j)\right]+(1-P^D)\mathbf{p}(N,1,1)+\sum_{j=1}^{t} r^{d_j}\mathbf{p}(N,1,d_j) \\ &+\sum_{i=1}^{s} r^{u_i}(1-P^D)\mathbf{p}(N,u_i,1)+\sum_{i=1}^{s}\sum_{j=1}^{t} r^{u_i}r^{d_j}\mathbf{p}(N,u_i,d_j). \end{aligned} \tag{5.55}$$

We can further simplify (5.55) by noticing that states $(N,u_i,1), i=1,\cdots,s$ and $(N,u_i,d_j), i=1,\cdots,s, j=1,\cdots,j$ are transient. Thus their steady state probabilities are zero. Thus, (5.55) can be simplified to

$$E_1 = \sum_{n=0}^{N-1} \left[\mathbf{p}(n,1,1) + \sum_{j=1}^{t} \mathbf{p}(n,1,d_j) \right] + (1-P^D)\mathbf{p}(N,1,1) + \sum_{j=1}^{t} r^{d_j} \mathbf{p}(N,1,d_j). \tag{5.56}$$

Combining (5.45) and (5.56), we see that $r^{u_i} D_1^{u_i} = p^{u_i} E_1$ where $D_1^{u_i} = \mathbf{p}(\alpha_1 = u_i, n < N), i = 1, \cdots, s$. This is expected as it says the repair frequency from failure mode u_i equals the failure frequency into that failure mode. This equality is proved by Gershwin (1994).

Modifications of E_2 due to no-delay machine(s)

From a perspective of the downstream machine, the production rate can be calculated as

$$E_2 = \mathbf{p}(\alpha_2(t+1) = 1, n(t) > 0) + \mathbf{p}(\alpha_1(t+1) = 1, \alpha_2(t+1) = 1, n(t) = 0). \tag{5.57}$$

As compared to (5.39), the second term on the right hand side of (5.57) is an additional term. It is due to the no-delay property of the upstream machine, the downstream machine, or both. As before, Equation (5.57) is awkward because it involves states at two different time steps. Let us consider the two terms on the right hand side of (5.57) separately and transform them into two statements about the state of the system at a single time step.

The first term $\mathbf{p}(\alpha_2(t+1) = 1, n(t) > 0)$, denoted by FT_d, can be transformed according to the Total Probability Theorem,

$$\begin{aligned} FT_d &= \mathbf{p}(\alpha_2(t+1) = 1 | \alpha_2(t) = 1, n(t) > 0)\mathbf{p}(\alpha_2(t) = 1, n(t) > 0) \\ &\quad + \sum_{j=1}^{t} \mathbf{p}(\alpha_2(t+1) = 1 | \alpha_2(t) = d_j, n(t) > 0)\mathbf{p}(\alpha_2(t) = d_j, n(t) > 0). \end{aligned} \tag{5.58}$$

Note that

$$\mathbf{p}(\alpha_2(t+1) = 1 | \alpha_2(t) = 1, n(t) > 0) = 1 - \sum_{j=1}^{t} p^{d_j} = 1 - P^D, \tag{5.59}$$

$$\mathbf{p}(\alpha_2(t+1)=1|\alpha_2(t)=d_j, n(t)>0)=r^{d_j}, j=1,\cdots,t. \tag{5.60}$$

Thus, (5.58) can be further simplified as

$$FT_d=(1-P^D)\mathbf{p}(\alpha_2(t)=1,n(t)>0)+\sum_{j=1}^{t} r^{d_j}\mathbf{p}(\alpha_2(t)=d_j,n(t)>0). \tag{5.61}$$

It can be shown that, for each $j=1,\cdots,t$,

$$\begin{aligned} r^{d_j}\mathbf{p}(\alpha_2(t)=d_j,n(t)>0) &= p^{d_j}\mathbf{p}(\alpha_2(t)=1,n(t)>0) \\ &\quad +p^{d_j}(1-P^U)\mathbf{p}(\alpha_1(t)=1,\alpha_2(t)=1,n(t)=0) \\ &\quad +\sum_{i=1}^{s} r^{u_i}p^{d_j}\mathbf{p}(\alpha_1(t)=u_i,\alpha_2(t)=1,n(t)=0). \end{aligned} \tag{5.62}$$

The left hand side is the probability that the system leaves the set of states $S_d=\{\alpha_2(t)=d_j \text{ and } n(t)>0\}$. This is because the only way the system can leave S_d is for the downstream machine to get repaired from the failure mode d_j. The righthand side is the probability that the system enters S_d.

Substituting (5.62) into (5.61),

$$\begin{aligned} FT_d &= (1-P^D)\mathbf{p}(\alpha_2(t)=1,n(t)>0)+\sum_{j=1}^{t} p^{d_j}\mathbf{p}(\alpha_2(t)=1,n(t)>0) \\ &\quad +\sum_{j=1}^{t} p^{d_j}(1-P^U)\mathbf{p}(\alpha_1(t)=1,\alpha_2(t)=1,n(t)=0) \\ &\quad +\sum_{j=1}^{t}\sum_{i=1}^{s} r^{u_i}p^{d_j}\mathbf{p}(\alpha_1(t)=u_i,\alpha_2(t)=1,n(t)=0) \\ &= \mathbf{p}(\alpha_2(t)=1,n(t)>0)+P^D(1-P^U)\mathbf{p}(\alpha_1(t)=1,\alpha_2(t)=1,n(t)=0) \\ &\quad +\sum_{i=1}^{s} P^D r^{u_i}\mathbf{p}(\alpha_1(t)=u_i,\alpha_2(t)=1,n(t)=0). \end{aligned} \tag{5.63}$$

Dropping the t arguments, we finally find

$$FT_d = \sum_{n=1}^{N} \left[\mathbf{p}(n,1,1) + \sum_{i=1}^{s} \mathbf{p}(n,u_i,1) \right] + P^D(1-P^U)\mathbf{p}(0,1,1) + \sum_{i=1}^{s} P^D r^{u_i} \mathbf{p}(0,u_i,1). \tag{5.64}$$

Next, let us consider the second term $\mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1, n(t)=0)$, denoted by ST_d. Again, we need to convert this expression into a statement about the state of the system at a single time step. According to the Total Probability Theorem,

$$\begin{aligned} ST_d \;=\; & \sum_{i=1}^{s} \mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=u_i, \alpha_2(t)=1, n(t)=0) \\ & \quad \times \mathbf{p}(\alpha_1(t)=u_i, \alpha_2(t)=1, n(t)=0) \\ & + \sum_{i=1}^{s} \sum_{j=1}^{t} \mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=u_i, \alpha_2(t)=d_j, n(t)=0) \\ & \quad \times \mathbf{p}(\alpha_1(t)=u_i, \alpha_2(t)=d_j, n(t)=0) \\ & + \sum_{j=1}^{t} \mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=1, \alpha_2(t)=d_j, n(t)=0) \\ & \quad \times \mathbf{p}(\alpha_1(t)=1, \alpha_2(t)=d_j, n(t)=0) \\ & + \mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=1, \alpha_2(t)=1, n(t)=0) \\ & \quad \times \mathbf{p}(\alpha_1(t)=1, \alpha_2(t)=1, n(t)=0). \end{aligned} \tag{5.65}$$

Note that, for $i = 1, \cdots, s$ and $j = 1, \cdots, t$,

$$\mathbf{p}\left(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=u_i, \alpha_2(t)=1, n(t)=0\right) = r^{u_i}\left(1-P^D\right), \tag{5.66}$$

$$\mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=u_i, \alpha_2(t)=d_j, n(t)=0) = r^{u_i} r^{d_j}, \tag{5.67}$$

$$\mathbf{p}(\alpha_1(t+1)=1, \alpha_2(t+1)=1 | \alpha_1(t)=1, \alpha_2(t)=d_j, n(t)=0) = \left(1-P^U\right) r^{d_j}, \tag{5.68}$$

$$\mathbf{p}(\alpha_1(t+1)=1,\alpha_2(t+1)=1|\alpha_1(t)=1,\alpha_2(t)=1,n(t)=0)=\left(1-P^U\right)\left(1-P^D\right). \tag{5.69}$$

Substituting (5.66) $\sim$ (5.69) into (5.65), we find

$$\begin{aligned} ST_d &= \sum_{i=1}^{s} r^{u_i}\left(1-P^D\right)\mathbf{p}(\alpha_1(t)=u_i,\alpha_2(t)=1,n(t)=0) \\ &+\sum_{i=1}^{s}\sum_{j=1}^{t} r^{u_i}r^{d_j}\mathbf{p}(\alpha_1(t)=u_i,\alpha_2(t)=d_j,n(t)=0) \\ &+\sum_{j=1}^{t}\left(1-P^U\right)r^{d_j}\mathbf{p}(\alpha_1(t)=1,\alpha_2(t)=d_j,n(t)=0) \\ &+\left(1-P^U\right)\left(1-P^D\right)\mathbf{p}(\alpha_1(t)=1,\alpha_2(t)=1,n(t)=0). \end{aligned} \tag{5.70}$$

Dropping the t arguments, we finally find

$$\begin{aligned} ST_d &= \sum_{i=1}^{s} r^{u_i}\left(1-P^D\right)\mathbf{p}(0,u_i,1)+\sum_{i=1}^{s}\sum_{j=1}^{t} r^{u_i}r^{d_j}\mathbf{p}(0,u_i,d_j) \\ &+\sum_{j=1}^{t}\left(1-P^U\right)r^{d_j}\mathbf{p}(0,1,d_j)+\left(1-P^U\right)\left(1-P^D\right)\mathbf{p}(0,1,1). \end{aligned} \tag{5.71}$$

Combining (5.64) and (5.71)

$$\begin{aligned} E_2 &= \mathbf{p}(\alpha_2(t+1)=1,n(t)>0)+\mathbf{p}(\alpha_1(t+1)=1,\alpha_2(t+1)=1,n(t)=0) \\ &= \sum_{n=1}^{N}\left[\mathbf{p}(n,1,1)+\sum_{i=1}^{s}\mathbf{p}(n,u_i,1)\right]+(1-P^U)\mathbf{p}(0,1,1)+\sum_{i=1}^{s} r^{u_i}\mathbf{p}(0,u_i,1) \\ &+\sum_{j=1}^{t}(1-P^U)r^{d_j}\mathbf{p}(0,1,d_j)+\sum_{i=1}^{s}\sum_{j=1}^{t} r^{u_i}r^{d_j}\mathbf{p}(0,u_i,d_j). \end{aligned} \tag{5.72}$$

We further simplify (5.72) by noticing that states $(0,1,d_j), j = 1,\cdots,t$ and

$(0, u_i, d_j), i = 1, \cdots, s, j = 1, \cdots, j$ are transient, and therefore their steady state probabilities are zero. Thus, (5.72) can be simplified to

$$E_2 = \sum_{n=1}^{N} \left[\mathbf{p}(n,1,1) + \sum_{i=1}^{s} \mathbf{p}(n,u_i,1) \right] + (1-P^U)\mathbf{p}(0,1,1) + \sum_{i=1}^{s} r^{u_i}\mathbf{p}(0,u_i,1). \quad (5.73)$$

Combining (5.62) and (5.73), we see that $r^{d_j} D_2^{d_j} = p^{d_j} E_2$ where $D_2^{d_j} = \mathbf{p}(\alpha_2 = d_j, n > 0), j = 1, \cdots, t$. This is expected as it says the repair frequency from failure mode d_j equals the failure frequency into that failure mode.

Summary of production rate of two-machine one-buffer building blocks with no-delay machine(s)

The production rates E_1 and E_2 of a building block are summarized as follows:

$$\begin{cases} E_1 = \sum_{n=0}^{N-1} \left[\mathbf{p}(n,1,1) + \sum_{j=1}^{t} \mathbf{p}(n,1,d_j) \right] + \left(1-P^D\right)\mathbf{p}(N,1,1) + \sum_{j=1}^{t} r^{d_j}\mathbf{p}(N,1,d_j), \\ E_2 = \sum_{n=1}^{N} \left[\mathbf{p}(n,1,1) + \sum_{i=1}^{s} \mathbf{p}(n,u_i,1) \right] + \left(1-P^U\right)\mathbf{p}(0,1,1) + \sum_{i=1}^{s} r^{u_i}\mathbf{p}(0,u_i,1). \end{cases} \quad (5.74)$$

Because of the conservation of flow, we expect $E_1 = E_2$. Four numerical experiments are provided in Table 5.1 to show the equivalence.

Note that Case 1 is a symmetric line. Case 2 and Case 3 have the same machines but different buffer spaces. Case 4 has multiple failure modes for both machines. The results for both ordinary machine case and no-delay machine case are considered. It can be seen from the four cases that

- $E_1 = E_2$, which verifies the conservation of flow;
- with no-delay machine(s), the production rate of the two-machine one-buffer building block is higher than that of the otherwise identical system without no-delay machines;
- from Cases 2 and 3, the increment of production rate due to no-delay machines

Table 5.1: Numerical evidence of $E_1 = E_2$

case						E_1	E_2	$\bar{n}$
r^{u_1}	p^{u_1}	r^{d_1}	p^{d_1}	N	Ordinary	.870541	.870541	10.000000
.1	.01	.1	.01	20	No-delay	.872354	.872354	10.000000
r^{u_1}	p^{u_1}	r^{d_1}	p^{d_1}	N	Ordinary	.925325	.925325	46.457620
.273	.0114	.09	.0072	60	No-delay	.925389	.925389	47.278691
r^{u_1}	p^{u_1}	r^{d_1}	p^{d_1}	N	Ordinary	.903442	.903442	2.917237
.273	.0114	.09	.0072	6	No-delay	.907095	.907095	2.982608
r^{u_1}	p^{u_1}	r^{d_1}	p^{d_1}	N				
.11	.04	.12	.02	17	Ordinary	.595136	.595136	4.615708
r^{u_2}	p^{u_2}	r^{d_2}	p^{d_2}		No-delay	.597540	.597540	4.262650
.08	.02	.1	.01					

is more significant when buffer size N is small. This is because the no-delay properties of machines only change the boundary conditions. When N is small, the system spends more time in boundary states. This is why the change in production rate is more obvious.

Table 5.2: Parameters of two-machine one-buffer building blocks with no-delay machine(s)

case	1	2	3	4
r^{u_1}	.1	.2	.15	.16
r^{u_2}	.14	.1	.12	.18
p^{u_1}	.01	.009	.015	.02
p^{u_2}	.01	.011	.01	.02
r^{d_1}	.2	.12	.19	.16
r^{d_2}	.24	.11	.19	.18
p^{d_1}	.03	.02	.01	.02
p^{d_2}	.01	.01	.01	.02
N	20	25	36	26

In addition to the results above, we provide another set of numerical experiments, which compare the analytical solutions against simulation. They are summarized in Tables 5.2 and 5.3. ($E_1 = E_2$ is not reported in the table, but it is verified by all these experiments.) In all cases, both the upstream machine and the downstream have two

failure modes. The results show the consistency between the analytical solutions and simulation results. The simulation is written exclusively for this purpose with exactly the same underlying two-machine one-buffer blocks with no-delay machines. For each experiment, the length of simulation is 21,000,000 time steps with the first 1,000,000 time steps being the warm up period. In addition for each experiment, the number of simulation runs is 20. We report both the mean and the standard deviation for the simulation results. These results indicate the accuracy of the analytical solutions.

Table 5.3: Numerical results of two-machine one-buffer building blocks with no-delay machine(s)

		Analytical	Sim – Mean	Sim – Stdev
Case 1	Production rate	.802973	.803032	.000302
	Average inventory	11.708614	11.717033	.013868
Case 2	Production rate	.775707	.775756	.000237
	Average inventory	17.022497	17.0142517	.027734
Case 3	Production rate	.840486	.840501	.000282
	Average inventory	9.537670	9.525308	.056079
Case 4	Production rate	.772196	.772139	.000309
	Average inventory	13.000000	13.005549	.028892

5.4.2 Two-Machine One-Buffer Building Block with a Buffer of Size One

In the previous section, we reveal the first problem in Werner's algorithm. In this section, the second problem is presented. We consider the system shown in Figure 5-16 again. Recall that it shows a modified system after the elimination of buffer thresholds.

In particular, we assume that the size of each buffer is 20. But the loop invariant is 21 (rather than 27 as previously). Therefore, in the modified system, the sizes of Buffers B_{21}, B_{31}, and B_{51} are 1. In other words, given the specific buffer sizes and the loop invariant of the system, it is possible to have buffers of size 1 after eliminating the thresholds. This means that there could be a set of two-machine one-

buffer building blocks for those buffers whose sizes are 1. However, the Markov chain model developed by Tolio et al. (2002) requires the buffer size $N \geq 2$ as it defines boundary states and internal states, and builds the model based on the transitions among those states. Therefore, the analytical solutions of Tolio et al. (2002) do not apply to the case where the size of the buffer is 1. As a result, in order to use the existing two-machine one-buffer building block model, two approximate alternatives can be considered when there are building blocks whose buffer sizes are 1:

1. keep $N = 1$ but use the continuous time continuous material model developed by Levantesi et al. (1999a) to approximate the discrete time discrete material model under consideration;

2. set $N = 2$ and still use the discrete time discrete material model. Werner's algorithm adopts this approach (Werner 2001).

We realize that whenever a buffer of size 1 appears after the thresholds are eliminated, either its upstream machine or its downstream machine or both of them are no-delay machines. This is because the size 1 buffer is generated by inserting perfectly reliable machines. Therefore, in the two-machine one-buffer building blocks that contain size 1 buffers, we have to consider the no-delay properties of machines. Although the two alternatives mentioned above provide good approximate evaluation results, it is desirable to develop analytical solutions for building blocks with buffers of size one and no-delay machines. We arrive at this conclusion by observing again from the Batman effect shown in Figure 5-7.

Recall that Figure 5-7 illustrates the production rate as a function of loop invariant for a closed three-machine three-buffer loop with identical machines and identical buffers. It can be seen that the production rate of the system when $I = 11$ is greater than the production rate of the system when $I = 12$, which is different from the simulation result. This is expected because Werner used the second approach to deal with buffers of size 1. As it is illustrated in Figure 5-25, when $I = 11$ there are six buffers whose sizes are 9, 1, 9, 1, 9, and 1 respectively after inserting perfectly reliable machines (denoted by $M^\star$ in the figure) and eliminating buffer thresholds. After that,

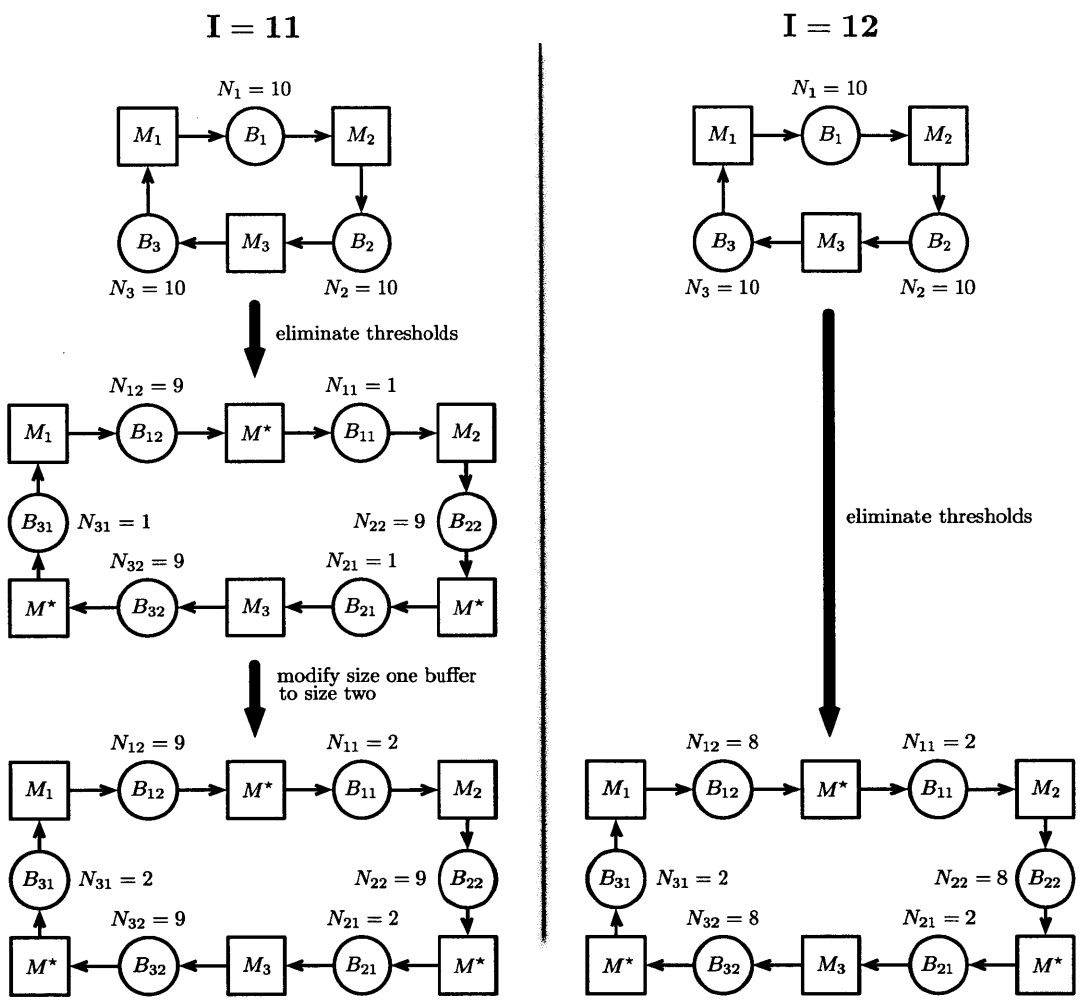


Figure 5-25: **I=11** vs **I=12** for a closed three-machine three-buffer loop

all buffers of size 1 are modified to size 2. In other words, when $I = 11$, there are six building blocks whose buffer sizes are 9, 2, 9, 2, 9, and 2, respectively. However, on the other hand, when $I = 12$, there are also six building blocks after eliminating buffer thresholds but their sizes are 8, 2, 8, 2, 8, and 2, respectively. Clearly, three building blocks when $I = 11$ have larger buffer spaces than the corresponding three building blocks when $I = 12$. As a results, there are (incorrectly) less blockage and starvation when $I = 11$. This explains why the production rate is larger when $I = 12$. However, we know that this is incorrect. The three size 1 buffers are enlarged to size 2 arbitrarily simply because such modifications are needed before the existing two-

machine one-buffer building block model can be used. This is why a new Markov chain model and new analytical solutions for two-machine one-buffer building blocks with size 1 buffers and no-delay machines are not only desirable but also necessary.

Analytical solutions

We derive the analytical solutions for two-machine one-buffer building blocks with size 1 buffers and no-delay machine(s). Note that the upstream machine, the downstream machine, or both of them can be no-delay machines. This makes no difference due to the two assumptions we make about no-delay machines in Section 5.4.1. Keeping using the notation in Tolio et al. (2002), there are all together eight sets of states for such a building block. They are

$$\begin{array}{cccc} (0,1,1) & (0,1,d_j) & (0,u_i,d_j) & (0,u_i,1) \\ (1,1,1) & (1,1,d_j) & (1,u_i,d_j) & (1,u_i,1) \end{array} \tag{5.75}$$

We first notice that the set of states $(0,1,d_j), j = 1,\cdots,t$ cannot be reached by any other set of states. Therefore, it is transient. In addition, $(0,u_i,d_j), i = 1,\cdots,s, j = 1,\cdots,j$ can be reached only from $(0,1,d_j)$ or itself, so it is also transient. Similarly, $(1,u_i,1), i = 1,\cdots,s$ cannot be reached by any other state. Therefore, it is also transient. Moreover, $(1,u_i,d_j), i = 1,\cdots,s, j = 1,\cdots,t$ can be reached only from $(1,u_i,1)$ or itself, so it is also transient. Hence, the four sets $(0,u_i,d_j)$, $(0,1,d_j)$, $(1,u_i,1)$, and $(1,u_i,d_j), i = 1,\cdots,s, j = 1,\cdots,t$ are transient. Therefore, eliminating all transient states (since their steady state probabilities are 0), the transition equations for the other four sets of recurrent states are

$$\mathbf{p}(0,1,1) = \mathbf{p}(0,1,1)(1-P^U)(1-P^D) + \sum_{i=1}^{s} \mathbf{p}(0,u_i,1) r^{u_i}(1-P^D), \tag{5.76}$$

$$\mathbf{p}(0,u_i,1) = \mathbf{p}(0,1,1)p^{u_i} + \mathbf{p}(0,u_i,1)(1-r^{u_i}) + \mathbf{p}(1,1,1)p^{u_i}(1-P^D) + \sum_{j=1}^{t} \mathbf{p}(1,1,d_j)p^{u_i}r^{d_j}, \tag{5.77}$$

$$\mathbf{p}(1,1,1) = \mathbf{p}(1,1,1)(1-P^U)(1-P^D) + \sum_{j=1}^{t} \mathbf{p}(1,1,d_j)(1-P^U)r^{d_j}, \tag{5.78}$$

and

$$\mathbf{p}(1,1,d_j) = \mathbf{p}(0,1,1)(1-P^U)p^{d_j} + \sum_{i=1}^{s} \mathbf{p}(0,u_i,1)r^{u_i}p^{d_j} + \mathbf{p}(1,1,1)p^{d_j} + \mathbf{p}(1,1,d_j)(1-r^{d_j}). \tag{5.79}$$

Now, we can solve for Equations (5.76), (5.77), (5.78), and (5.79) to find the steady state probabilities of all recurrent states. By (5.76) we know that

$$\sum_{i=1}^{s} \mathbf{p}(0,u_i,1)r^{u_i} = \frac{P^U + P^D - P^U P^D}{1-P^D}\mathbf{p}(0,1,1). \tag{5.80}$$

Plugging (5.80) into (5.79),

$$\begin{aligned} r^{d_j}\mathbf{p}(1,1,d_j) &= \mathbf{p}(0,1,1)\left[(1-P^U)p^{d_j} + \frac{P^U + P^D - P^U P^D}{1-P^D}p^{d_j}\right] + \mathbf{p}(1,1,1)p^{d_j} \\ &= \mathbf{p}(0,1,1)\frac{p^{d_j}}{1-P^D} + \mathbf{p}(1,1,1)p^{d_j}. \end{aligned} \tag{5.81}$$

Therefore,

$$\mathbf{p}(1,1,d_j) = \frac{p^{d_j}}{r^{d_j}}\left[\frac{1}{1-P^D}\mathbf{p}(0,1,1) + \mathbf{p}(1,1,1)\right], \quad j = 1,\cdots,t. \tag{5.82}$$

Similarly, by (5.78) we know that

$$\sum_{j=1}^{t} \mathbf{p}(1,1,d_j)r^{d_j} = \frac{P^U + P^D - P^U P^D}{1-P^U}\mathbf{p}(1,1,1). \tag{5.83}$$

Plugging (5.83) into (5.77),

$$\begin{aligned} r^{u_i}\mathbf{p}(0,u_i,1) &= \mathbf{p}(1,1,1)\left[(1-P^D)p^{u_i}+\frac{P^U+P^D-P^UP^D}{1-P^U}p^{u_i}\right]+\mathbf{p}(0,1,1)p^{u_i} \\ &= \mathbf{p}(1,1,1)\frac{p^{u_i}}{1-P^U}+\mathbf{p}(0,1,1)p^{u_i}. \end{aligned} \tag{5.84}$$

Therefore,

$$\mathbf{p}(0,u_i,1)=\frac{p^{u_i}}{r^{u_i}}\left[\frac{1}{1-P^U}\mathbf{p}(1,1,1)+\mathbf{p}(0,1,1)\right],\quad i=1,\cdots,s. \tag{5.85}$$

In addition, substituting (5.85) into (5.76) yields

$$\mathbf{p}(0,1,1)=\mathbf{p}(0,1,1)(1-P^U)(1-P^D)+(1-P^D)P^U\left[\frac{1}{1-P^U}\mathbf{p}(1,1,1)+\mathbf{p}(0,1,1)\right], \tag{5.86}$$

or

$$P^D\mathbf{p}(0,1,1)=\frac{(1-P^D)P^U}{1-P^U}\mathbf{p}(1,1,1). \tag{5.87}$$

Thus,

$$\mathbf{p}(0,1,1)=\frac{(1-P^D)P^U}{(1-P^U)P^D}\mathbf{p}(1,1,1). \tag{5.88}$$

As a check, if we substitute (5.82) into (5.78), we will derive exactly the same relationship between $\mathbf{p}(1,1,1)$ and $\mathbf{p}(0,1,1)$. Substituting (5.88) into (5.82), we find

$$\begin{aligned} \mathbf{p}(1,1,d_j) &= \frac{p^{d_j}}{r^{d_j}}\left[\frac{P^U}{(1-P^U)P^D}+1\right]\mathbf{p}(1,1,1) \\ &= \frac{p^{d_j}}{r^{d_j}}\frac{P^U+P^D-P^UP^D}{(1-P^U)P^D}\mathbf{p}(1,1,1),\quad j=1,\cdots,t. \end{aligned} \tag{5.89}$$

Similarly, substituting (5.88) into (5.85) gives,

$$\begin{aligned}\mathbf{p}(0,u_i,1) &= \frac{p^{u_i}}{r^{u_i}}\left[\frac{1}{1-P^U}+\frac{(1-P^D)P^U}{(1-P^U)P^D}\right]\mathbf{p}(1,1,1) \\ &= \frac{p^{u_i}}{r^{u_i}}\frac{P^U+P^D-P^UP^D}{(1-P^U)P^D}\mathbf{p}(1,1,1), \quad i=1,\cdots,s.\end{aligned} \tag{5.90}$$

Finally, the normalization condition requires that

$$\mathbf{p}(1,1,1)+\mathbf{p}(0,1,1)+\sum_{i=1}^{s}\mathbf{p}(0,u_i,1)+\sum_{j=1}^{t}\mathbf{p}(1,1,d_j)=1. \tag{5.91}$$

or, plugging (5.88), (5.89), and (5.90),

$$\left[1+\frac{(1-P^D)P^U}{(1-P^U)P^D}+\frac{P^U+P^D-P^UP^D}{(1-P^U)P^D}\left(\sum_{i=1}^{s}\frac{p^{u_i}}{r^{u_i}}+\sum_{j=1}^{t}\frac{p^{d_j}}{r^{d_j}}\right)\right]\mathbf{p}(1,1,1)=1. \tag{5.92}$$

Thus,

$$\mathbf{p}(1,1,1)=\left[1+\frac{(1-P^D)P^U}{(1-P^U)P^D}+\frac{P^U+P^D-P^UP^D}{(1-P^U)P^D}\left(\sum_{i=1}^{s}\frac{p^{u_i}}{r^{u_i}}+\sum_{j=1}^{t}\frac{p^{d_j}}{r^{d_j}}\right)\right]^{-1}, \tag{5.93}$$

and the steady state probabilities of other sets of recurrent states are

$$\mathbf{p}(0,1,1)=\frac{(1-P^D)P^U}{(1-P^U)P^D}\left[1+\frac{(1-P^D)P^U}{(1-P^U)P^D}+\frac{P^U+P^D-P^UP^D}{(1-P^U)P^D}\left(\sum_{i=1}^{s}\frac{p^{u_i}}{r^{u_i}}+\sum_{j=1}^{t}\frac{p^{d_j}}{r^{d_j}}\right)\right]^{-1}, \tag{5.94}$$

$$\begin{aligned}\mathbf{p}(0,u_i,1) &= \frac{p^{u_i}}{r^{u_i}}\frac{P^U+P^D-P^UP^D}{(1-P^U)P^D}\times \\ &\left[1+\frac{(1-P^D)P^U}{(1-P^U)P^D}+\frac{P^U+P^D-P^UP^D}{(1-P^U)P^D}\left(\sum_{i=1}^{s}\frac{p^{u_i}}{r^{u_i}}+\sum_{j=1}^{t}\frac{p^{d_j}}{r^{d_j}}\right)\right]^{-1} \\ &\qquad i=1,\cdots,s.\end{aligned} \tag{5.95}$$

$$\mathbf{p}(1,1,d_j) = \frac{p^{d_j}}{r^{d_j}} \frac{P^U + P^D - P^U P^D}{(1-P^U)P^D} \times$$

$$\left[1 + \frac{(1-P^D)P^U}{(1-P^U)P^D} + \frac{P^U + P^D - P^U P^D}{(1-P^U)P^D} \left(\sum_{i=1}^{s} \frac{p^{u_i}}{r^{u_i}} + \sum_{j=1}^{t} \frac{p^{d_j}}{r^{d_j}} \right) \right]^{-1}$$

$$j = 1, \cdots, t. \tag{5.96}$$

The production rate of the building block, $P(N)$, can be calculated as

$$P(N) = \mathbf{p}(0,1,1) + (1-P^D)\mathbf{p}(1,1,1) + \sum_{j=1}^{t} r_j \mathbf{p}(1,1,d_j), \tag{5.97}$$

and the average inventory of the building block, $\bar{n}$, can be calculated as

$$\bar{n} = \mathbf{p}(1,1,1) + \sum_{j=1}^{t} \mathbf{p}(1,1,d_j). \tag{5.98}$$

The same four two-machine one-buffer building blocks as in Section 5.4.1 are provided (with exception that the sizes of buffers are changed to 1) to show the accuracy of the analytical solutions against the simulation results. They are listed in Tables 5.4 and 5.5, which demonstrate the accuracy of the analytical solutions.

Table 5.4: Parameters of two-machine one-buffer building blocks with buffers of size 1 and no-delay machine(s)

case	1	2	3	4
r^{u_1}	.1	.2	.15	.16
r^{u_2}	.14	.1	.12	.18
p^{u_1}	.01	.009	.015	.02
p^{u_2}	.01	.011	.01	.02
r^{d_1}	.2	.12	.19	.16
r^{d_2}	.24	.11	.19	.18
p^{d_1}	.03	.02	.01	.02
p^{d_2}	.01	.01	.01	.02
N	1	1	1	1

Table 5.5: Numerical results of two-machine one-buffer building blocks with buffers of size 1 and no-delay machine(s)

		Analytical	Sim – Mean	Sim – Stdev
Case 1	Production rate	.740970	.740922	.000243
	Average inventory	.632662	.632780	.000783
Case 2	Production rate	.714066	.714068	.000423
	Average inventory	.608897	.608613	.000729
Case 3	Production rate	.782864	.782822	.000322
	Average inventory	.425460	.425566	.000733
Case 4	Production rate	.688793	.688869	.000229
	Average inventory	.500000	.499934	.000530

Next, we provide another set of experiments (Table 5.6) that compare the new analytical solutions for building blocks with buffers of size 1 and the other two alternatives. The simulation results are also provided. In addition to the production rate and average inventory, we report the probability of starvation of the downstream machine (p_s) and the probability of blockage (p_b) of the upstream machine as well. The reason why we care about these two quantities is that they are used in the decomposition (Tolio and Matta 1998). In particular, p_s and p_b of one building block are used to calculate the failure mode parameters of some other building blocks by iteration in the decomposition approach. An inaccuracy in these two quantities may lead to an inaccuracy of the decomposition. In Table 5.6, the parameters of four cases are shown on the left, while the results of these cases are shown on the right. For each case, "New", "Continuous", and "Dis ($N = 2$)"correspond to the new analytical solutions, the continuous model approximation (alternative 1), and the existing discrete model with $N = 2$ (alternative 2), respectively. The following observations can be made according to these results:

- For all four quantities of interest, the analytical solutions of the new model are very accurate as compared to the simulation results.

- In terms of the production rate and average inventory, the continuous model is indeed a good approximation. The production rate from "Dis ($N = 2$)"is

Table 5.6: Comparison of the modified algorithm with the other two approximate approaches and simulation

case						$P(N)$	$\bar{n}$	p_s	p_b
					New	.836838	.500000	.083684	.083684
r^{u_1}	p^{u_1}	r^{d_1}	p^{d_1}	N	Continuous	.836969	.500000	.079334	.079334
.1	.01	.1	.01	1	Dis ($N=2$)	.833699	1.000000	.082931	.082931
					Sim – Mean	.836784	.499886	.083743	.083684
					Sim – Stdev	.000370	.001060	.000234	.000322
					New	.773223	.331604	.154645	.077322
r^{u_1}	p^{u_1}	r^{d_1}	p^{d_1}	N	Continuous	.773544	.332261	.149101	.071747
.1	.02	.1	.01	1	Dis ($N=2$)	.769854	.923015	.153160	.076175
					Sim – Mean	.773256	.331781	.154620	.077311
					Sim – Stdev	.000379	.000828	.000327	.000301
					New	.852695	.481051	.094744	.056846
r^{u_1}	p^{u_1}	r^{d_1}	p^{d_1}	N	Continuous	.852739	.474392	.090412	.052513
.09	.01	.15	.01	1	Dis ($N=2$)	.849375	.962250	.094000	.056250
					Sim – Mean	.852616	.480659	.094830	.056839
					Sim – Stdev	.000280	.001108	.000243	.000158
					New	.850760	.814120	.050045	.106345
r^{u_1}	p^{u_1}	r^{d_1}	p^{d_1}	N	Continuous	.850652	.798747	.043016	.099309
.17	.01	.4	.05	1	Dis ($N=2$)	.845432	1.055948	.048889	.104837
					Sim – Mean	.850752	.814154	.050030	.106363
					Sim – Stdev	.000216	.000554	.000186	.000144

also very close to that from the new analytical solutions. However, the average inventory from "Dis ($N = 2$)"is not accurate. This is expected, because we arbitrarily enlarge N from 1 to 2 to use the model. Consequently, the average inventory is higher than it should be.

- In terms of p_s and p_b, the "Dis ($N = 2$)"approach is a good approximation. However, the "Continuous"approach underestimates both p_b and p_s in all cases.

Therefore, alternative 1 outperforms alternative 2 in terms of $P(N)$ and $\bar{n}$, but alternative 2 is better in terms of p_b and p_s. Since we want to improve the evaluation accuracy and to defeat the Batman effect, which is very important in optimization, the new analytical solutions are desired.

5.4.3 Numerical Evidence about the Improvement of Evaluation – Revisiting the Batman Effect

Up to this point, we have identified the two problems that lead to evaluation inaccuracy in the existing algorithm. To summarize, the first problem is that by inserting the perfectly reliable machines to eliminate buffer thresholds, we bring some additional time delay that reduces the production rate in the decomposition. This is resolved in Section 5.4.1. The second problem is that it is possible to have buffers of size 1 after inserting perfectly reliable machines but the existing discrete model for the two-machine one-buffer building block cannot deal with the case where $N = 1$. Therefore, two alternatives can be adopted but they result in evaluation inaccuracy with respect to different measures. This is addressed in Section 5.4.2. With both two modifications of the loop evaluation, we will be able to mitigate the Batman effect. In other words, the production rate curve as loop invariant I varies will be smoother. We provide six experiments below. Since we want to compare the new evaluation results with Werner's algorithm, we will only consider closed-loop systems. Numerical experiments involving single open-loop systems are provided in Section 5.5.

Experiment 1

We start with the case discussed in Section 5.2.3. It is a closed three-machine three-buffer loop with identical machines. Each machine has a single failure mode with failure probability of .01 in each time unit while it is up, and repair probability of .1 in each time unit while it is down. The size of each buffer is 10. We vary the loop invariant I from 4 to 26. The results are depicted in Figure 5-26. It can be seen that with those two modifications, the production rate curve is smoother and there is no Batman effect in this experiment.

Experiment 2

The second experiment is also a closed three-machine three-buffer loop but with different machines. Each machine has a single failure mode. Machine parameters are

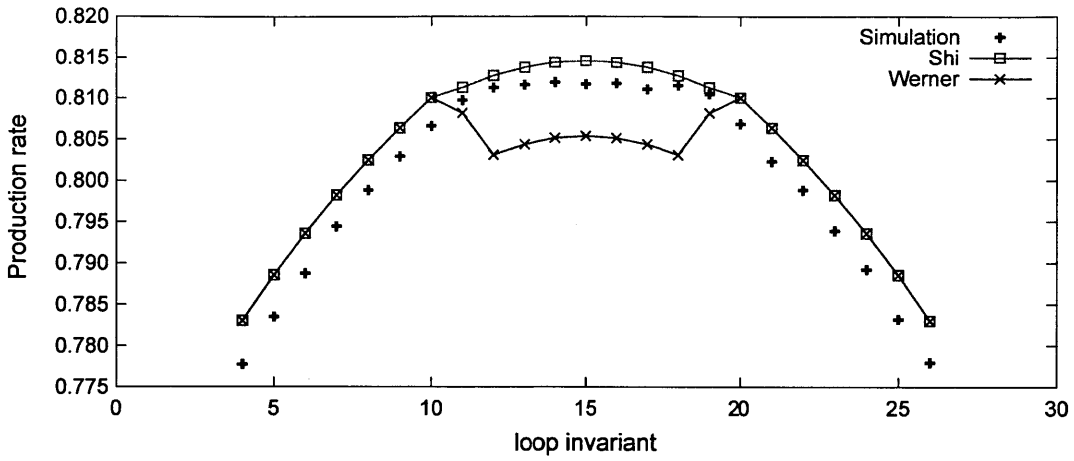


Figure 5-26: Numerical experiment 1 about the elimination of the Batman effect

$r_1 = .1$, $p_1 = .01$, $r_2 = .1$, $p_2 = .01$, $r_3 = .2$, and $p_3 = .01$. The size of each buffer is still 10. We vary the loop invariant I from 4 to 26. The results are depicted in Figure 5-27. The production rate curve is smoother and there is no Batman effect in this experiment.

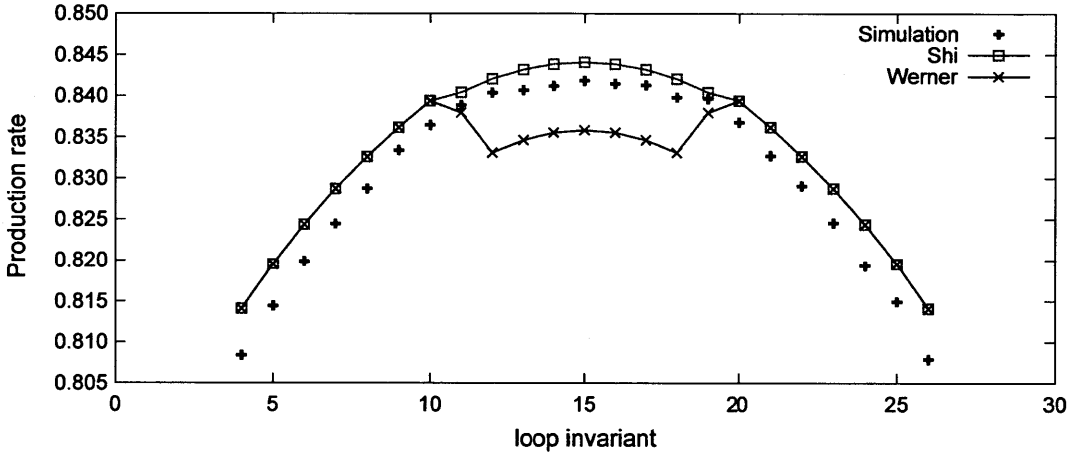


Figure 5-27: Numerical experiment 2 about the elimination of the Batman effect

Experiment 3

We consider another three-machine three-buffer loop with totally different machines and buffers in experiment 3. Each machine has a single failure mode. Machine parameters are $r_1 = .1$, $p_1 = .01$, $r_2 = .11$, $p_2 = .009$, $r_3 = .2$, and $p_3 = .013$. The

sizes of buffers are 12, 13, and 9, respectively. We vary the loop invariant I from 4 to 30. The results are illustrated in Figure 5-28. The production rate curve is smoother and there is no Batman effect.

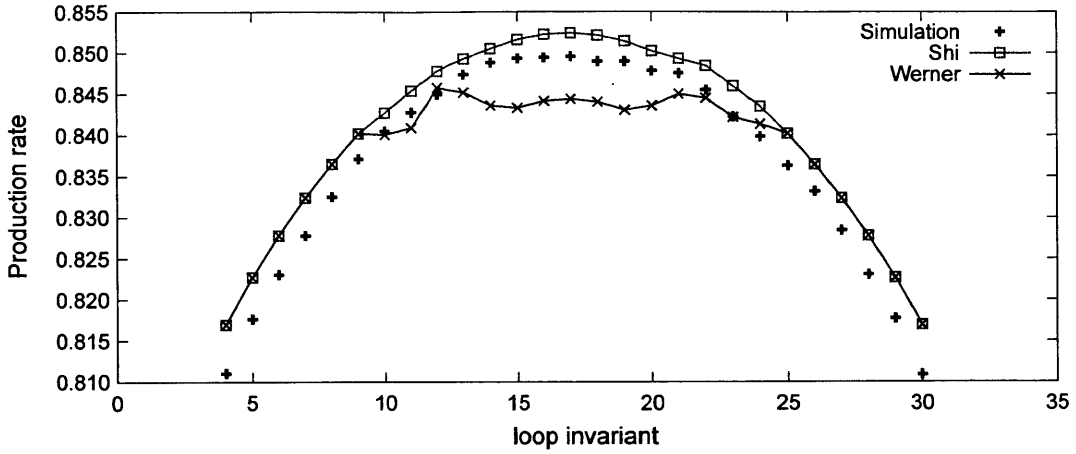


Figure 5-28: Numerical experiment 3 about the elimination of the Batman effect

Experiment 4

Now, we consider a four-machine four-buffer closed-loop with different machines and buffers. Each machine has a single failure mode. Machine parameters are $r_1 = .1$, $p_1 = .01$, $r_2 = .2$, $p_2 = .02$, $r_3 = .2$, $p_3 = .02$, $r_4 = .1$, and $p_4 = .01$. The sizes of buffers are 8, 12, 16, and 14, respectively. We vary the loop invariant I from 5 to 45. The results are shown in Figure 5-29. The production rate curve is smoother and there is no Batman effect.

Experiment 5

Experiment 5 considers another four-machine four-buffer loop with identical machines and buffers. Each machine has a single failure mode. Each machine has a failure probability of .01 in each time unit while it is up, and a repair probability of .1 in each time unit while it is down. The size of each buffer 10. We vary the loop invariant I from 5 to 35. The results are shown in Figure 5-30. The production rate curve is smoother with small bumps.

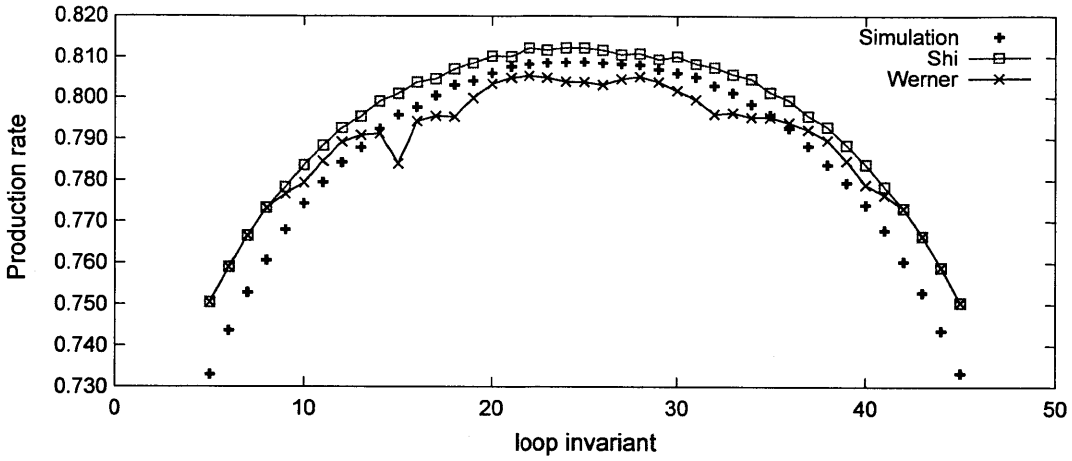


Figure 5-29: Numerical experiment 4 about the elimination of the Batman effect

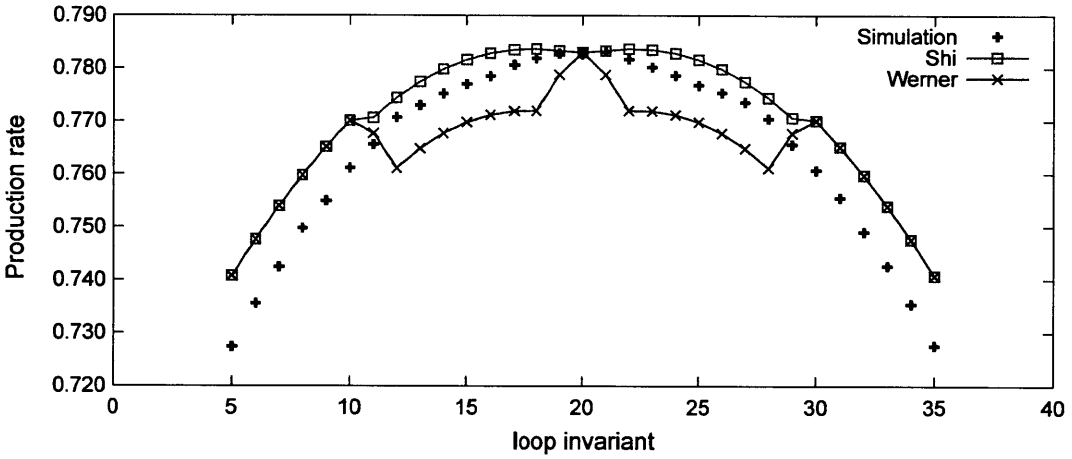


Figure 5-30: Numerical experiment 5 about the elimination of the Batman effect

Experiment 6

Finally, we consider a five-machine five-buffer loop with different machines and buffers. Each machine has a single failure mode. Machine parameters are $r_1 = .1$, $p_1 = .01$, $r_2 = .13$, $p_2 = .011$, $r_3 = .09$, $p_3 = .01$, $r_4 = .1$, $p_4 = .013$, $r_5 = .12$, and $p_5 = .012$. The sizes of buffers are 13, 10, 15, 12, and 15, respectively. We vary the loop invariant I from 5 to 60. The results are illustrated in Figure 5-31. The production rate curve is smoother as compared to Werner's algorithm. However, we have to point out that even with the two modifications, there are also small bumps in the curve. This is because the two modifications make the evaluation of two-machine one-buffer

building blocks more accurate. However, the decomposition approach that utilizes and evaluates a set of two-machine one-buffer building blocks is an approximation in evaluating the original loop system.

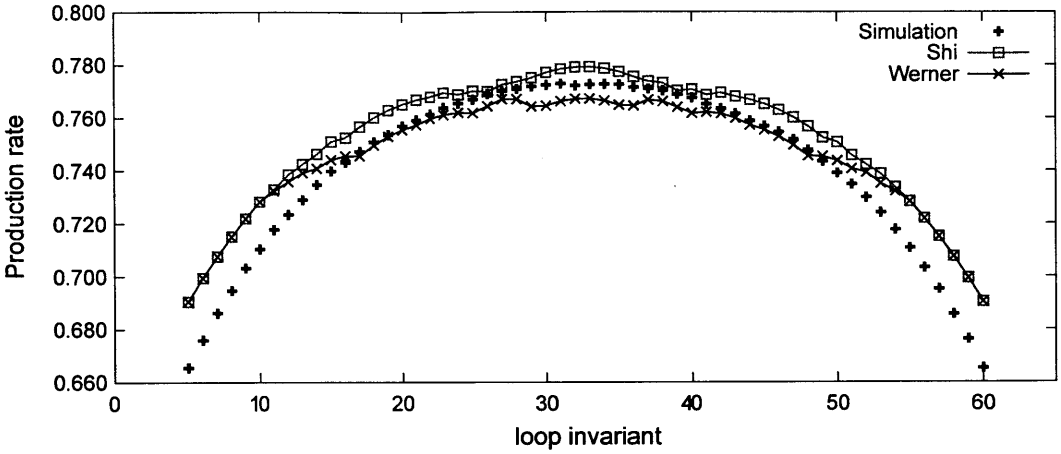


Figure 5-31: Numerical experiment 6 about the elimination of the Batman effect

5.5 Numerical Experiments of Single Open-Loop System Evaluation

In this chapter, we extend the evaluation algorithm of Werner (2001) and Gershwin and Werner (2007) for closed-loop systems to single open-loop systems. More importantly, we demonstrate two problems with the existing algorithms and resolve them. With these two modifications, loop evaluation results are more accurate. In this section, we study numerical experiments for single open-loop systems. In particular, we first compare the results with both simulation and the algorithm developed by Zhang (2006), whose objective is multiple loop systems. Since single open-loop systems belong to a subset of multiple loop systems, the algorithm of Zhang (2006) can be applied here for comparison directly. After that, 700 numerical experiments for open-loop systems up to 10 machines and 10 buffers are given to show the accuracy of modified evaluation algorithm.

Experiment 1

The first system under consideration is illustrated in Figure 5-32. System parameters are listed in Table 5.7. The loop invariant I is 37. The evaluation results are shown in Table 5.8, where $P(\mathbf{N}, I)$ is the production rate whose unit is parts per time unit, $\bar{n}_i$ is the average inventory level of Buffer B_i, L is the set that contains all buffers within the loop, and $\sum_{i \in L} \bar{n}_i$ is the total number of parts traveling in the loop. It should be equal to I, but there exists a small error due to the decomposition approach. In this example, both the modified algorithm with the two modifications and the algorithm of Zhang (2006) are accurate as compared with simulation. However, the modified evaluation algorithm is more accurate as the $P(\mathbf{N}, I)$, $\bar{n}_5$, and $\bar{n}_6$ from the algorithm are much closer to the simulation than those from the existing algorithm of Zhang (2006).

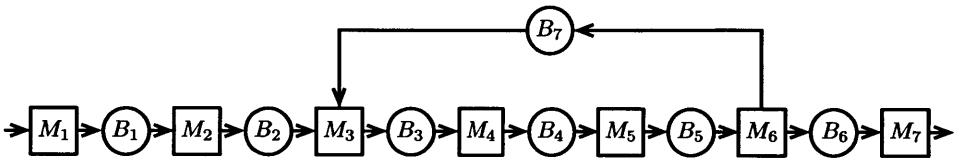


Figure 5-32: Single open-loop system, Experiment 1

Table 5.7: Parameters of the single open-loop system, Experiment 1

machine	M_1	M_2	M_3	M_4	M_5	M_6	M_7
r_i	.083	.021	.019	.096	.097	.082	.092
p_i	.076	.020	.005	.093	.093	.012	.073
e_i	.522	.512	.792	.508	.511	.872	.558
buffer	B_1	B_2	B_3	B_4	B_5	B_6	B_7
N_i	23	76	6	15	13	30	40

Experiment 2

Let us next consider the system described in Figure 5-33. System parameters are listed in Table 5.10. The loop invariant I is 42. The evaluation results are shown

Table 5.8: Results of single open-loop system, Experiment 1

	Simulation	Shi	Zhang
$P(\mathbf{N}, I)$	.364183	.367672	.358605
$\bar{n}_1$	13.543882	13.530605	14.477300
$\bar{n}_2$	45.967628	45.83004	48.548300
$\bar{n}_3$	4.492122	4.533134	4.856140
$\bar{n}_4$	6.254219	6.219535	6.253310
$\bar{n}_5$	1.300277	1.210711	0.870787
$\bar{n}_6$	6.465453	6.336823	5.705060
$\bar{n}_7$	24.953382	25.087090	25.024500
$\sum_{i \in L} \bar{n}_i$	37.000000	37.050469	37.004700

in table 5.10. Again, both the new algorithm with the two modifications and the algorithm of Zhang (2006) are accurate.

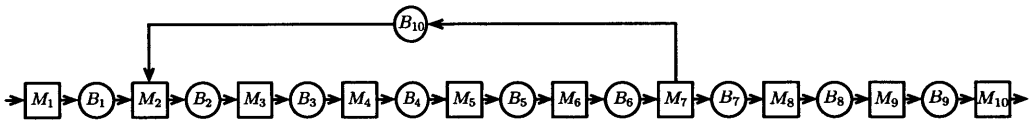


Figure 5-33: Single open-loop system, Experiment 2

Table 5.9: Parameters of the single open-loop system, Experiment 2

machine	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
r_i	.200	.100	.090	.100	.200	.110	.090	.070	.200	.150
p_i	.030	.010	.020	.010	.011	.010	.008	.006	.011	.009
e_i	.870	.909	.818	.909	.948	.917	.918	.921	.948	.943
buffer	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}
N_i	17	22	9	38	20	15	7	18	16	30

Table 5.10: Results of single open-loop system, Experiment 2

	Simulation	Shi	Zhang
$P(\mathbf{N}, I)$	.733686	.741017	.750005
$\bar{n}_1$	11.367146	11.271616	11.357500
$\bar{n}_2$	12.252673	12.739613	13.028900
$\bar{n}_3$	2.535181	2.099647	2.097710
$\bar{n}_4$	3.644253	3.163109	3.246130
$\bar{n}_5$	6.432062	6.155862	5.949010
$\bar{n}_6$	5.570066	4.921985	5.053040
$\bar{n}_7$	2.013722	1.840309	1.475680
$\bar{n}_8$	2.796034	2.784287	2.402450
$\bar{n}_9$	2.526037	2.447165	2.013970
$\bar{n}_{10}$	11.565766	12.968350	13.453600
$\sum_{i \in L} \bar{n}_i$	42.000000	42.048567	42.828400

Experiment 3

Zhang's algorithm provides accurate evaluation results for multiple loop systems (Zhang 2006). But, since the two problems discussed in this chapter appear in his model as well, we observe the Batman effect. The Batman effect shows a discontinuity of the production rate as a function of loop invariant, which is undesirable to the loop optimization work that we will discuss in Chapter 6.

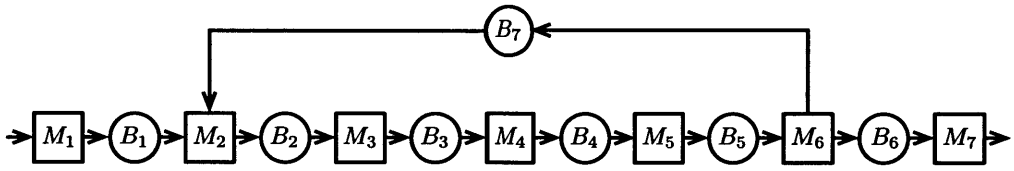


Figure 5-34: Single open-loop system, Experiment 3

Here an example is given. Consider the system shown in Figure 5-34. Assume that machines are identical. Each machine has one failure model. The failure probability is .01 and the repair probability is .1. We vary the loop invariant from 21 to 29. The results are shown in Figure 5-35. It can be seen from the graph that both the new algorithm and the algorithm of Zhang (2006) are very accurate as compared with

simulation (those discrete dots). However, results from Zhang (2006) show obvious discontinuities.

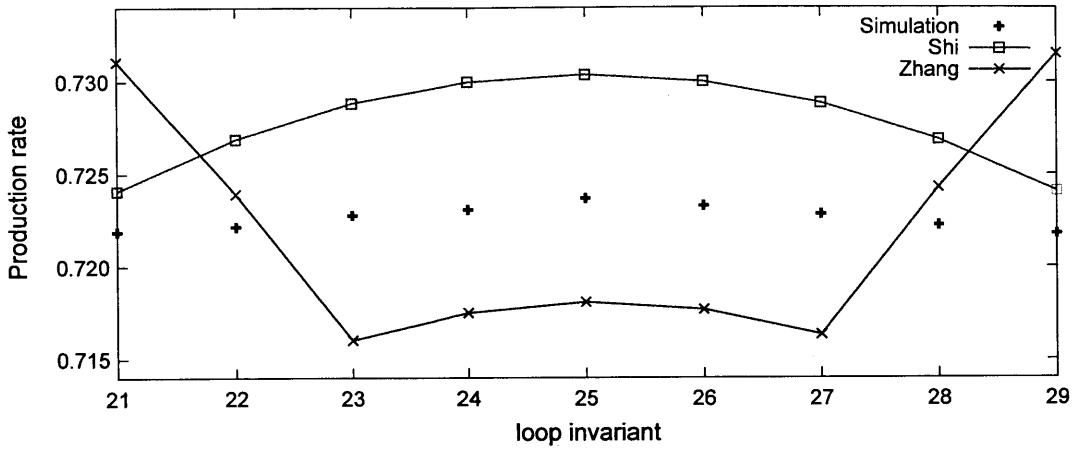


Figure 5-35: Batman phenomenon in a single open-loop system

More numerical results

Finally, we provide numerical results for 700 other single open-loop systems. The sizes of these systems vary from three-machine three-buffer loops up to 10-machine 10-buffer loops. Their machine parameters, buffer spaces, and loop structures are generated randomly according to a method that is similar to the one described in Gershwin (2011). For each experiment, the evaluation results from the improved evaluation algorithm are compared against simulation results. Moreover, we report both production rate error (P_{err}) and the maximum average inventory error ($\bar{n}_{\text{err}}$), which are defined as:

$$P_{\text{err}} = \left| \frac{P^{\text{sim}} - P^{\text{num}})}{P^{\text{sim}}} \right| \times 100\%, \tag{5.99}$$

and

$$\bar{n}_{\text{err}} = \max_{i=1,\cdots,k} \left\{ \left| \frac{\bar{n}_i^{\text{sim}} - \bar{n}_i^{\text{num}}}{\bar{n}_i^{\text{sim}}} \right| \times 100\% \right\} \tag{5.100}$$

where P^{sim} and P^{num} are the production rates from the simulation and the analytical solution, respectively; while $\bar{n}_i^{\text{sim}}$ and $\bar{n}_i^{\text{num}}$ are the average inventory levels of

Buffer B_i from the simulation and the analytical solution, respectively.

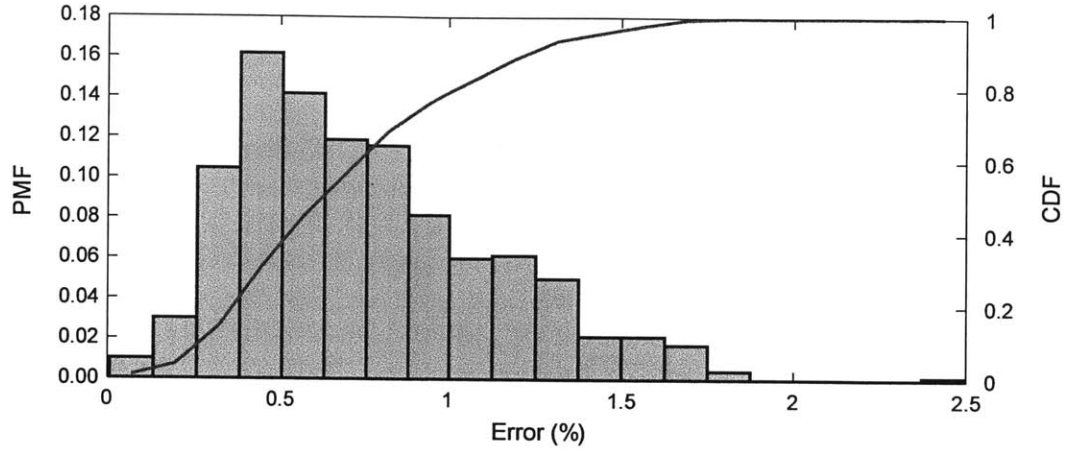


Figure 5-36: Production rate error of 700 experiments

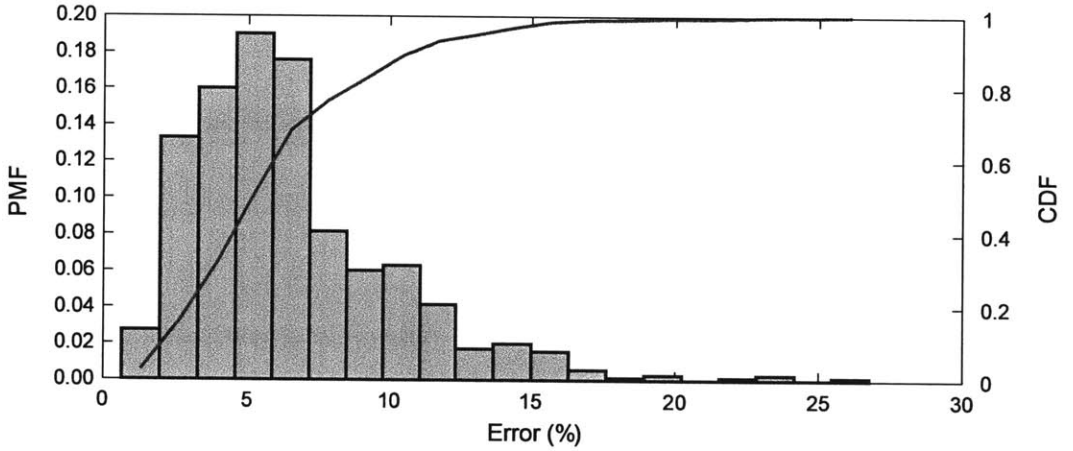


Figure 5-37: Average inventory error of 700 experiments

Figure 5-36 shows the distribution of production rate errors of the 700 experiments. In these experiments 76% of these cases have a production rate error that is less than 1%, while 699 out of 700 cases have a production rate error that is less than 2%. The average production rate error of these 700 experiments is 0.75%. Similarly, Figure 5-37 shows the distribution of maximum average inventory error of the 700 experiments. In these experiments 83.86% of these cases have an average inventory level error that is less than 10%, and 96.71% of these cases have an average inventory level error that is less than 15%. The average average inventory error of these 700 experiments

is 6.54%. Werner (2001) studies a large number of experiments for three-machine, six-machine, and 10-machine closed-loop systems, and reports the errors in terms of the production rate and average inventory level as well. The errors shown here are comparable to the errors in his cases. In particular, the magnitude of production rate errors in the 700 experiments considered here are smaller than those cases in Werner (2001).

These results demonstrate the evaluation accuracy of the improved algorithm for single loop systems. More importantly, the modified algorithm mitigates the undesirable Batman effects and makes the production rate curve smoother. With the modified evaluation algorithm, we extend the profit maximization algorithm developed for transfer lines in Chapter 4 to single closed-loop systems in Chapter 6.

Chapter 6

Profit Maximization for Single Closed-Loop Production Systems

6.1 Scope of the Problem

We indicate in Chapter 5 that single closed-loop systems are common in manufacturing, where the total number of parts is controlled. Given their importance, in this chapter, we study the profit maximization problem subject to a production rate constraint for single closed-loop systems by extending the algorithm proposed in Chapter 4 for transfer lines to such systems.

Recall that in Chapter 4, in order to solve the profit maximization problem subject to a production rate constraint for transfer lines (i.e., Problem (4.3)), we formulate a corresponding unconstrained problem (4.4) by dropping off the production rate constraint. The constrained problem is solved by conducting a one-dimension search in the revenue coefficient A (which is replaced by A' in the search) in which for each A' the unconstrained problem is solved with a gradient method (see Section 4.2). Solving the unconstrained problem (4.4) with the gradient method requires that the profit space being searched has a single maximum. This point has been confirmed by Schor (1995), Gershwin and Schor (2000), Shi and Gershwin (2009a), (2009b), as well as the numerical experiments provided in Chapter 3 for transfer lines. However, we observe that this might not necessarily be true for a single closed-loop system. We

provide an example.

Table 6.1: Parameters of the five-machine five-buffer closed-loop example

Machine	M_1	M_2	M_3	M_4	M_5
r_i	.1	.1	.1	.1	.02
p_i	.01	.01	.01	.01	.01
$r_i/(r_i+p_i)$	.909	.909	.909	.909	.667

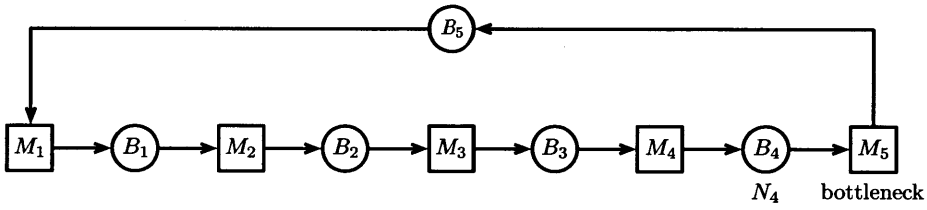


Figure 6-1: A five-machine five-buffer closed-loop system

Consider a five-machine five-buffer single closed-loop system where all machines except M_5 are identical, and M_5 is the bottleneck of the system. Machine parameters are listed in Table 6.1. The loop is illustrated in Figure 6-1. Let $N_1 = N_2 = N_3 = N_5 = 100$ and $I = 400$. We vary N_4 from 4 to 400 and evaluate the loop system for each N_4. Both the analytical approach described in Chapter 5 and simulation are used for evaluation. In particular, for each set of system parameters, the length of the simulation is 5,100,000 time steps with the first 100,000 time steps being the warm up period, and we run the simulation 20 times.

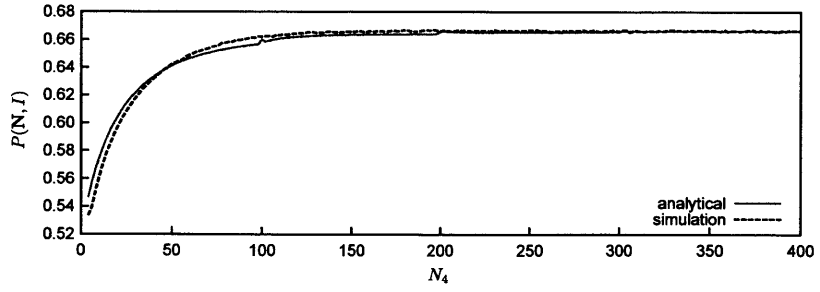


Figure 6-2: The production rate of the five-machine five-buffer closed-loop system

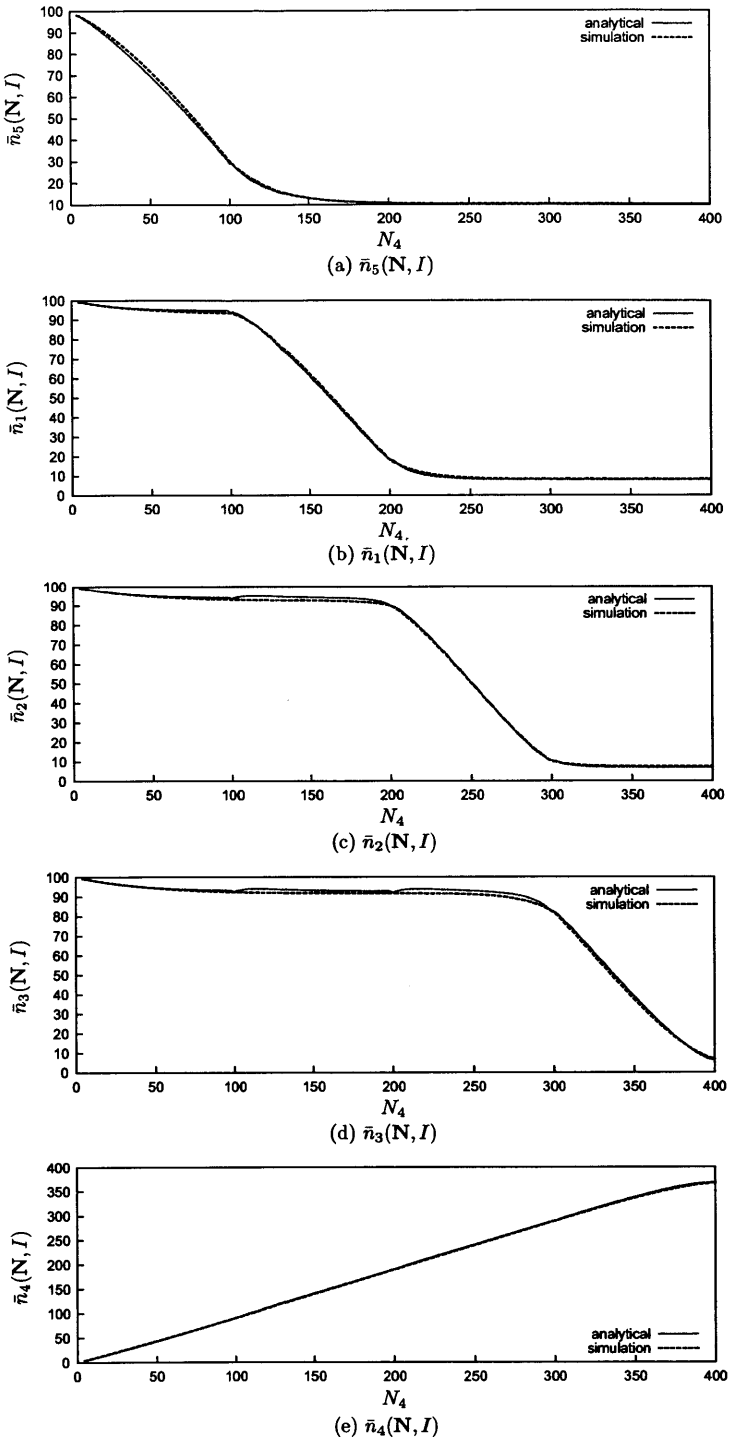


Figure 6-3: The average inventory levels of the five buffers of the five-machine five-buffer closed-loop system

The production rate and the average inventory levels of the five buffers as functions of N_4 are illustrated in Figures 6-2 and 6-3, respectively. (Note that the average level graphs shown in Figure 6-3 are in the order of $\bar{n}_5$, $\bar{n}_1$, $\bar{n}_2$, $\bar{n}_3$, and $\bar{n}_4$.) It can be seen that the results from the analytical approach and the simulation are very close. It appears that the production rate is a concave function of N_4, and we will discuss more about the qualitative property of the production rate of a single closed-loop system in Section 6.2.

The average inventory levels of the five buffers of this system are of interest. We observe from Figure 6-3 that, since M_5 is the bottleneck machine and its isolated production rate is much smaller than all the other machines, parts tend to accumulate in its upstream buffer B_4. When N_4 is small, the loop invariant ($I = 400$) is large enough to make all Buffers B_3, B_2, B_1, and B_5 (which is the furthest buffer for M_5) to be almost full in steady state. As N_4 gets bigger, there is less blockage in the system and more parts accumulated in upstream buffers B_3, B_2, B_1, and B_5 tend to move forward. In other words, as N_4 increases, B_4 will be filled with some parts from B_3. This leaves extra space in B_3, but this space will be filled by parts from B_2. Similarly, extra space available in Buffer B_2 will be filled by parts from B_1. Finally, space available in Buffer B_1 will be compensated by parts from Buffer B_5. As a results of part movement due to the increase of N_4, Buffers B_3, B_2, and B_1 still tend to be full, while $\bar{n}_5$ becomes smaller and smaller as parts are moving forward to B_4 eventually through B_1, B_2, and B_3 (see Figure 6-1), and this trend will not change until N_4 reaches 100. Therefore, when N_4 is between 4 and 100, $\bar{n}_5$ decreases significantly while $\bar{n}_4$ increases significantly.

Once N_4 passes 100, the average inventory level of B_1 starts to decrease significantly. This is because more and more parts in B_1 tends to move downstream as a result of the increase of N_4. However, few parts are added into B_1 from B_5 because the latter already has a very low inventory level. The results in Figure 6-3 confirms this as $\bar{n}_1$ starts to decrease significantly once N_4 is greater than 100 when $\bar{n}_5$ already decreases to a relatively low level. The rest of the results shown in Figure 6-3 can be explained in a similar manner. $\bar{n}_2$ starts to decrease dramatically once N_4 is greater

than 200 when both $\bar{n}_5$ and $\bar{n}_1$ already decrease to low levels. Eventually, $\bar{n}_3$ will start to decrease drastically once N_4 is greater than 300 when all other buffers except B_4 get almost empty.

Given the interesting behaviors of average inventory levels due to the specific parameters of the loop system, we consider the following profit function:

$$J(\mathbf{N}, I) = 1000P(\mathbf{N}, I) - 3\bar{n}_1 - \bar{n}_2 - N_4$$

and the profit of the loop is plotted in Figure 6-4.

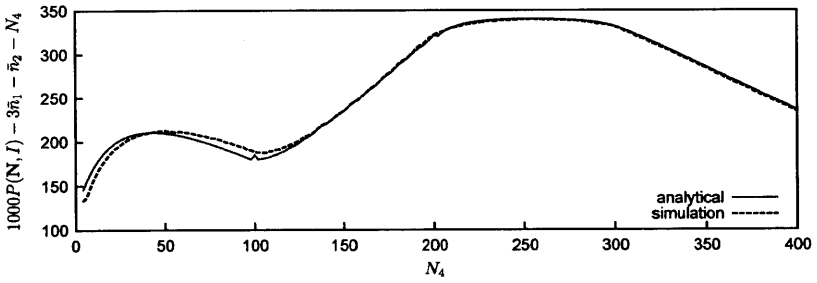


Figure 6-4: The profit of the five-machine five-buffer closed-loop system

It can be seen from Figure 6-4 that the profit curve has two maxima[1]. First, there is a local maximum of the profit when N_4 is around 50. Then, there is a global maximum of the profit when N_4 is around 250. The local maximum and the global maximum are caused by different factors.

As N_4 increases from 0 to 50, the production rate of the loop increases rapidly (see Figure 6-2) and the revenue increase associated with it exceeds the cost of an increasing N_4[2]. Once N_4 keeps increasing after 50, the rate of increase of $P(\mathbf{N}, I)$ decreases due to its concavity. Therefore, the revenue increase associated with $P(\mathbf{N}, I)$ cannot compensate for the cost increase of N_4. As a consequence, the profit decreases and this leads to the local profit maximum around $N_4 = 50$.

[1]The small bump on the profit curve when $I = 100$ is not a local optimum. But rather, it is due to the remaining batman effect after the two modifications discussed in Chapter 5 have been made to improve the loop evaluation.

[2]Note that as N_4 increases, $\bar{n}_1$ and $\bar{n}_2$ also decrease a little, and therefore they also bring minor contributions to the profit. However, the fast increases of the production rate and the revenue are the key drivers.

On the other hand, once N_4 pasts 100 and keeps increasing, the revenue increment associated with $P(\mathbf{N}, I)$ is very small. The profit increase is due to the fact that $\bar{n}_1$ decreases rapidly. The inventory holding cost coefficient of B_1 is 3, so, as N_4 increases, parts move from more expensive buffers to cheaper buffers. The savings on the inventory holding cost exceeds the cost of increasing N_4, and therefore the profit increases again. Once N_4 passes 300, the first three terms of J (i.e., $1000P(\mathbf{N}, I)$, $-3\bar{n}_1$, and $-\bar{n}_2$) become stable. Therefore, the profit decreases approximately linear with N_4. The behavior of the profit described above results in the global maximum.

Due to the existence of two maxima, it is possible that a gradient search method will stop at the local maximum rather than the global one. This will prevent the algorithm from finding the correct global maximum of the profit. Given this fact, we impose a certain assumption to our objective function, and this also defines the scope of our loop optimization problem.

The assumption that will be considered in closed-loop optimization is that the average inventory cost coefficients of all buffers are the same. In other words, $c_1 = c_2 = \cdots = c_k = c$. This assumption indicates that the inventory holding costs at different buffers are the same. Although this assumption narrows the scope of the study on loop optimization, we consider it a fair assumption because it is common in manufacturing systems where all machines and buffers are operating in the same environment, or for products that do not deteriorate or lose values over the time scale of the lead time. Examples where this assumption holds include the fabrication of metal parts (Bard and Feo 1989 and Wang and Bourne 1997), the assembly of subsystems like the engine and the transmission of a car, and the fabrication of complex printed circuit boards like computer mother boards (Carano and Fjelstad 2003)[3]. Also, see Ip et al. (2007) for an example of a CONWIP-based control of a lamp assembly production system. The assembly system contains five single loops in which three of them have equal inventory holding costs. However, it is important to point out that the assumption of constant inventory holding cost does not hold in some cases. For example, the inventory holding cost in a clean room environment can

[3]The fabrication of the board itself rather than the assembly of components on the boards.

be much more expensive than elsewhere in the system.

Provided the assumption on average inventory cost coefficients, the objective function of the loop profit maximization is

$$\begin{aligned} J(N_1, \cdots, N_k, I) &= AP(N_1, \cdots, N_k, I) - \sum_{i=1}^{k} b_i N_i - \sum_{i=1}^{k} c\bar{n}_i \\ &= AP(N_1, \cdots, N_k, I) - \sum_{i=1}^{k} b_i N_i - cI. \end{aligned} \tag{6.1}$$

It can be seen that in the objective function, the buffer space cost and the average inventory holding cost are linear functions of the decision variables $\mathbf{N}$ and I. Therefore, whether or not (6.1) has a single global maximum that can be found by the gradient method depends on the property of the production rate of the closed-loop system $P(\mathbf{N}, I)$. We will study this in Section 6.2 and indicate the concavity of $P(\mathbf{N}, I)$ as a good and reasonable assumption. With this concavity assumption, the profit function will be a concave function of $\mathbf{N}$ and I, and this will guarantee the single maximum.

The rest of this chapter is organized as follows. We first discuss some qualitative properties of the production rate of loop systems (which is a function of both buffer sizes $\mathbf{N}$ and loop invariant I) in Section 6.2. The optimization algorithm is stated in Section 6.3, followed by numerical experiments in Section 6.4. We summarize this chapter in Section 6.5.

6.2 Qualitative Property of $P(\mathbf{N}, I)$

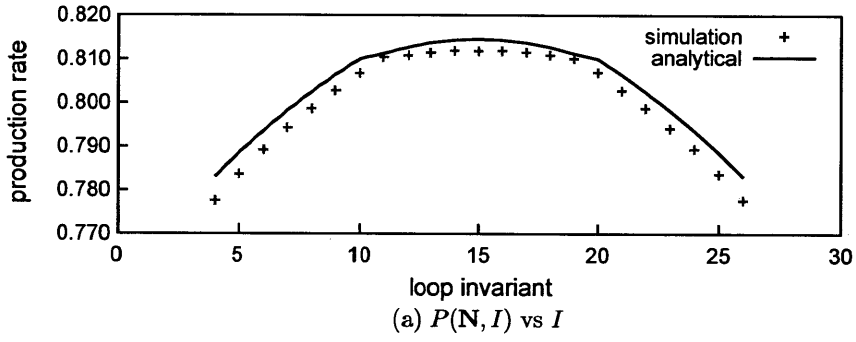
6.2.1 Numerical Observation

We discuss the qualitative property of $P(\mathbf{N}, I)$ for single closed-loop systems with numerical experiments in this section. In particular, we consider a symmetric three-machine loop and an asymmetric three-machine loop.

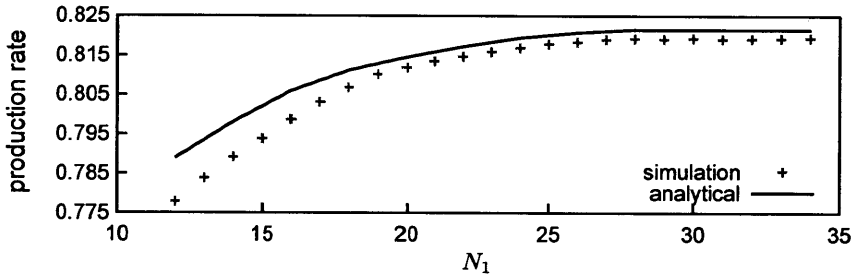
In the symmetric loop, machines are identical with $r_i = .1$ and $p_i = .01$. We

study how $P(\mathbf{N}, I)$ changes with I (given $N_1 = N_2 = N_3 = 10$) as well as with N_1 (given $I = 28$ and $N_2 = N_3 = 10$). The results are illustrated in Figure 6-5, where both analytical results and simulation results are provided. It appears from Figure 6-5(a) that $P(\mathbf{N}, I)$ is a concave function of I. It addition, we observe the symmetric property of $P(\mathbf{N}, I)$, which is conjectured by some research listed in Section 6.2.2, such that

$$P(\mathbf{N}, I) = P\left(\mathbf{N}, \sum_{i=1}^{3} N_i - I\right). \tag{6.2}$$



(a) $P(\mathbf{N}, I)$ vs I



(b) $P(\mathbf{N}, I)$ vs N_1

Figure 6-5: $P(\mathbf{N}, I)$ vs N and I, symmetric loop

On the other hand, from Figure 6-5(b), it appears that $P(\mathbf{N}, I)$ is a concave function of N_i. In addition, we notice that the production rate stops increasing but becomes constant when $N_1 \geq 28$. This is because the loop invariant I is 28 in this case. Therefore, once $N_1 \geq I$, in extra space in Buffer B_1 has no impact on other machines in terms of starvation or blocking, and therefore the production rate of the system will remain unchanged.

In the asymmetric loop example, machines are different with their parameters listed in Table 6.2. Again, we study how $P(\mathbf{N}, I)$ changes with I as well as with N_1,

N_2, and N_3. In particular, $N_1 = 12$, $N_2 = 16$, and $N_3 = 10$. we first vary I from 4 to 34. Then by choosing I to be 26, we vary N_1, N_2, and N_3 once a time. The results are illustrated in Figure 6-6. It appears from Figure 6-6(a) that $P(\mathbf{N}, I)$ is a concave function of I. As before, we observe the symmetric property of $P(\mathbf{N}, I)$. From Figure 6-6(b), it appears that $P(\mathbf{N}, I)$ is a concave function of N_i as well. The production rate stops increasing but becomes constant once $N_i \geq I$.

Table 6.2: Parameters of the three-machine asymmetric closed-loop

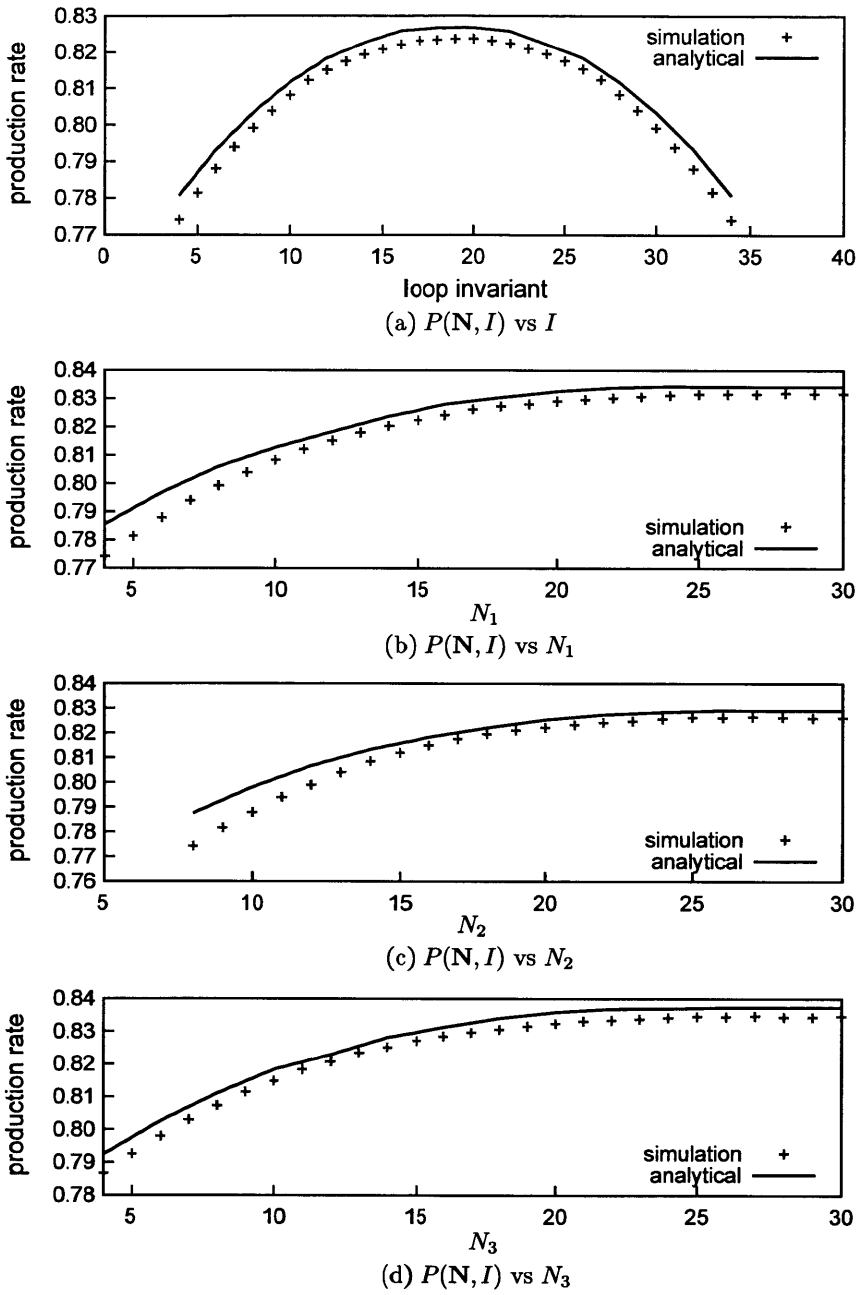
machine	M_1	M_2	M_3
r_i	.10	.20	.10
p_i	.01	.02	.011
$r_i/(r_i + p_i)$	.909	.909	.901

The two examples for a symmetric loop and an asymmetric loop suggests that $P(\mathbf{N}, I)$ appears to be a concave function of both the buffer sizes and the loop invariant.

6.2.2 Literature Review

Some research has been dedicated to study some qualitative properties (e.g., the symmetry, the monotonicity, and the concavity) of continuous time closed queueing systems with finite buffers and blocking, where machines are modeled as reliable machines whose service times are assumed to follow either exponential distributions or phase-type distributions (Neuts 1981). Although the research is for continuous systems, we believe that the discrete closed-loop we study in the thesis exhibit the similar property. Therefore, the research in similar systems combined with the numerical observation provided in Section 6.2.1 indicate that the concavity property of $P(\mathbf{N}, I)$ is a reasonable assumption.

Shanthikumar and Yao (1989a) study cyclic queueing networks with finite buffer capacity and blocking before service (Perros 1990, Onvural and Perros 1986, Gün and Makowski 1989). In addition, the service process at each stage is Markovian with the



(a) $P(\mathbf{N}, I)$ vs I

(b) $P(\mathbf{N}, I)$ vs N_1

(c) $P(\mathbf{N}, I)$ vs N_2

(d) $P(\mathbf{N}, I)$ vs N_3

Figure 6-6: $P(\mathbf{N}, I)$ vs N and I, asymmetric loop

service rate μ_i at stage i depending only on the number of jobs of that stage and if the server is blocked. They show that the production rate of such queueing networks is a monotonic function of buffer capacity and a concave function of population size.

In addition, Shanthikumar and Yao (1988) show that the throughput of a single-

class closed queueing network of Jackson type (Jackson 1963), as a function of the job population, is nondecreasing concave if the service rate at each node, as a function of the local queue length, has the same property.

Onvural and Perros (1987) conjecture that the production rate of a closed tandem queueing network with finite buffers and exponential servers is symmetrical with respect to the population of the network. Onvural and Perros (1989) present an approximation algorithm for estimating the throughput of closed networks as a function of the number of jobs (i.e., loop population) in it and indicate that in closed queueing networks with blocking, the throughput increases as a function of the number of jobs up to a point after which it decreases. The point at which the throughput is maximized with respect to the loop population is conjectured to be equal to $\sum_{i=1}^{k} N_i$. This is confirmed by Onvural (1990). Onvural and Perros (1989) also conjecture that the throughput curve with respect to the loop population is symmetric around the point at which it reaches its maximum.

Dallery and Towsley (1991) prove the symmetry of the production rate conjectured by Onvural and Perros (1987) for closed tandem queueing networks with finite buffers. In particular, they assume blocking before service and the service times of all servers are i.i.d. random variables with phase-type distribution[4]. In other words, the machines considered in their model are reliable with i.i.d. service times, and the system is a continuous model. Dallery and Gershwin (1992) explain that an unreliable machine with exponential processing time μ, uptime r, and downtime p can be modeled as a reliable machine with phase-type distribution. Therefore, we expect that the symmetry property holds for the production rate of a closed-loop system with identical unreliable machines and finite buffers in the exponential production line model.

The numerical experiments in concert with the literature review indicate that the concavity of $P(\mathbf{N}, I)$ is a reasonable assumption. With this assumption, we see that the profit of a closed-loop system, Equation (6.1), where all c_i's are the same is a concave function. Therefore there is a single maximum and it can be found by the

[4]Their results also apply to exponential servers.

gradient method. The proposed profit maximization algorithm for such closed-loop systems is described in Section 6.3.

6.3 Profit Maximization Algorithm for Loops

In this section, we present the profit maximization algorithm for single closed-loop systems. Similar to what we do for transfer lines in Chapter 4, we try to maximize the profit of single closed-loop systems subject to a production rate constraint. Note that in closed-loop optimization, the decision variables are the buffer sizes $N_i, i = 1, \cdots, k$ and the loop invariant I. The optimization problem is formulated as

$$\begin{aligned}
\max_{\mathbf{N},I} \quad & J(N_1, \cdots, N_k, I) &=\ & AP(N_1, \cdots, N_k, I) - \sum_{i=1}^{k} b_i N_i - cI \\
\text{subject to} \quad & P(N_1, \cdots, N_k, I) &\geq\ & \hat{P}, \\
& N_i &\geq\ & N_{\min}, \forall i = 1, \cdots, k, \\
& \sum_{i=1}^{k} N_i &\geq\ & I, \\
& I &\geq\ & 0
\end{aligned} \tag{6.3}$$

where the constraint $\sum_{i=1}^{k} N_i \geq I$ says that the total number of parts that are allowed in the system at any time should not exceed the total size of all buffers and the constraint $I \geq 0$ says that the number of parts in the system should be nonnegative.

According to the optimization technique developed in Chapter 4 for transfer lines, we have the following similar assertion for single closed-loop systems: the constrained problem

$$\begin{aligned}
\max_{\mathbf{N},I} \quad & J(N_1,\cdots,N_k,I) = A'P(N_1,\cdots,N_k,I) - \sum_{i=1}^{k} b_i N_i - cI \\
\text{subject to} \quad & P(N_1,\cdots,N_k,I) \geq \hat{P}, \\
& N_i \geq N_{\min}, \forall i = 1,\cdots,k, \\
& \sum_{i=1}^{k} N_i \geq I, \\
& I \geq 0
\end{aligned} \tag{6.4}$$

has the same solution for all A' in which the solution of the corresponding unconstrained problem

$$\begin{aligned}
\max_{\mathbf{N},I} \quad & J(N_1,\cdots,N_k,I) = A'P(N_1,\cdots,N_k,I) - \sum_{i=1}^{k} b_i N_i - cI \\
\text{subject to} \quad & N_i \geq N_{\min}, \forall i = 1,\cdots,k, \\
& \sum_{i=1}^{k} N_i \geq I, \\
& I \geq 0
\end{aligned} \tag{6.5}$$

has $P(N_1^u,\cdots,N_k^u,I^u) < \hat{P}$, where $(N_1^u,\cdots,N_k^u,I^u)$, or denoted by $(\mathbf{N}^u,I^u)$, is the solution of the unconstrained problem (6.5). This is because the solution of problem (6.4) will satisfy $P(N_1^\star,\cdots,N_k^\star,I^\star) = \hat{P}$ so the objective function is equivalent to $A'\hat{P} - \sum_{i=1}^{k} b_i N_i - cI$. Since the first term is independent of all of the N_i and I, it has no effect on the solution of the problem.

Similar to what we do in Section 4.2 of Chapter 4 for transfer lines, we apply the

KKT conditions again to prove the assertion for single closed-loop systems. We first convert the constrained problem (6.3) into the minimization form:

$$\begin{aligned}
\min_{\mathbf{N},I} \quad & -J(N_1,\cdots,N_k,I) &=& -AP(N_1,\cdots,N_k,I)+\sum_{i=1}^{k} b_i N_i + cI \\
\text{subject to} \quad & \hat{P}-P(N_1,\cdots,N_k,I) &\leq& 0, \\
& N_{\min}-N_i &\leq& 0, \forall i=1,\cdots,k, \\
& I-\sum_{i=1}^{k} N_i &\leq& 0, \\
& -I &\leq& 0.
\end{aligned} \tag{6.6}$$

We realize that for loop systems, even with the deterministic model where parts are discrete and buffer sizes are integers, we can still treat N_i, as well as I, as continuous variables. This is because in evaluating loop systems, the underlying evaluation of two-machine line building blocks do not require buffer sizes to be integer. Similar, the decomposition approach does not have that requirement either. A loop system differs from a transfer line in the sense that the imposed loop bring impact to the blocking and starvation of buffers within the loop. However, from the system evaluation standpoint, as long as we identify the failure modes of both the upstream and downstream pseudo-machines of all buffers, the loop evaluation procedure is the same as the transfer line evaluation procedure (see Chapter 5, Gershwin and Werner 2007, and Zhang 2006 for loop evaluation). Therefore, the buffer sizes as well as the loop invariant can be considered as continuous variables. As a result, $P(\mathbf{N},I)$ and $J(\mathbf{N},I)$ can be considered as continuously differentiable functions as well.

Let us consider the KKT conditions. We first point out a necessary condition that guarantees the existence of Lagrange multipliers. The appropriate one for the problem is again the Slater constraint qualification for convex inequalities (and we

first apply it in Chapter 4 for transfer lines). Let us now consider the constrained problem (6.6) for loops. In this problem, there are no equality constraints but there are $k+3$ inequality constraints:

$$\begin{aligned} g_0(\mathbf{N}, I) &= \hat{P} - P(N_1, \cdots, N_k, I) \leq 0, \\ g_i(\mathbf{N}, I) &= N_{\min} - N_i \leq 0, \forall i = 1, \cdots, k, \\ g_{k+1}(\mathbf{N}, I) &= I - \sum_{i=1}^{k} N_i \leq 0, \\ g_{k+2}(\mathbf{N}, I) &= -I \leq 0. \end{aligned} \tag{6.7}$$

Due to the concavity assumption of $P(\mathbf{N}, I)$, $g_0(\mathbf{N}, I)$ is a convex function. All other $g_i(\mathbf{N}, I)$ are linear and therefore they are also convex. In addition, it is not hard to find a feasible vector to make the problem satisfy the Slater constraint qualification. Since the required production rate, $\hat{P}$, has to be feasible for the line, there exists sufficiently large $\hat{\mathbf{N}}$ and $\hat{I}$ such that $P(\hat{N}_1, \cdots, \hat{N}_k, \hat{I}) > \hat{P}$ and $\hat{I} < \sum_{i=1}^{k} \hat{N}_i$. Thus, $g_0(\hat{N}_1, \cdots, \hat{N}_k, \hat{I}) < 0$ and $g_{k+1}(\hat{N}_1, \cdots, \hat{N}_k, \hat{I}) < 0$. In addition, $g_i(\hat{N}_1, \cdots, \hat{N}_k, \hat{I}) < 0, \forall i = 1, \cdots, k$ because $N_{\min} - \hat{N}_i < 0, \forall i = 1, \cdots, k$, and $g_{k+2}(\hat{N}_1, \cdots, \hat{N}_k, \hat{I}) < 0$ because $\hat{I} > 0$. Hence, the constrained problem (6.6) satisfies the Slater constraint qualification, and there exist unique Lagrange multipliers $\mu_i^\star, i = 0, \cdots, k+2$ for the problem to satisfy the KKT conditions:

$$\begin{aligned} -\nabla J(\mathbf{N}^\star, I^\star) + \mu_0^\star \nabla \left(\hat{P} - P(\mathbf{N}^\star, I^\star) \right) + \sum_{i=1}^{k} \mu_i^\star \nabla (N_{\min} - N_i^\star) \\ + \mu_{k+1}^\star \nabla \left(I^\star - \sum_{i=1}^{k} N^\star \right) + \mu_{k+2}^\star \nabla (-I^\star) = 0 \end{aligned} \tag{6.8}$$

or

$$-\begin{pmatrix} \dfrac{\partial J(\mathbf{N}^\star, I^\star)}{\partial N_1} \\ \vdots \\ \dfrac{\partial J(\mathbf{N}^\star, I^\star)}{\partial N_k} \\ \dfrac{\partial J(\mathbf{N}^\star, I^\star)}{\partial I} \end{pmatrix} - \mu_0^\star \begin{pmatrix} \dfrac{\partial P(\mathbf{N}^\star, I^\star)}{\partial N_1} \\ \vdots \\ \dfrac{\partial P(\mathbf{N}^\star, I^\star)}{\partial N_k} \\ \dfrac{\partial P(\mathbf{N}^\star, I^\star)}{\partial I} \end{pmatrix} - \mu_1^\star \begin{pmatrix} 1 \\ \vdots \\ 0 \\ 0 \end{pmatrix} - \cdots - \mu_k^\star \begin{pmatrix} 0 \\ \vdots \\ 1 \\ 0 \end{pmatrix} \tag{6.9}$$

$$+\mu_{k+1}^\star \begin{pmatrix} -1 \\ \vdots \\ -1 \\ 1 \end{pmatrix} - \mu_{k+2}^\star \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix},$$

and

$$\mu_i^\star \geq 0, \forall i = 0, \cdots, k+2, \tag{6.10}$$

$$\mu_0^\star \left(\hat{P} - P(\mathbf{N}^\star, I^\star) \right) = 0, \tag{6.11}$$

$$\mu_i^\star (N_{\min} - N_i^\star) = 0, \forall i = 1, \cdots, k, \tag{6.12}$$

$$\mu_{k+1}^\star \left(I^\star - \sum_{i=1}^{k} N_i^\star \right) = 0, \tag{6.13}$$

$$\mu_{k+2}^\star I^\star = 0, \tag{6.14}$$

where $(\mathbf{N}^\star, I^\star)$ is the optimal solution of the constrained problem (6.6).

Next, we show that finding the Lagrange multipliers $\mu_i^\star, i = 0, \cdots, k+2$ and the optimal solution $(\mathbf{N}^\star, I^\star)$ to satisfy the KKT conditions (6.9) to (6.14) is equivalent to solving the constrained problem (6.3) by the algorithm. Suppose that $\mathbf{N}^\star$ satisfies $N_i^\star > N_{\min}, \forall i$ and therefore $\mathbf{N}^\star$ is an interior solution[5]. (In all our experiments, the optimal solutions have this feature.) In this case, by condition (6.12), we know that $\mu_i^\star = 0, \forall i = 1, \cdots, k$. In addition, the optimal solution should satisfy $I^\star < \sum_{i=1}^k N_i^\star$ because otherwise (if $I^\star = \sum_{i=1}^k N_i^\star$) the production rate of the loop would be 0 because the entire loop will be full of parts and there will be no part movement at all due to the way that a loop system is modeled[6]. Similarly, the production rate of the loop system will be 0 if $I^\star = 0$ since there are no parts traveling through the loop. Hence $I^\star$ has to be greater than 0. Thus, according to (6.13) and (6.14), we know that $\mu_{k+1}^\star = \mu_{k+2}^\star = 0$. Consequently, we simplify the KKT conditions (6.9) to (6.14) to

$$-\begin{pmatrix} \frac{\partial J(\mathbf{N}^\star, I^\star)}{\partial N_1} \\ \vdots \\ \frac{\partial J(\mathbf{N}^\star, I^\star)}{\partial N_k} \\ \frac{\partial J(\mathbf{N}^\star, I^\star)}{\partial I} \end{pmatrix} - \mu_0^\star \begin{pmatrix} \frac{\partial P(\mathbf{N}^\star, I^\star)}{\partial N_1} \\ \vdots \\ \frac{\partial P(\mathbf{N}^\star, I^\star)}{\partial N_k} \\ \frac{\partial P(\mathbf{N}^\star, I^\star)}{\partial I} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \tag{6.15}$$

$$\mu_0^\star \left(\hat{P} - P(\mathbf{N}^\star, I^\star) \right) = 0, \tag{6.16}$$

where $\mu_0^\star \geq 0$. We know, since $(\mathbf{N}^\star, I^\star)$ is not the optimal solution of the unconstrained problem, that $\nabla J(\mathbf{N}^\star, I^\star) \neq 0$. $\nabla J(\mathbf{N}^\star, I^\star) \neq 0$ means that not all

[5]We provide the proof for the case where not all $N_i^\star$ satisfy $N_i^\star > N_{\min}$ in Appendix E.

[6]Although we do not provide any formal mathematic proofs, we know that the optimal solution should satisfy $2I^\star \leq \sum_{i=1}^k N_i^\star$ and this indicates that $I^\star \neq \sum_{i=1}^k N_i^\star$. This is because, due to the symmetry of $P(\mathbf{N}, I)$ in I, if $I^\star > 0.5 \sum_{i=1}^k N_i^\star$, we will be able to find another $I' = \sum_{i=1}^k N_i - I^\star$ that gives the same production rate with lower cost and therefore higher profit. Therefore, $I^\star$ must not be larger than $0.5 \sum_{i=1}^k N_i^\star$.

$\partial J(\mathbf{N}^\star)/\partial N_i$ or $\partial J(\mathbf{N}^\star)/\partial I$ are equal to 0. Thus, $\mu_0^\star \neq 0$ since otherwise condition (6.15) would be violated. By condition (6.16), the optimal solution $(\mathbf{N}^\star, I^\star)$ satisfies $P(\mathbf{N}^\star, I^\star) = \hat{P}$.

Conditions (6.15) and (6.16) reveal how we could find $\mu_0^\star$ and ($\mathbf{N}^\star$, $I^\star$). For every $\mu_0^\star$, (6.15) determines ($\mathbf{N}^\star$, $I^\star$) since there are $k+1$ equations and $k+1$ unknowns. Therefore, we can think of $\mathbf{N}^\star = \mathbf{N}^\star(\mu_0^\star)$ and $I^\star = I^\star(\mu_0^\star)$. We search for a value of $\mu_0^\star$ such that $P(\mathbf{N}^\star(\mu_0^\star), I^\star(\mu_0^\star)) = \hat{P}$. As we indicate in the following, this is exactly what the algorithm does.

Replacing $\mu_0^\star$ by $\mu_0 > 0$ in constraint (6.15) gives

$$-\begin{pmatrix} \dfrac{\partial J(\bar{\mathbf{N}}, \bar{I})}{\partial N_1} \\ \vdots \\ \dfrac{\partial J(\bar{\mathbf{N}}, \bar{I})}{\partial N_k} \\ \dfrac{\partial J(\bar{\mathbf{N}}, \bar{I})}{\partial I} \end{pmatrix} - \mu_0 \begin{pmatrix} \dfrac{\partial P(\bar{\mathbf{N}}, \bar{I})}{\partial N_1} \\ \vdots \\ \dfrac{\partial P(\bar{\mathbf{N}}, \bar{I})}{\partial N_k} \\ \dfrac{\partial P(\bar{\mathbf{N}}, \bar{I})}{\partial I} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \tag{6.17}$$

where $(\bar{\mathbf{N}}, \bar{I})$ is the unique solution of (6.17). Note that $(\bar{\mathbf{N}}, \bar{I})$ is the solution of the following optimization problem

$$\begin{aligned} \min_{\mathbf{N},I} \quad & -\bar{J}(\mathbf{N}, I) = -J(\mathbf{N}, I) + \mu_0 \left(\hat{P} - P(\mathbf{N}, I)\right) \\ \text{subject to} \quad & N_{\min} - N_i \leq 0, \forall i = 1, \cdots, k, \\ & I - \sum_{i=1}^{k} N_i \leq 0, \\ & -I \leq 0, \end{aligned} \tag{6.18}$$

which is equivalent to

$$\begin{aligned} \max_{\mathbf{N},I} \quad & \bar{J}(\mathbf{N},I) = J(\mathbf{N},I) - \mu_0 \left(\hat{P} - P(\mathbf{N},I)\right) \\ \text{subject to} \quad & N_{\min} - N_i \leq 0, \forall i = 1, \cdots, k, \\ & I - \sum_{i=1}^{k} N_i \leq 0, \\ & -I \leq 0, \end{aligned} \tag{6.19}$$

or

$$\begin{aligned} \max_{\mathbf{N},I} \quad & \bar{J}(\mathbf{N},I) = AP(\mathbf{N},I) - \sum_{i=1}^{k} b_i N_i - cI - \mu_0 \left(\hat{P} - P(\mathbf{N},I)\right) \\ \text{subject to} \quad & N_{\min} - N_i \leq 0, \forall i = 1, \cdots, k, \\ & I - \sum_{i=1}^{k} N_i \leq 0, \\ & -I \leq 0, \end{aligned} \tag{6.20}$$

or

$$\begin{aligned} \max_{\mathbf{N},I} \quad & \bar{J}(\mathbf{N},I) = (A+\mu_0)P(\mathbf{N},I) - \sum_{i=1}^{k} b_i N_i - cI \\ \text{subject to} \quad & N_i \geq N_{\min}, \forall i = 1, \cdots, k, \\ & \sum_{i=1}^{k} N_i \geq I, \\ & I \geq 0, \end{aligned} \tag{6.21}$$

or, finally,

$$\begin{aligned} \max_{\mathbf{N},I} \quad & \bar{J}(\mathbf{N},I) = A'P(\mathbf{N},I) - \sum_{i=1}^{k} b_i N_i - cI \\ \text{subject to} \quad & N_i \geq N_{\min}, \forall i = 1, \cdots, k, \\ & \sum_{i=1}^{k} N_i \geq I, \\ & I \geq 0, \end{aligned} \tag{6.22}$$

where $A' = A + \mu_0$. This is exactly the unconstrained problem (6.5) and $(\bar{\mathbf{N}}, \bar{I})$ is its optimal solution. Note that $\mu_0 > 0$ indicates that $A' > A$. In addition, the KKT condition (6.16) indicates that the optimal solution of the constrained problem $(\mathbf{N}^\star, I^\star)$ satisfies $P(\mathbf{N}^\star, I^\star) = \hat{P}$. This means that, for every $A' > A$ (or $\mu_0 > 0$), we can find the corresponding optimal solution $(\bar{\mathbf{N}}, \bar{I})$ satisfying condition (6.17) by solving problem (6.5), and, we need to find the A' such that the solution to problem (6.5), denoted as $(\mathbf{N}(A'), I(A'))$, satisfies $P(\mathbf{N}(A'), I(A')) = \hat{P}$. Then, $\mu_0 = A' - A$ and $(\mathbf{N}(A'), I(A'))$ satisfy conditions (6.15) and (6.16). Hence, $\mu_0 = A' - A$ is exactly the Lagrange multiplier satisfying the KKT conditions of the constrained problem, and $(\mathbf{N}^\star, I^\star) = (\mathbf{N}(A'), I(A'))$ is the optimal solution of the constrained problem. Consequently, solving the constrained problem (6.3) through the algorithm is essentially finding the unique Lagrange multipliers and optimal solution of the problem. We have proven the assertion.

6.4 Numerical Experiments

In this section, we provide numerical experiments of the profit maximization algorithm for single closed-loops. In particular, when we conduct numerical experiments, we focus on three-machine three-buffer closed-loops and four-machine four-buffer closed-loops. The reasons of this include the following two points:

1. Although the number of machines in a transfer line or a closed-loop system can be greater than 3 or 4, the number of buffers could be much smaller than the number of machines. When modeling of both a transfer line and a closed-loop system, it is the number of buffers that determines the structure and the scale of the system. This is because consecutive machines without buffers can be grouped together and modeled as a single machine with the technique introduced in Gershwin (1994). Therefore, three-machine three-buffer loop systems and four-machine four-buffer loop systems could be quite common in real industry settings.

   For example, Werner (2001) studies an industry case for a Japanese electronics company who produces a network connection device that is used to improve the quality of signals transmitted over long distances. The system is modeled as a four-machine four-buffer closed-loop and the production rate of the system is studied as a function of the loop population. In addition, Resano Lázaro and Luis Pérez (2008), (2009) model an automobile assembly line as a network of four closed-loops coupled together, where three of them have at most four buffers[7]. Ip et al. (2007) analyze a lamp assembly production system that consists of five closed-loops. In particular, four of these loops have four or fewer machines each. Miller et al. (2010) indicate in a case study that a furniture production company utilizes a CONWIP operational model that can be considered as a three-machine three-buffer loop.

2. The second reason is that although two modification about loop evaluation has been developed in Chapter 5 and they significant mitigate the batman effect (i.e., undesirable jump of $P(\mathbf{N}, I)$ in loop evaluation introduced in Section 5.2.3), there is still minor remaining batman effect that potentially affects the accuracy of loop evaluation and eventually loop optimization[8]. Therefore, it is valuable to first well understand the accuracy of the proposed optimization algorithm

[7]The last one is a five-machine five-buffer loop.

[8]Thoughts about how to further reduce the batman effect and therefore to improve the loop evaluation accuracy are discussed as future work in Chapter 10.

in small loop systems before we move to larger systems in the future. From this perspective, a good understanding about small closed-loop systems such as three-machine three-buffer loops and four-machine four-buffer loops is valuable.

For these reasons, in this section, we concentrate on three-machine three-buffer loops and four-machine four-buffer loops for numerical experiments. However, it is necessary to point out that the loop optimization algorithm presented in this chapter applies to larger systems and finds accurate optimal results as well, provided that the batman effect in loop evaluation is further mitigated or eventually eliminated.

6.4.1 Three-Machine Three-Buffer Closed-Loops

We start with a balanced loop with identical machines. In addition, buffer cost coefficients are also identical, which means that $b_i = c_i = 1, \forall i$. The parameters of the first three-machine three-buffer closed-loop system is listed in Table 6.3. The target production rate is $\hat{P} = .87$. The revenue coefficient is $A = 1500$.

Table 6.3: Parameters of three-machine closed-loop, Experiment 1

machine	M_1	M_2	M_3
r_i	.10	.10	.10
p_i	.01	.01	.01
$r_i/(r_i + p_i)$	.909	.909	.909

As in Chapter 4 for transfer lines, in order to verify the optimal solution, we compare the optimal solution from the algorithm with that from $\hat{P}$ surface search. The results from both the algorithm and the surface search are summarized in Table 6.4. The optimal solutions from the algorithm and from the $\hat{P}$ surface search are identical. The computer time for the algorithm to find the optimal solution is 42.31 seconds.

Next, we break the balance of the previous three-machine loop system by changing the buffer cost coefficient b_2 to 2 while keeping other parameters unchanged. The target production rate is still $\hat{P} = .87$. Since it is more expensive to create buffer

Table 6.4: Results of three-machine closed-loop, Experiment 1

	$\hat{P}$ Surface Search	The algorithm	error
$P(\mathbf{N}^\star, I^\star)$	.8701	.8701	0%
$N_1^\star$	35	35	0%
$N_2^\star$	36	36	0%
$N_3^\star$	36	36	0%
$I^\star$	51	51	0%
$\bar{n}_1(\mathbf{N}^\star, I^\star)$	16.8156	16.8156	0%
$\bar{n}_2(\mathbf{N}^\star, I^\star)$	17.1314	17.1314	0%
$\bar{n}_3(\mathbf{N}^\star, I^\star)$	17.4197	17.4197	0%
$J(\mathbf{N}^\star)$ ($/time unit)	1146.77	1146.77	0%

space between Machines M_2 and M_3, we expect $N_2^\star$ to be smaller as compared to the result in the previous example, while $N_1^\star$ and $N_3^\star$ to be larger such that the target production rate can be satisfied.

The results from both the algorithm and the surface search are summarized in Table 6.5. The optimal solutions from the algorithm and from the $\hat{P}$ surface search are identical. The computer time for the algorithm to find the optimal solution is 54.18 seconds.

Table 6.5: Results of three-machine closed-loop, Experiment 2

	$\hat{P}$ Surface Search	The algorithm	error
$P(\mathbf{N}^\star, I^\star)$	.8702	.8702	0%
$N_1^\star$	38	38	0%
$N_2^\star$	32	32	0%
$N_3^\star$	38	38	0%
$I^\star$	52	52	0%
$\bar{n}_1(\mathbf{N}^\star, I^\star)$	19.3095	19.3095	0%
$\bar{n}_2(\mathbf{N}^\star, I^\star)$	15.5497	15.5497	0%
$\bar{n}_3(\mathbf{N}^\star, I^\star)$	17.4461	17.4461	0%
$J(\mathbf{N}^\star)$ ($/time unit)	1112.93	1112.93	0%

6.4.2 Four-Machine Four-Buffer Closed-Loops

We consider two four-machine closed-loops. In particular, buffer cost coefficients are also identical, which means that $b_i = c_i = 1, \forall i$. The parameters of the first four-machine closed-loop system is listed in Table 6.6. The target production rate is $\hat{P} = .86$, while the revenue coefficient is $A = 2000$.

Table 6.6: Parameters of four-machine closed-loop, Experiment 1

machine	M_1	M_2	M_3	M_4
r_i	.10	.20	.10	.10
p_i	.01	.01	.01	.01
$r_i/(r_i + p_i)$	.909	.952	.909	.909

The results from both the algorithm and $\hat{P}$ surface search for this system are summarized in Table 6.7. The optimal solutions from the algorithm and from the $\hat{P}$ surface search are identical. The computer time for the algorithm to find the optimal solution is 195.19 seconds.

Table 6.7: Results of four-machine closed-loop, Experiment 1

	$\hat{P}$ Surface Search	The algorithm	error
$P(\mathbf{N}^\star, I^\star)$	.8600	.8600	0%
$N_1^\star$	21	21	0%
$N_2^\star$	20	20	0%
$N_3^\star$	31	31	0%
$N_4^\star$	33	33	0%
$I^\star$	50	50	0%
$\bar{n}_1(\mathbf{N}^\star, I^\star)$	9.6371	9.6371	0%
$\bar{n}_2(\mathbf{N}^\star, I^\star)$	10.6250	10.6250	0%
$\bar{n}_3(\mathbf{N}^\star, I^\star)$	14.1320	14.1320	0%
$\bar{n}_4(\mathbf{N}^\star, I^\star)$	15.8885	15.8885	0%
$J(\mathbf{N}^\star)$ ($/time unit)	1565.01	1565.01	0%

In the second example, we consider different machines. Buffer cost coefficients are $b_1 = 0.72$, $b_2 = 1.29$, $b_3 = 1.79$, $b_4 = 1.73$, and $c_i = 0.59, \forall i$. The parameters of the loop system is listed in Table 6.8. The target production rate is $\hat{P} = .86$, while the

revenue coefficient is $A = 1500$.

Table 6.8: Parameters of four-machine closed-loop, Experiment 2

machine	M_1	M_2	M_3	M_4
r_i	.116	.125	.102	.133
p_i	.007	.009	.005	.008
$r_i/(r_i+p_i)$	.943	.934	.953	.943

Given the parameters of machines and buffers, the production rate constraint is inactive for the loop. Therefore, we conduct a surface search of $(\mathbf{N}, I)$ such that $P(\mathbf{N}, I) \geq \hat{P}$ for comparison. The results are summarized in Table 6.9. The maximum error between the algorithm and the surface search is 6.25% and it appears at $N_3^\star$. However, it can be seen that the optimal solution from the algorithm satisfies the production rate constraint and the profit associated with it is very close to the one from the surface search (the error is 0.01%). The computer time for the algorithm to find the optimal solution this is 31.31 seconds.

Table 6.9: Results of four-machine closed-loop, Experiment 2

	Surface Search	The algorithm	error
$P(\mathbf{N}^\star, I^\star)$	.8647	.8651	0.05%
$N_1^\star$	16	17	0.00%
$N_2^\star$	12	12	0.00%
$N_3^\star$	8	8	6.25%
$N_4^\star$	8	8	0.00%
$I^\star$	20	20	0.00%
$\bar{n}_1(\mathbf{N}^\star, I^\star)$	6.4401	6.6665	3.52%
$\bar{n}_2(\mathbf{N}^\star, I^\star)$	6.1685	6.1736	0.08%
$\bar{n}_3(\mathbf{N}^\star, I^\star)$	4.2994	4.2865	0.30%
$\bar{n}_4(\mathbf{N}^\star, I^\star)$	3.1268	2.9622	5.26%
$J(\mathbf{N}^\star)$ (\$/time unit)	1230.01	1229.88	0.01%

6.4.3 More Numerical Experiments

Finally, we provide numerical experiments for 100 three-machine closed-loop systems and 100 four-machine closed-loop systems. These systems are generated randomly according to the method of Gershwin (2011). In all these lines, the isolated production rate $P_i = r_i/(r_i + p_i)$ of any given machine is between .909 and .938 for three-machine loops and between .923 and .952 for four-machine loops with r_i and p_i generated randomly. In addition, the buffer cost coefficients b_i and c_i for any buffer are also generated randomly with the restriction that all c_i's be the same. The target production rate $\hat{P}$ is .87 for three-machine loops and .86 for four-machine loops. The revenue coefficient A is 1500 for the three-machine loops and 2000 for the four-machine loops. We compare the results from the algorithm with surface search and compute four types of error. They are the production rate error, the profit error, the maximum buffer size error, and the loop invariant error. The subscripts *alg* and *ss* are used to represent the optimal buffer allocations associated with the algorithm and the surface search, respectively. The four types of error are defined as

$$J_{\text{err}} = \left| \frac{J(\mathbf{N}^\star_{\text{ss}}, I^\star_{\text{ss}}) - J(\mathbf{N}^\star_{\text{alg}}, I^\star_{\text{alg}})}{J(\mathbf{N}^\star_{\text{ss}}, I^\star_{\text{ss}})} \right| \times 100\%,$$

$$P_{\text{err}} = \left| \frac{P(\mathbf{N}^\star_{\text{ss}}, I^\star_{\text{ss}}) - P(\mathbf{N}^\star_{\text{alg}}, I^\star_{\text{alg}})}{P(\mathbf{N}^\star_{\text{ss}}, I^\star_{\text{ss}})} \right| \times 100\%,$$

$$N_{\text{err}} = \max_{i=1,\cdots,k} \left\{ \left| \frac{\mathbf{N}^\star_{\text{ss}}(B_i) - \mathbf{N}^\star_{\text{alg}}(B_i)}{\mathbf{N}^\star_{\text{ss}}(B_i)} \right| \times 100\% \right\},$$

and finally

$$I_{\text{err}} = \left| \frac{I^\star_{\text{ss}} - I^\star_{\text{alg}}}{I^\star_{\text{ss}}} \right| \times 100\%.$$

The four types of errors for the 100 three-machine loops are illustrated in Figure 6-7. We rank the four types of errors in their corresponding ascending orders respectively. (Therefore, the ith case in the profit error graph, for instance, may not necessary be the same as the ith case in the production rate error graph.) The

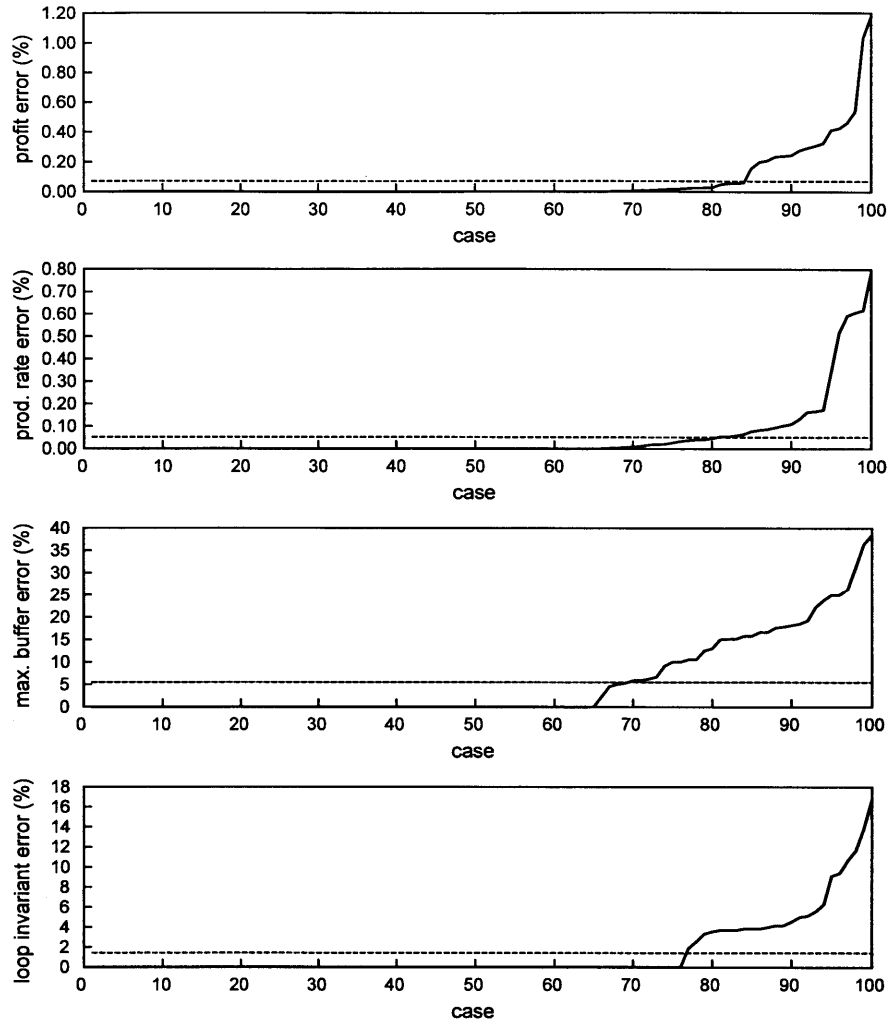


Figure 6-7: Results of one hundred randomly generated three-machine closed-loops

average error of each type is also provided. In particular, in 65 out of the 100 cases, the optimal buffer sizes from the algorithm and the surface search are the same, and therefore the four types of errors in these 65 cases are 0. In addition, the average profit error, the average production rate error, the average maximum buffer size error, and the average loop invariant error of these 100 cases are .069%, .052%, 5.48%, and 1.44%, respectively.

The four types of errors for the 100 four-machine loops are illustrated in Figure 6-8. In particular, in 52 out of the 100 cases, the optimal buffer sizes from the algorithm and the surface search are the same, and therefore the four types of errors in these 52

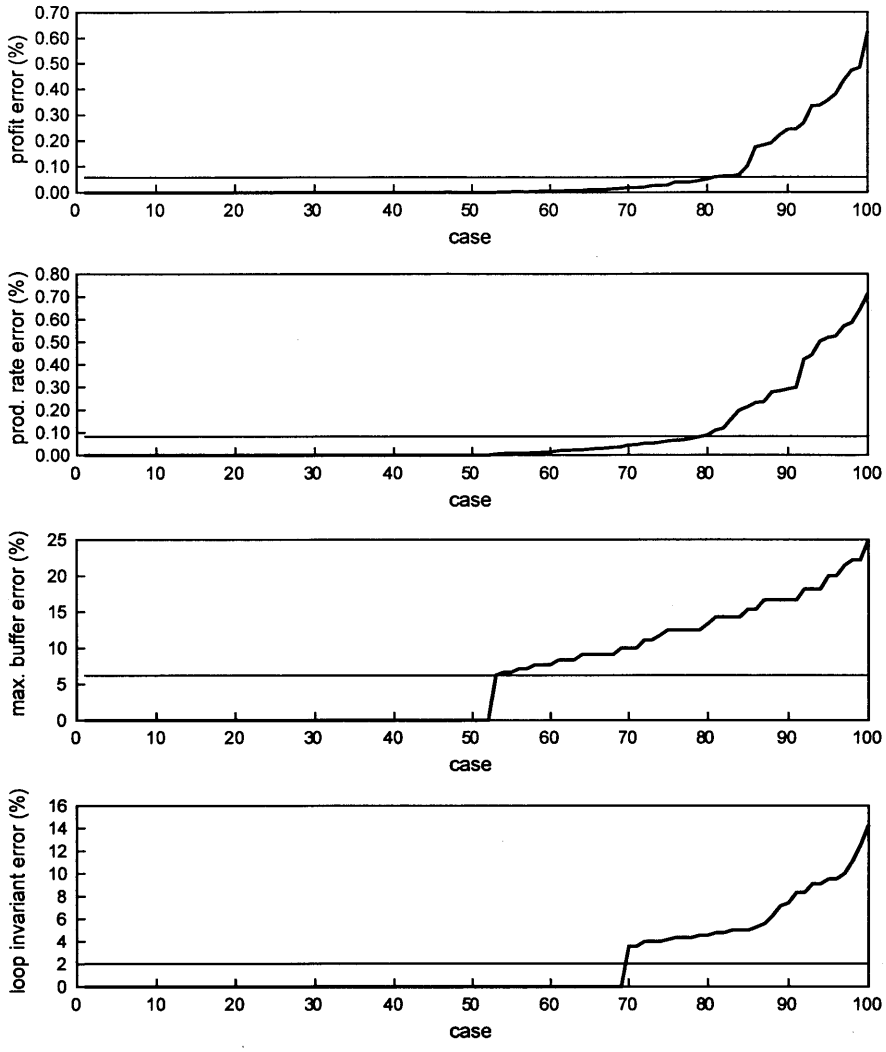


Figure 6-8: Results of one hundred randomly generated four-machine closed-loops

cases are 0. In addition, the average profit error, the average production rate error, the average maximum buffer error, and the average loop invariant error of these 100 cases are .025%, .084%, 6.24%, and 2.03%, respectively.

In these 200 numerical experiments for three-machine and four-machine closed-loops, although the maximum buffer size error and the loop invariant error can be large, the profit error and the production rate error are always small. These experiments demonstrate the accuracy and the reliability of the proposed algorithm.

6.5 Summary

In this chapter, we extend the profit maximization algorithm developed in Chapter 4 to closed-loop systems. We show in Section 6.1 that it is possible that the profit of a closed-loop system has multiple local maxima and therefore we focus the scope of our problem to the case where the average inventory cost coefficients of all buffers (c_i) are the same, which is also a common situation in many actual manufacturing settings. Under this scope, with the concavity assumption of $P(\mathbf{N}, I)$ discussed in Section 6.2, the profit of a closed-loop system has only one single maximum and therefore the algorithm of Chapter 4 can be extended to closed-loops directly.

The performance of the algorithm is shown by studying a set of three-machine and four-machine closed-loop systems. The reasons why we focus on smaller systems are discussed in Section 6.4. All these numerical experiments studied in this chapter show the accuracy of the proposed algorithm.

Chapter 7

Maximum Part Waiting Time Constraint between Adjacent Operations

7.1 Motivation

In some kinds of manufacturing, the time a part may spend in a buffer between successive operations (such as cleaning and baking in semiconductor fabrication) is limited. Parts that wait longer than this time must be reworked or discarded due to the risk of quality degradation. This can be seen as a maximum part waiting time constraint between operations. It says that the time for a part to wait for the next operation after the previous operation should be kept less than a fixed value, to guarantee the quality of the part. This constraint is common and important in several industries, especially the semiconductor industry (Lee and Park 2005, Kitamura et al. 2006). As examples, Kim and Lee (2008) and Rostami et al. (2001) indicate that the time a wafer spends in a processing module within a cluster tool should be limited. Kim et al. (2003) point out when the wafer delay at each process step of low pressure chemical vapor deposition (LPCVD) exceeds 20 seconds, the wafer surface deteriorates because of excessive exposure to residual gases under high temperature

and the wafer is scrapped. Robinson and Giglio (1999) mention that a baking operation must be started within two hours of a prior clean operation. If more than two hours elapse, the lot must be sent back to be cleaned again. Lu et al. (1994) study the efficient scheduling policies to reduce mean and variance of cycle-time, and point out that the shorter the period that wafers are exposed to aerial contaminants while waiting for processing, the smaller is the yield loss. Yang and Chern (1995) indicate the consideration of such a part waiting time constraint in food production, chemical production, and steel production. For surveys, see Neacy et al. (1994) and Uzsoy et al. (1992).

In this chapter, we extend the algorithm for production line profit maximization to cope with this additional maximum part waiting time constraint. In other words, we will consider both the production rate constraint and the maximum part waiting time constraint. However, we want to clearly point out that this constraint can be imposed on a single buffer, on more than one buffer, or on the entire manufacturing process. For our purpose, we will assume that the maximum part waiting time constraint is imposed on a single buffer $B_{\hat{i}}$.

Because of the randomness of machine failures, it is impractical to require the waiting times of all parts that enter Buffer $B_{\hat{i}}$ to be bounded. However, it is possible to statistically assure the waiting times of at least a given percentage of parts to be upper bounded. That is, we can require

$$\mathbf{p}\left(T(\mathbf{N}) \leq W_{\hat{i}}\right) \geq 1 - \alpha \tag{7.1}$$

where $W_{\hat{i}}$ is the maximum part waiting time allowed at Buffer $B_{\hat{i}}$, $T(\mathbf{N})$ is a random variable that indicates the part waiting time (and it is a function of buffer sizes $\mathbf{N}$), and $1 - \alpha$ is the given percentage. With this constraint, the production line profit maximization problem becomes

$$
\begin{aligned}
\max_{\mathbf{N}} \quad & J(\mathbf{N}) = AP(\mathbf{N}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i(\mathbf{N}) \\
\text{subject to} \quad & P(\mathbf{N}) \geq \hat{P} \\
& \mathbf{p}\left(T(\mathbf{N}) \leq W_{\hat{i}}\right) \geq 1-\alpha \\
& N_i \geq N_{\min}, \quad \forall i = 1, \cdots, k-1.
\end{aligned}
\tag{7.2}
$$

In order to quantify the maximum part waiting time constraint, we derive an analytical formulation for the part waiting time distribution for two machine lines and apply it to a single buffer in long lines with the help of decomposition in Section 7.2. This allows us to compute the probability $\mathbf{p}(T(\mathbf{N}) \leq W_{\hat{i}})$. However, in the analytical formulation, we do not have a closed form expression for $\mathbf{p}(T(\mathbf{N}))$. Instead, the probability distribution is computed numerically through iteration. Due to the lack of a closed form expression, we cannot treat the decision variable $\mathbf{N}$ as a continuous variable when computing $\mathbf{p}(T(\mathbf{N}) \leq W_{\hat{i}})$ in solving (7.2). To resolve this concern, we transform (7.2) to a similar problem where Constraint (7.1) is replaced by an average part waiting time constraint. The transformed problem will be solved and its solution will be checked against (7.1) iteratively. We discuss the transformed problem in great detail in Section 7.3.

The rest of the chapter is organized as follows. We derive the analytical solution of the part waiting time distribution for two-machine lines in Section 7.2. A transformed profit maximization problem is introduced in detail in Section 7.3. We extend the profit maximization algorithm derived in Chapter 4 to solve the transformed problem. Numerical experiments are provided in Section 7.4 to show the accuracy and reliability of the proposed algorithm.

7.2 Part Waiting Time Distribution for Two-Machine Lines

7.2.1 Derivation

In this section, we derive the part waiting time distribution for two-machine lines. In particular, we consider the Gershwin (1994) version of the Buzacott model (Buzacott 1967a). However, the approach we use here can be applied as well to the deterministic time and discrete material model of Tolio et al. (2002), which allows both machines to have multiple failure modes (Shi and Gershwin 2011a).

Recall that in Buzacott model, the operation times for both machines are deterministic, identical and set as the time unit. Either machine can produce exactly one part in each time unit if it is not down in a failed state, blocked, or starved. In addition, machine failures and repairs follow geometric distributions whose means are measured in terms of the time unit. As a result, part waiting times are integers in this model. In addition, we want to indicate clearly that we assume that the buffer under consideration is a FIFO buffer, which means that parts inside the buffer follow a first-in first-out discipline.

The convention in the two-machine line model is crucial in defining the part waiting time in the buffer. We emphasize it here, define the part waiting time, and derive the distribution of the part waiting time according to the convention. By the convention of Gershwin (1994), the status of both machines change at the beginning of a time unit while buffer level changes at the end of a time unit. This implies that any new part produced by the upstream machine M_1 will enter the buffer at the end of the current time unit, or equivalently, the same instant as the beginning of the next time unit. To clarify this, consider Figure 7-1.

In Figure 7-1, there is a horizontal time axis on which the discrete time instants $t-1$, t, and $t+1$ represent the beginning of the time units after them and the end of the time units before them, respectively. For example, time instant t is the beginning of time unit t, while time instant $t+1$ is the beginning of time unit $t+1$ as well as the

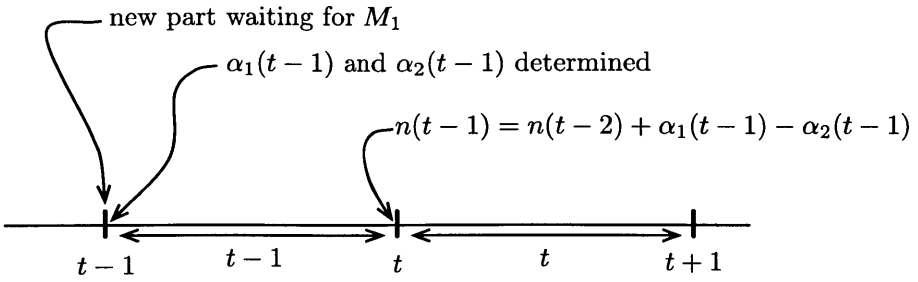


Figure 7-1: Convention of Gershwin (1994) version of the Buzacott model

end of time unit t. The time interval between the time instants t and $t+1$ represents time unit t.

In the model convention of Gershwin (1994), machine states get updated at the beginning of a time unit, while buffer level gets updated at the end of a time unit. At the beginning of time unit $t-1$, the states of the two machines $\alpha_1(t-1)$ and $\alpha_2(t-1)$ are determined. At the end of time unit $t-1$, the buffer level $n(t-1)$ is determined. According to Gershwin (1994), the relationship between machine states and the buffer level is $n(t-1) = n(t-2) + \alpha_1(t-1) - \alpha_2(t-1)$, as it is shown in Figure 7-1 (assuming $1 \leq n(t-1) \leq N-1$).

Now, suppose that a new part is waiting to be produced by the upstream machine M_1 at time instant $t-1$. At the beginning of time unit $t-1$, the states of the two machines $\alpha_1(t-1)$ and $\alpha_2(t-1)$ are determined. If the upstream machine is up, then the new part will be added into the buffer at the end of time unit $t-1$, which is when the buffer level of time unit $t-1$ is determined. Therefore, this new part will enter the buffer at the end of time unit $t-1$ and it will be the $n(t-1)$th part in the buffer. Since the end of time unit $t-1$ is exactly the same instant as the beginning of time unit t, it will be convenient to say that the part above enters the buffer at the beginning of time unit t in the rest of this chapter. This is why any new part produced by the upstream machine will enter the buffer at the beginning of a time unit. The waiting time of the part is then counted from the instant it enters until the instant it leaves the buffer at the end of some time unit in the future.

Suppose that at the beginning of a time unit, a new part enters an empty buffer

and the downstream machine M_2 is up. Then M_2 will work on that part during that time unit. The part will leave the buffer at the end of that time unit and we consider that time unit as the waiting time in the buffer of that part. Therefore, the minimum waiting time for any given part is one time unit. Let $T(N)$ be a random variable that indicates the waiting time of the last part to enter the buffer and it is a function of the buffer size N of the two-machine line (as well as machine parameters). In what follows, we derive the probability mass function (PMF) of part waiting time $\mathbf{p}(T(N) = \tau)$ where τ is a positive integer[1].

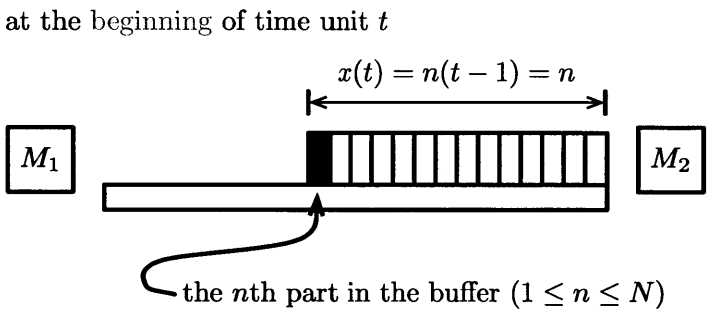


Figure 7-2: Position of the new part that enters the buffer at the beginning of time unit t

Suppose a new part enters the buffer at the beginning of some time unit t. We derive the part waiting time distribution for this specific part and this part is called the *target part*. We start the derivation by defining the position of the new part in the buffer. We assume that the part that enters the buffer at the beginning of time unit t is the nth ($1 \leq n \leq N$) part in the buffer (and there are $n-1$ parts already in the buffer in front of it). Specifically, we interpret n as the position of the new part. In other words, if at the beginning of time unit t there are $n-1$ parts in the buffer and the target part enters it, then the position of the target part is n. Define $x(t)$ to be the position of the target part that enters the buffer at the beginning of time unit t. According to the model convention, the buffer level gets updated at the end of a time unit. Therefore, the buffer level at the beginning of time unit t is exactly

[1]The part waiting time depends on both the size of the buffer and the parameters of the two machines. However, since (for production lines) the decision variables considered throughout the thesis are buffer sizes, we choose to use the notation $T(N)$.

the same as the buffer level at the end of time unit $t-1$. Therefore, once the target part arrives, its position is $x(t) = n(t-1) = n$ (see Figure 7-2 for illustration). If we can find the conditional probability that the waiting time of the part is $T(N) = \tau$ given that its position is $x(t) = n$ when it enters the buffer, then we can find the unconditional probability $\mathbf{p}(T(N) = \tau)$ by the Total Probability Theorem (Bertsekas and Tsitsiklis 2008).

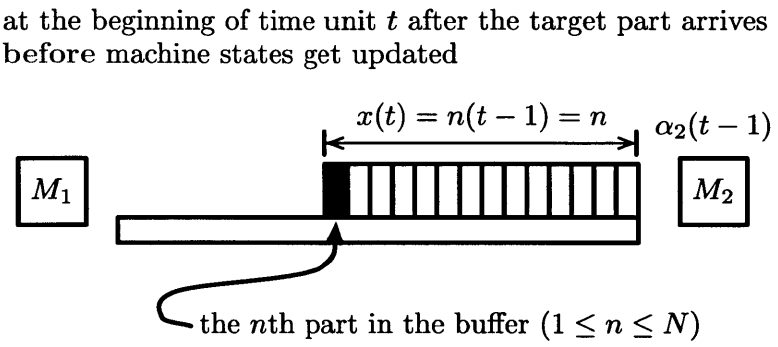


Figure 7-3: State of M_2 at the beginning of time unit t before it gets updated

On the other hand, once the target part enters the buffer, its waiting time in the buffer has nothing to do with the state of the upstream machine M_1. It depends only on the state of the downstream machine M_2 after the target part enters the buffer. (For instance, if M_2 fails for a very long time, then the part will have to wait in the buffer for that long failure period of M_2 and its waiting time will be long.) According to the model convention, machine states get updated at the beginning of a time unit. This means that once they are updated, the state of M_2 is $\alpha_2(t)$ during time unit t. However, in order to derive $\mathbf{p}(T(N) = \tau)$, we are interested in the state of M_2 at the instant that the target part enters the buffer and before α_2 gets updated. At that moment, both machines still assume their respective states from the previous time unit (i.e., $\alpha_1(t-1)$ and $\alpha_2(t-1)$). Therefore, at the instant that the target part enters the buffer and before the machine states get updated, the state of M_2 is $\alpha_2(t-1)$ (see Figure 7-3).

According to the discussion above, we see that $T(N)$ depends on both the position of the target part $x(t)$ and the state of M_2 before it gets updated $\alpha_2(t-1)$. Hence,

we need to find $\mathbf{p}(T(N) = \tau | x(t) = n, \alpha_2(t-1) = 1)$ and $\mathbf{p}(T(N) = \tau | x(t) = n, \alpha_2(t-1) = 0)$. Once we find them, the unconditional probability $\mathbf{p}(T(N) = \tau)$ can be found by

$$\mathbf{p}(T(N) = \tau) =$$

$$\sum_{n=1}^{N} \Big[\mathbf{p}(T(N) = \tau | x(t) = n, \alpha_2(t-1) = 1) \mathbf{p}(x(t) = n, \alpha_2(t-1) = 1) \tag{7.3}$$

$$+ \mathbf{p}(T(N) = \tau | x(t) = n, \alpha_2(t-1) = 0) \mathbf{p}(x(t) = n, \alpha_2(t-1) = 0) \Big].$$

Note that although both $\mathbf{p}(x(t) = n, \alpha_2(t-1) = 1)$ and $\mathbf{p}(x(t) = n, \alpha_2(t-1) = 0), 1 \leq n \leq N$ look like unconditional probabilities, they imply an underlying condition that there is indeed a new part entering the buffer at the beginning of time unit t. In other words, the universal set in our probabilistic model is established on the condition that a new part enters the buffer at the beginning of some time unit t. Therefore, for instance, $\mathbf{p}(x(t) = n, \alpha_2(t-1) = 1)$ should be read as the probability that, given a new part entering the buffer at the beginning of time unit t, that target part has a position of n and M_2 is up before it gets updated. To verify this, note that $\mathbf{p}(T(N) = \tau)$ is the probability that the waiting time of the part, which enters the buffer at the beginning of time unit t, is $T(N) = \tau$. Therefore, the underlying condition is that a part enters the buffer at the beginning of time unit t. These two sets of probabilities can be derived easily from the analytical solution of two-machine line evaluation developed by Gershwin (1994), and we will discuss them later in this section. Therefore, the difficulty in deriving $\mathbf{p}(T(N) = \tau)$ is to find $\mathbf{p}(T(N) = \tau | x(t) = n, \alpha_2(t-1) = 1)$ and $\mathbf{p}(T(N) = \tau | x(t) = n, \alpha_2(t-1) = 0)$. We would like to point out that due to the convention in the two-machine line model, if the waiting time of the new part with a position n is τ, then $\mathbf{p}(T(N) = \tau | x(t) = n, \alpha_2(t-1) = 1) = \mathbf{p}(T(N) = \tau | x(t) = n, \alpha_2(t-1) = 0) = 0, \forall \tau < n$. This is because it will at least take n time units before the nth part can leave the buffer.

In the following derivation, for convenience, define

$$p_t(\tau, n) = \mathbf{p}(T(N) = \tau | x(t) = n, \alpha_2(t-1) = 1),$$

and

$$q_t(\tau, n) = \mathbf{p}(T(N) = \tau | x(t) = n, \alpha_2(t-1) = 0).$$

We find $p_t(\tau, n)$ and $q_t(\tau, n)$ by iteration. Recall that the target part under consideration enters the buffer at the beginning of time unit t and $x(t) = n$. Suppose $\alpha_2(t-1) = 1$ for now. Let us discuss what may happen on Machine M_2 during time unit t. Since the status of machines are updated at the beginning of a time unit, M_2 can fail during time unit t with probability p_2 (i.e., $\alpha_2(t) = 0$) or it can remain up during time unit t with probability $1 - p_2$ (i.e., $\alpha_2(t) = 1$). We consider them separately:

- if M_2 fails with probability p_2, then it cannot process any part in the buffer during time unit t. Therefore, at the beginning of time unit $t+1$, the target part will still be the nth part in the buffer (no matter if the upstream machine adds a new part to the buffer or not). Thus $\alpha_2(t) = 0$ and $x(t+1) = n$. More importantly, the waiting time for the target part is $T(N) = \tau$ and it is counted at the beginning of time unit t. However, if it is instead counted at the beginning of time unit $t+1$, then there are $T(N) = \tau - 1$ steps to go since time unit t is passed.

- if M_2 remains up with probability $1-p_2$, then it will process the part in position 1 in the buffer during time unit t. Therefore, at the beginning of time unit $t+1$, the target part will be the $n-1$th part in the buffer (no matter if the upstream machine adds a new part to the buffer or not). Thus $\alpha_2(t) = 1$ and $x(t+1) = n - 1$. As before, the waiting time for the target part is $T(N) = \tau$ and it is counted at the beginning of time unit t. If it is instead counted at the beginning of time unit $t+1$, then $T(N) = \tau - 1$.

With the two possibilities above, we establish the following equation (also see

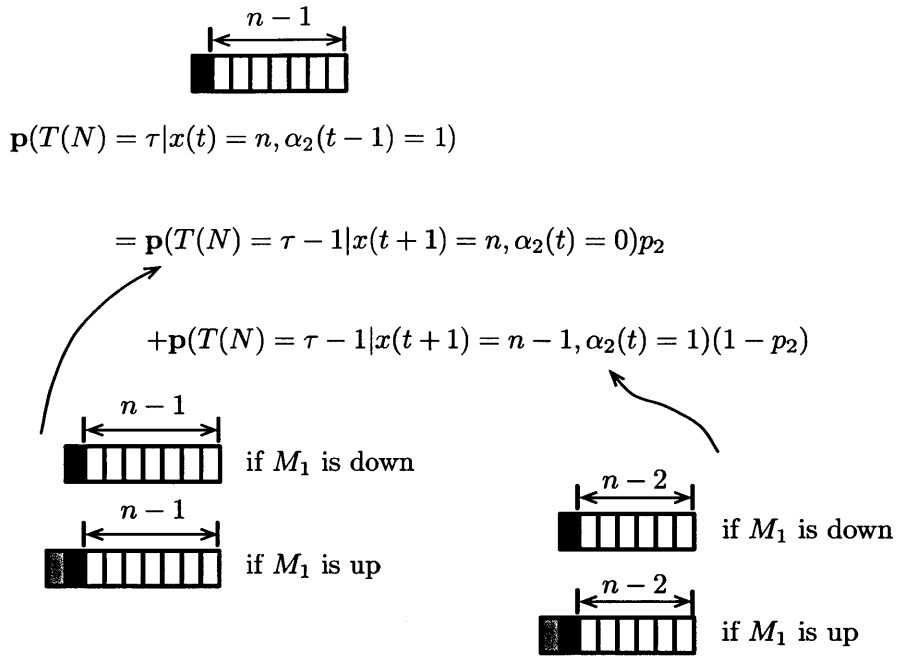


Figure 7-4: Illustration of the transition equation when $x(t) = n, \alpha_2(t-1) = 1$

Figure 7-4),

$$\mathbf{p}(T(N) = \tau | x(t) = n, \alpha_2(t-1) = 1) =$$

$$p_2 \mathbf{p}(T(N) = \tau - 1 | x(t+1) = n, \alpha_2(t) = 0) \tag{7.4}$$

$$+ (1 - p_2)\mathbf{p}(T(N) = \tau - 1 | x(t+1) = n - 1, \alpha_2(t) = 1)$$

or with the convenient notation, we have

$$p_t(\tau, n) = p_2 q_{t+1}(\tau - 1, n) + (1 - p_2) p_{t+1}(\tau - 1, n - 1). \tag{7.5}$$

We can safely disregard the time subscripts in Equation (7.5) because the system is in steady state and therefore these probabilities are independent of the time argument t. Therefore, we take away the time subscripts in Equation (7.5) and get

$$p(\tau, n) = p_2 q(\tau - 1, n) + (1 - p_2) p(\tau - 1, n - 1). \tag{7.6}$$

Next, suppose that $\alpha_2(t-1) = 0$ when the target part enters the buffer at the beginning of time unit t and before the state of M_2 gets updated, then

- if M_2 gets repaired with probability r_2, then it will process the part in position 1 in the buffer during time unit t. Therefore, at the beginning of time unit $t+1$, the target part will be the $n-1$th part in the buffer (no matter if the upstream machine adds a new part to the buffer or not). Thus $\alpha_2(t) = 1$ and $x(t+1) = n-1$.

- if M_2 remains down with probability $1-r_2$, then it cannot process any part in the buffer during time unit t. Therefore, at the beginning of time unit $t+1$, the target part will still be the nth part in the buffer (no matter if the upstream machine adds a new part to the buffer or not). Thus $\alpha_2(t) = 0$ and $x(t+1) = n$.

Similarly, we establish the following equation (after dropping the time subscripts),

$$q(\tau, n) = r_2 p(\tau - 1, n-1) + (1-r_2) q(\tau - 1, n). \tag{7.7}$$

Equations (7.6) and (7.7) are the two basic recursive equations of the iteration approach. When $n = 1$, these two equations are simplified to

$$p(\tau, 1) = p_2 q(\tau - 1, 1), \tag{7.8}$$

and

$$q(\tau, 1) = (1-r_2) q(\tau - 1, 1). \tag{7.9}$$

In addition, the two initial conditions of the iteration approach are $p(1,1) = 1-p_2$ and $q(1,1) = r_2$. Recall that $p(\tau, n) = q(\tau, n) = 0, \forall \tau < n$, then according to the analysis above, we summarize the expressions of non-zero $p(\tau, n)$ and $p(\tau, n)$ below:

- $\tau = 1$

$$p(1,1) \;=\; 1-p_2,$$

$$q(1,1) = r_2,$$

- $2 \leq \tau \leq N$

$$p(\tau,1) = p_2 q(\tau-1,1),$$

$$q(\tau,1) = (1-r_2)q(\tau-1,1),$$

$$p(\tau,n) = p_2 q(\tau-1,n) + (1-p_2)p(\tau-1,n-1), \quad 2 \leq n \leq \tau-1,$$

$$q(\tau,n) = r_2 p(\tau-1,n-1) + (1-r_2)q(\tau-1,n), \quad 2 \leq n \leq \tau-1,$$

$$p(\tau,n) = (1-p_2)p(\tau-1,n-1), \quad n = \tau,$$

$$q(\tau,n) = r_2 p(\tau-1,n-1), \quad n = \tau,$$

- $\tau > N$

$$p(\tau,1) = p_2 q(\tau-1,1),$$

$$q(\tau,1) = (1-r_2)q(\tau-1,1),$$

$$p(\tau,n) = p_2 q(\tau-1,n) + (1-p_2)p(\tau-1,n-1), \quad 2 \leq n \leq N,$$

$$q(\tau,n) = r_2 p(\tau-1,n-1) + (1-r_2)q(\tau-1,n), \quad 2 \leq n \leq N.$$

With the iteration approach above, we are able to find $\mathbf{p}(T(N) = \tau | x(t) = n, \alpha_2(t-1) = 1)$ and $\mathbf{p}(T(N) = \tau | x(t) = n, \alpha_2(t-1) = 0)$. Next, we explain

how to find $\mathbf{p}(x(t) = n, \alpha_2(t-1) = 1)$ and $\mathbf{p}(x(t) = n, \alpha_2(t-1) = 0), 1 \leq n \leq N$ from the analytical solution of two-machine line evaluation of Gershwin (1994). The analytical solution specifies the steady state probabilities $\mathbf{p}(n, \alpha_1, \alpha_2)$ of the line in different states in terms of buffer level n as well as the status of both machines α_1 and α_2.

We have mentioned that although both $\mathbf{p}(x(t) = n, \alpha_2(t-1) = 1)$ and $\mathbf{p}(x(t) = n, \alpha_2(t-1) = 0), 1 \leq n \leq N$ look like unconditional probabilities, they imply an underlying condition that there is indeed a new part entering the buffer at the beginning of time unit t. As a result, $\mathbf{p}(x(t) = n, \alpha_2(t-1) = 1)$, for instance, can be depicted as

$$\mathbf{p}(x(t) = n, \alpha_2(t-1) = 1) = \frac{\mathbf{p}\left(\begin{array}{l}\text{a new part enters the buffer} \\ \text{at the beginning of time unit } t, \\ \text{its position is } n, \text{ and} \\ M_2 \text{ is up before it gets updated}\end{array}\right)}{\mathbf{p}\left(\begin{array}{l}\text{a new part enters the buffer} \\ \text{at the beginning of time unit } t\end{array}\right)}.$$

For the rest of the derivation, let $\mathbf{p}^u(x(t) = n, \alpha_2(t-1) = 1)$ be the corresponding unconditional probability of $\mathbf{p}(x(t) = n, \alpha_2(t-1) = 1)$. In other words, we have

$$\mathbf{p}^u(x(t) = n, \alpha_2(t-1) = 1) = \mathbf{p}\left(\begin{array}{l}\text{a new part enters the buffer} \\ \text{at the beginning of time unit } t, \\ \text{its position is } n, \text{ and} \\ M_2 \text{ is up before it gets updated}\end{array}\right), \qquad (7.10)$$

and similarly,

$$\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=0)=\mathbf{p}\left(\begin{array}{l}\text{a new part enters the buffer}\\ \text{at the beginning of time unit } t,\\ \text{its position is } n, \text{ and}\\ M_2 \text{ is down before it gets updated}\end{array}\right). \quad (7.11)$$

Therefore, $\mathbf{p}(x(t)=n,\alpha_2(t-1)=1)$ can be expressed as

$$\mathbf{p}(x(t)=n,\alpha_2(t-1)=1)=$$

$$\frac{\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=1)}{\sum_{j=1}^{N}\left(\mathbf{p}^u(x(t)=j,\alpha_2(t-1)=1)+\mathbf{p}^u(x(t)=j,\alpha_2(t-1)=0)\right)} \quad (7.12)$$

where the denominator is the unconditional probability that a new part enters the buffer at the beginning of time unit t. Similarly, $\mathbf{p}(x(t)=n,\alpha_2(t-1)=0)$ is

$$\mathbf{p}(x(t)=n,\alpha_2(t-1)=0)=$$

$$\frac{\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=0)}{\sum_{j=1}^{N}\left(\mathbf{p}^u(x(t)=j,\alpha_2(t-1)=1)+\mathbf{p}^u(x(t)=j,\alpha_2(t-1)=0)\right)}. \quad (7.13)$$

Next, we show how to find $\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=1)$ and $\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=0)$ with the steady state probabilities $\mathbf{p}(n,\alpha_1,\alpha_2)$ of a two-machine line (see Appendix A for those probabilities). Let us consider $\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=1)$ where $2\leq n\leq N-1$ first. We analyze the possible states in which the system has to be in at the beginning of time unit $t-1$ such that a new part will enter the buffer at the beginning of time unit t, the position of the new part is n once it enters the buffer, and M_2 is up before its state is updated at beginning of time unit t. To make this to happen, the following scenarios are possible:

- the system is in state $(n,0,0)$ at the beginning of time unit $t-1$, and both the

upstream and the downstream machines are repaired with probabilities r_1 and r_2, respectively, during time unit $t-1$.

- the system is in state $(n,0,1)$ at the beginning of time unit $t-1$, the upstream machine gets repaired with probability r_1, and the downstream machine does not fail with probability $1-p_2$ during time unit $t-1$.

- the system is in state $(n,1,0)$ at the beginning of time unit $t-1$, the upstream machine does not fail with probability $1-p_1$, and the downstream machine gets repaired with probability r_2 during time unit $t-1$.

- the system is in state $(n,1,1)$ at the beginning of time unit $t-1$, and both the upstream and the downstream machine do not fail with probability $1-p_1$ and $1-p_2$, respectively, during time unit $t-1$.

In all the four cases above, the upstream machine will add a part to the buffer and the downstream machine will remove a part from the buffer at the end of time unit $t-1$, or equivalently the beginning of time unit t. That is to say that the new part from the upstream machine will enter the buffer and the downstream machine is up at the beginning of time unit t before its state is updated again for time unit t. In addition, since the upstream machine adds a part while the downstream machine removes a part, the buffer level remains to be n and therefore the position of the new part is n. According to the analysis above, $\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=1), 2\leq n\leq N-1$, can be computed by

$$\begin{aligned}\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=1) &= r_1r_2\mathbf{p}(n,0,0)+r_1(1-p_2)\mathbf{p}(n,0,1)\\ &\quad +(1-p_1)r_2\mathbf{p}(n,1,0)+(1-p_1)(1-p_2)\mathbf{p}(n,1,1).\end{aligned} \tag{7.14}$$

Next we consider $\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=1)$ where $n=1$. To make this to happen, the following scenarios are possible:

- the system is in state $(1,0,0)$ at the beginning of time unit $t-1$, and both the upstream and the downstream machines are repaired with probabilities r_1 and

r_2, respectively, during time unit $t-1$.

- the system is in state $(1,0,1)$ at the beginning of time unit $t-1$, the upstream machine gets repaired with probability r_1, and the downstream machine does not fail with probability $1-p_2$ during time unit $t-1$.

- the system is in state $(1,1,1)$ at the beginning of time unit $t-1$, and both the upstream and the downstream machine do not fail with probability $1-p_1$ and $1-p_2$, respectively, during time unit $t-1$.

- the system is in state $(0,0,1)$ at the beginning of time unit $t-1$ and the upstream machine gets repaired with probability r_1. Note that due to the convention of the two-machine line model, the downstream machine will be starved during the entire time unit $t-1$.

In the first three cases above, the upstream machine will add a part to the buffer and the downstream machine will remove a part from the buffer at the end of time unit $t-1$, or equivalently the beginning of time unit t. Therefore the new part from the upstream machine will enter the buffer and the downstream machine is up at the beginning of time unit t before its state is determined again. Since the upstream machine adds a part while the downstream machine removes a part, then the buffer level remains to be 1 and therefore the position of the new part is 1. In the four case above, the buffer is empty at the beginning of time unit $t-1$ and the upstream machine adds a new part to the buffer at the beginning of time unit t and therefore the buffer level becomes 1. Therefore, the position of the new part is 1. According to the analysis above, $\mathbf{p}^u(x(t)=1,\alpha_2(t-1)=1)$ can be computed by

$$\begin{aligned}\mathbf{p}^u(x(t)=1,\alpha_2(t-1)=1) &= r_1r_2\mathbf{p}(1,0,0)+r_1(1-p_2)\mathbf{p}(1,0,1)\\ &\quad +(1-p_1)(1-p_2)\mathbf{p}(1,1,1)+r_1\mathbf{p}(0,0,1).\end{aligned} \tag{7.15}$$

Next we consider $\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=1)$ where $n=N$. First, if the buffer level is N, the the upstream machine is blocked and the downstream machine is up

during time unit $t-1$. Therefore, the downstream machine will remove a part from the buffer and the buffer level becomes $N-1$. On the other hand, if the buffer level is $N-1$ during time unit $t-1$, then it will remain at $N-1$ if both machines are up, or it will be $N-2$ if the upstream machine is down while the downstream machine is up during time unit $t-1$. Therefore, the buffer level cannot be N while M_2 is up at the beginning of time unit t. Consequently, $\mathbf{p}^u(x(t)=N,\alpha_2(t-1)=1)=0$. We summarize $\mathbf{p}^u(x(t)=n,\alpha_2(t)=1)$, $1\leq n\leq N$ as follows

$$\begin{aligned}
\mathbf{p}^u(x(t)=1,\alpha_2(t-1)=1) &= r_1r_2\mathbf{p}(1,0,0)+r_1(1-p_2)\mathbf{p}(1,0,1)\\
&\quad +(1-p_1)(1-p_2)\mathbf{p}(1,1,1)+r_1\mathbf{p}(0,0,1),\\
\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=1) &= r_1r_2\mathbf{p}(n,0,0)+r_1(1-p_2)\mathbf{p}(n,0,1)\\
&\quad +(1-p_1)r_2\mathbf{p}(n,1,0)+(1-p_1)(1-p_2)\mathbf{p}(n,1,1),\\
&\quad 2\leq n\leq N-1,\\
\mathbf{p}^u(x(t)=N,\alpha_2(t-1)=1) &= 0.
\end{aligned}$$

We analyze $\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=0)$, $1\leq n\leq N$ similarly and they are summarized as follows

$$\begin{aligned}
\mathbf{p}^u(x(t)=1,\alpha_2(t-1)=0) &= 0,\\
\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=0) &= r_1(1-r_2)\mathbf{p}(n-1,0,0)+r_1p_2\mathbf{p}(n-1,0,1)\\
&\quad +(1-p_1)(1-r_2)\mathbf{p}(n-1,1,0)\\
&\quad +(1-p_1)p_2\mathbf{p}(n-1,1,1),\quad 2\leq n\leq N-1,
\end{aligned}$$

$$\begin{aligned}\mathbf{p}^u(x(t)=N,\alpha_2(t-1)=0) &= r_1(1-r_2)\mathbf{p}(N-1,0,0)\\ &\quad +(1-p_1)(1-r_2)\mathbf{p}(N-1,1,0)\\ &\quad +(1-p_1)p_2\mathbf{p}(N-1,1,1).\end{aligned}$$

After finding $\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=1)$ and $\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=0)$, we are able to calculate $\mathbf{p}(x(t)=n,\alpha_2(t-1)=0)$ and $\mathbf{p}(x(t)=n,\alpha_2(t-1)=0)$ according to (7.12) and (7.13). Finally, we compute $\mathbf{p}(T(N)=\tau)$ according to (7.3),

$$\begin{aligned}&\mathbf{p}(T(N)=\tau)\\ &=\sum_{n=1}^{N}\Big[\mathbf{p}(T(N)=\tau|x(t)=n,\alpha_2(t-1)=1)\mathbf{p}(x(t)=n,\alpha_2(t-1)=1)\\ &\quad+\mathbf{p}(T(N)=\tau|x(t)=n,\alpha_2(t-1)=0)\mathbf{p}(x(t)=n,\alpha_2(t-1)=0)\Big]\\ &=\sum_{n=1}^{N}\left[p(\tau,n)\frac{\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=1)}{\sum_{j=1}^{N}\Big(\mathbf{p}^u(x(t)=j,\alpha_2(t-1)=1)+\mathbf{p}^u(x(t)=j,\alpha_2(t-1)=0)\Big)}\right.\\ &\quad\left.+\,q(\tau,n)\frac{\mathbf{p}^u(x(t)=n,\alpha_2(t-1)=0)}{\sum_{j=1}^{N}\Big(\mathbf{p}^u(x(t)=j,\alpha_2(t-1)=1)+\mathbf{p}^u(x(t)=j,\alpha_2(t-1)=0)\Big)}\right].\end{aligned}\tag{7.16}$$

To summarize, for any given positive integer τ, we solve $p(\tau,n)$ and $q(\tau,n)$, $1\leq n\leq N$, by iteration. Then we use (7.16) to calculate $\mathbf{p}(T(N)=\tau)$. (The derivation of the part waiting time distribution discussed in this section is also summarized in Shi and Gershwin 2011a.)

7.2.2 Test with Little's Law

In this section, we verify our calculation of $\mathbf{p}(T(N) = \tau)$ by applying Little's Law (Little 1961) to the buffer:

$$L_b = \lambda w_b$$

where L_b is the average number of parts in the buffer, λ is the arrival rate, and w_b is the average waiting time in the buffer. In our notation, the average inventory level is $\bar{n}$, and the arrival rate is the production rate of the line, denoted by $P(N)$. If we denote $\mathbf{E}[T(N)]$ as the average waiting time, then according to Little's Law,

$$\mathbf{E}[T(N)] = \frac{\bar{n}}{P(N)} \tag{7.17}$$

where $\mathbf{E}[T(N)]$ can be computed with the PMF of $T(N)$ as

$$\mathbf{E}[T(N)] = \sum_{\tau=1}^{\infty} \tau \mathbf{p}(T(N) = \tau). \tag{7.18}$$

Consider the five examples shown in Table 7.1, in which we use the analytical solutions of two-machine lines to calculate $\bar{n}$ and $P(N)$, and (7.17) to compute $\mathbf{E}[T(N)]$. The numerical solution for the PMF of $T(N)$ is verified by Little's Law.

Table 7.1: Test with Little's Law

Case	1	2	3	4	5
r_1	.1	.1	.2	.1	.5
p_1	.01	.01	.01	.04	.04
r_2	.1	.1	.1	.2	.4
p_2	.01	.01	.04	.01	.04
N	20.00	50.00	20.00	20.00	20.00
$P(N)$	.870541	.887845	.713445	.713445	.904528
$\bar{n}$	10.000000	25.000000	17.974264	2.025736	12.472901
$\bar{n}/P(N)$	11.487113	28.158078	25.193633	2.839374	13.789396
$\mathbf{E}[T(N)]$	11.487113	28.158078	25.193633	2.839374	13.789396

7.2.3 Comparison with Simulation

We provide numerical experiments to show the accuracy of the numerical solution for $\mathbf{p}(T(N) = \tau)$. To justify the correctness of the numerical solution, we compare it with results from a discrete time simulation that is written for this purpose. In all the experiments below, the length of each simulation is 21,000,000 time units with the first 1,000,000 time units being the warm up period, and we run the simulation 30 times and use the average as the simulation result.

Experiment 1

In the first experiment, the parameters of the line are $p_1 = .01$, $r_1 = .1$, $p_2 = .01$, $r_2 = .1$, and $N = 20$. The numerical results and the simulation results are shown in Figure 7-5. The horizontal axis is the waiting time τ. The vertical axis is $\mathbf{p}(T(N) = \tau)$. It can be seen that the numerical results and the simulation results are highly consistent. In addition, according to the Central Limit Theory (Bertsekas and Tsitsiklis 2008), the average of $\mathbf{p}(T(N) = \tau), \forall \tau$ derived from simulation follows a normal distribution. Since we run the simulation for 30 times, we can compute the confidence interval for each $\mathbf{p}(T(N) = \tau)$ derived from simulation[2]. The detailed simulation results including the mean, the standard deviation, and the 95% confidence interval for each $\mathbf{p}(T(N) = \tau)$, as well as the numerical results are provided in Table 7.2. The value of $\mathbf{p}(T(N) = \tau)$ computed by the numerical solution for each of those 30 values of τ is within the corresponding 95% confidence interval.

Observe that $\mathbf{p}(\tau = 0)$ and $\mathbf{p}(\tau = 19)$ are much bigger than the others. In addition, there is a small tail indicating that there are a small portion of parts whose waiting times are longer than 19 time units. We know that Machines M_1 and M_2 are identical and that the size of the buffer is 20, which is not large. Since the size of the buffer is only 2MTTR, once a machine fails, the buffer tends to be full or empty frequently and the other machine is forced to be idle. Therefore, if the upstream machine fails

[2]Note that since the underlying true standard deviation of $\mathbf{p}(T(N) = \tau)$ for each τ is unknown, the sample standard deviation and the t-distribution (Albright et al. 2009) are used to compute the confidence interval.

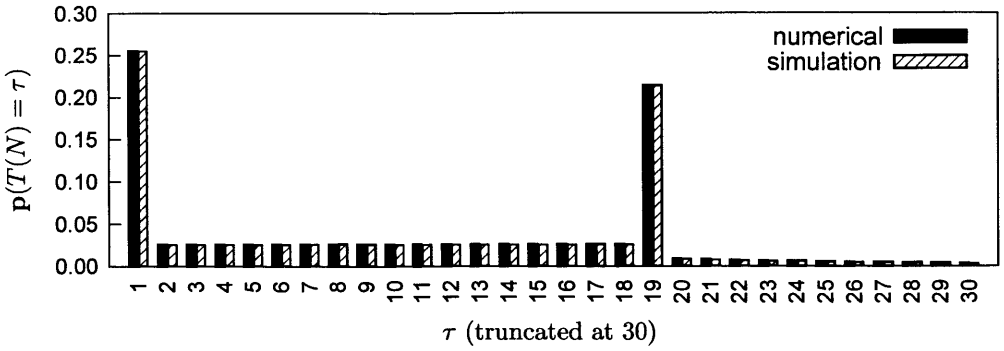


Figure 7-5: PMF of $T(N)$, numerical solution vs. simulation, Experiment 1

and the buffer becomes empty, then after the upstream machine gets repaired the system will run with an inventory level of 1 before the next failure happens. In other words, during this period, parts will spend 1 time unit in the buffer — each part that enters the buffer at the beginning of a time unit will be processed by M_2 immediately and will leave the system at the end of that time unit. On the other hand, if the downstream machine fails and the buffer gets full, then after the downstream machine gets repaired, the system will run with an inventory level of 19 (not 20, due to the convention of the model) before the next failure takes place. In other words, during this period, each new part will be the 19th part in the buffer and will therefore have to wait for 19 time units (if no failures of M_2 happen) in the buffer before it can leave. If Machine M_2 fails again when the system running with an inventory level of 19, the waiting time for those parts in the buffer will be longer than 19. In conclusion, the small buffer makes it most likely for the system to run with either an empty buffer or a full buffer. When the system is run with an inventory level of 1, parts' waiting times are 1; when the system is run with an inventory level of 19, parts' waiting time are 19 or longer.

Also, from the probability distribution of the two-machine one-buffer line, $\mathbf{p}(n = 1)$ and $\mathbf{p}(n = 19)$ are much larger (see Figure 7-6). We should expect a relationship between the PMF of the inventory level n ($\mathbf{p}(n)$) and the PMF of part waiting time $T(N)$. In Figure 7-6, $n = 0$ means that the buffer is empty and the downstream machine is starved, while $n = 20(= N)$ means that the buffer is full and the upstream

Table 7.2: Comparison between numerical and simulation results, Experiment 1

τ	numerical	sim-average	sim-stdev	95% confidence interval
1	0.255155	0.255209	0.001296	[0.254725, 0.255693]
2	0.025773	0.025665	0.000389	[0.025520, 0.025811]
3	0.025773	0.025774	0.000384	[0.025631, 0.025917]
4	0.025773	0.025728	0.000372	[0.025589, 0.025867]
5	0.025773	0.025692	0.000327	[0.025570, 0.025815]
6	0.025773	0.025671	0.000362	[0.025536, 0.025807]
7	0.025773	0.025847	0.000380	[0.025705, 0.025989]
8	0.025773	0.025843	0.000430	[0.025683, 0.026004]
9	0.025773	0.025776	0.000422	[0.025619, 0.025934]
10	0.025773	0.025681	0.000449	[0.025514, 0.025849]
11	0.025773	0.025808	0.000339	[0.025681, 0.025934]
12	0.025773	0.025763	0.000381	[0.025621, 0.025905]
13	0.025773	0.025764	0.000289	[0.025656, 0.025871]
14	0.025773	0.025782	0.000467	[0.025607, 0.025956]
15	0.025773	0.025762	0.000425	[0.025603, 0.025921]
16	0.025773	0.025785	0.000478	[0.025607, 0.025964]
17	0.025773	0.025801	0.000429	[0.025641, 0.025961]
18	0.025773	0.025738	0.000353	[0.025606, 0.025870]
19	0.213386	0.213478	0.001051	[0.213086, 0.213871]
20	0.008483	0.008491	0.000090	[0.008457, 0.008525]
21	0.007715	0.007743	0.000101	[0.007705, 0.007780]
22	0.007016	0.007032	0.000067	[0.007007, 0.007057]
23	0.006380	0.006366	0.000090	[0.006332, 0.006399]
24	0.005801	0.005802	0.000086	[0.005770, 0.005835]
25	0.005275	0.005255	0.000088	[0.005222, 0.005287]
26	0.004796	0.004779	0.000056	[0.004758, 0.004800]
27	0.004360	0.004386	0.000074	[0.004359, 0.004414]
28	0.003964	0.003975	0.000065	[0.003951, 0.003999]
29	0.003604	0.003607	0.000059	[0.003585, 0.003629]
30	0.003277	0.003279	0.000059	[0.003257, 0.003301]

machine is blocked. It can be seen that $\mathbf{p}(n = 1)$ and $\mathbf{p}(n = 19)$ are much larger than others. This indicates that since the buffer is not very big, the buffer alternates between empty and full. Therefore, the probabilities of $T(N) = 1$ and $T(N) = 19$ will be large, which is consistent with Figure 7-5. This is why we should expect to observe larger $\mathbf{p}(T(N) = 1)$ and $\mathbf{p}(T(N) = 19)$ as well as a small tail of long waiting times. Figure 7-5 also indicates the values of $\mathbf{p}(T(N) = \tau)$ are the same for τ between

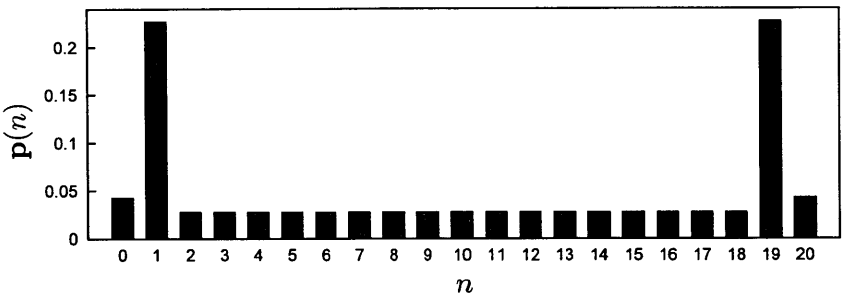


Figure 7-6: $\mathbf{p}(n)$, two-machine line, Experiment 1

2 and 18 time units, but much less than those two larger ones.

Based on the analysis above, if we reduce the MTTR of both machines or increase the size of the buffer, it will be harder for the system to reach an empty or a full buffer. As a result, the probability of $T(N) = 0$ and the probability of $T(N) = N - 1$ time units will be smaller. This is verified by Figure 7-7, where the buffer size is still 20 but parameters of both machines are changed to $r_1 = r_2 = .6$ and $p_1 = p_2 = .04$. With a smaller MTTR for both machines, $\mathbf{p}(T(N) = 0)$ and $\mathbf{p}(T(N) = 19)$ are smaller, since the system spends more time between $n = 1$ and $n = 19$.

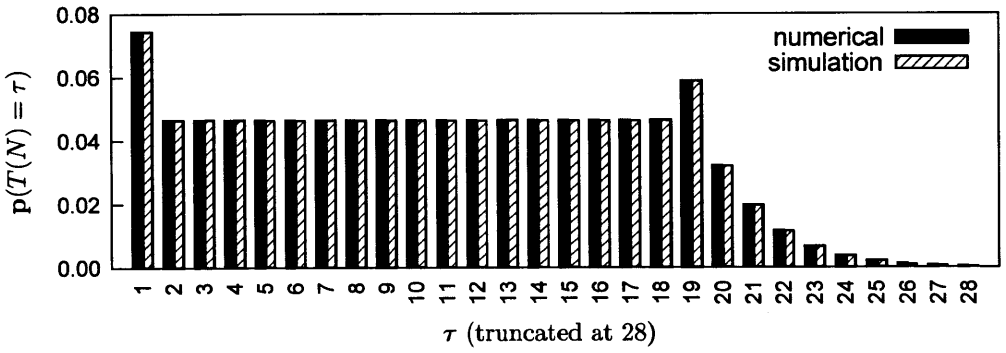


Figure 7-7: PMF of $T(N)$, numerical solution vs. simulation, Experiment 1, modified

Experiment 2

In the second experiment, the parameters of the line are $p_1 = .02$, $r_1 = .2$, $p_2 = .03$, $r_2 = .2$, and $N = 16$, and M_2 is the bottleneck. The results are shown in Figure

7-8. The numerical results and simulation results are consistent. Since the isolated efficiency of M_2 ($e_2 = r_2/(r_2 + p_2) = .870$) is smaller than the isolated efficiency of M_1 ($e_1 = r_1/(r_1 + p_1) = .909$), the average inventory level is high and most of the parts have large waiting times.

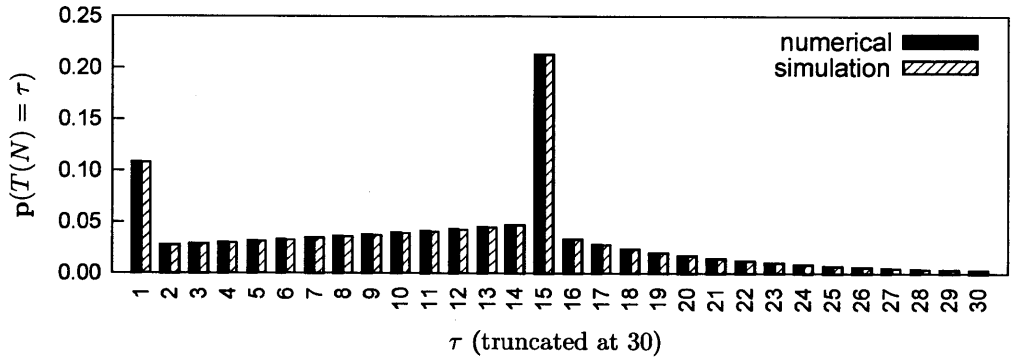


Figure 7-8: PMF of $T(N)$, numerical solution vs. simulation, Experiment 2

Experiment 3

In the third experiment, we consider another example where M_2 is the bottleneck of the line. The parameters of the line are $p_1 = .04$, $r_1 = .5$, $p_2 = .04$, $r_2 = .4$, and $N = 20$. The results are shown in Figure 7-9. The numerical results and simulation results are consistent.

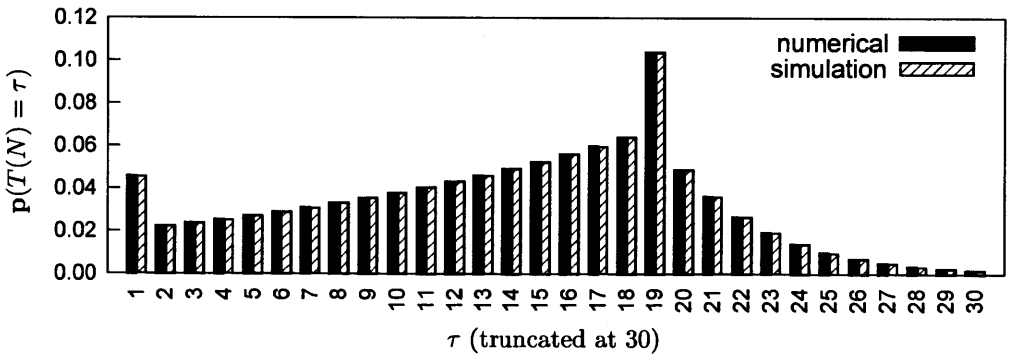


Figure 7-9: PMF of $T(N)$, numerical solution vs. simulation, Experiment 3

Experiment 4

In the last experiment, we consider an example where M_1 is the bottleneck of the line. The parameters of the line are $p_1 = .05$, $r_1 = .3$, $p_2 = .05$, $r_2 = .5$, and $N = 30$. The results are shown in Figure 7-10. In this case, the isolated efficiency of M_2 ($e_2 = r_2/(r_2 + p_2) = .909$) is larger than the isolated efficiency of M_1 ($e_1 = r_1/(r_1 + p_1) = .857$), the average inventory level is low and most of the parts have small waiting times. These experiments studied in this section demonstrate that the numerical solutions are accurate.

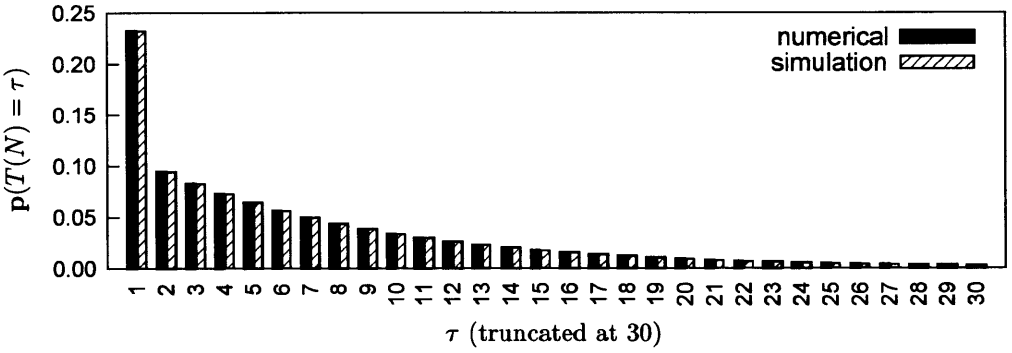


Figure 7-10: PMF of $T(N)$, numerical solution vs. simulation, Experiment 4

7.2.4 Part Waiting Time in Long Lines

The problem addressed in this chapter deals with the part time waiting time in a given buffer $B_{\hat{i}}$ of a long line. In Section 7.2.1, we have derived the analytical solution of the part waiting time distribution for two-machine lines. For long lines, the decomposition method (Gershwin 1987a) is adopted to evaluate the production rate as well as the average inventory level of each buffer.

For a k-machine $k-1$-buffer line, the decomposition analyzes $k-1$ two-machine one-buffer building blocks. For each building block i, two pseudo-machines with geometric repair and failure probabilities are used to approximate the material behavior in Buffer B_i of the original line. Specifically, the parameters of the two pseudo-machines include $r^u(i), p^u(i), r^d(i)$ and $p^d(i)$, where $r^u(i)$ and $p^u(i)$ are parameters of

the upstream pseudo-machine of Buffer B_i, while $r^d(i)$ and $p^d(i)$ are parameters of the downstream pseudo-machine of Buffer B_i. With these parameters, we can apply the analytical formulation (7.16) to compute the part waiting time for any given buffer B_i. This provides us an approximation of the part waiting time distribution in the given buffer of a long line. We provide two long line experiments here.

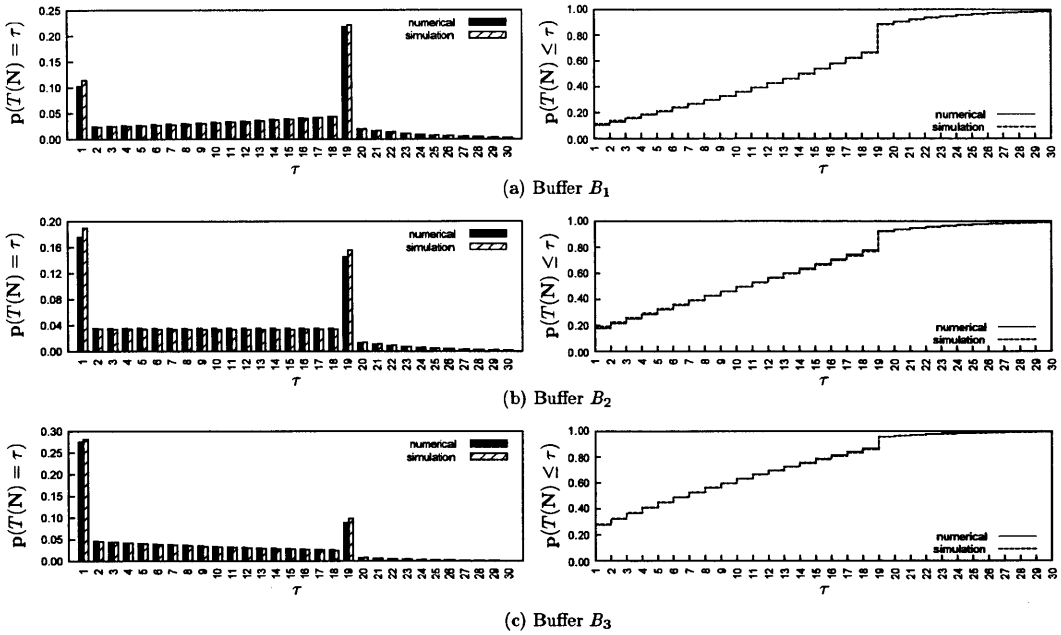
Consider a balanced four-machine three-buffer line first. Machine parameters are $r_i = .2$ and $p_i = .01, i = 1, 2, 3$, and 4. The size of each buffer is 20. To compute approximations of the part waiting time distributions for the three buffers in the line, we first evaluate the line with decomposition and derive the pseudo-machine parameters of all building blocks. These parameters are listed in Table 7.3.

Table 7.3: Pseudo-machine parameters, Experiment 1

building block	$r^u(i)$	$p^u(i)$	$r^d(i)$	$p^d(i)$
1	.200000	.010000	.200000	.013875
2	.200000	.012178	.200000	.012178
3	.200000	.013875	.200000	.010000

Applying the analytical formulation (7.16) to these pseudo-machine parameters allows us to find an approximation of the part waiting time distributions in the three buffers of the original four-machine three-buffer line. On the other hand, a discrete time simulation, which is able to compute the part waiting time distributions of all buffers in a given long line, is written to compare with the numerical solution. The length of each simulation is 21,000,000 time units with the first 1,000,000 time units being the warm up period. We run the simulation 20 times and use the average as the simulation result.

The results of $\mathbf{p}(T(\mathbf{N}))$ for all three buffers are illustrated in Figure 7-11. It can be seen that, despite the small discrepancy between the numerical solution and the simulation solution, the numerical solution is a very good approximation of the part waiting time distribution of a buffer in a long line. In addition to the probability mass function, the cumulative distribution function (CDF) $\mathbf{p}(T(\mathbf{N}) \leq \tau)$ is also considered. $\mathbf{p}(T(\mathbf{N}) \leq \tau)$ from the analytical approach is very close to that from the



(a) Buffer B_1

(b) Buffer B_2

(c) Buffer B_3

Figure 7-11: $\mathbf{p}(T(\mathbf{N}))$ in long lines, Experiment 1

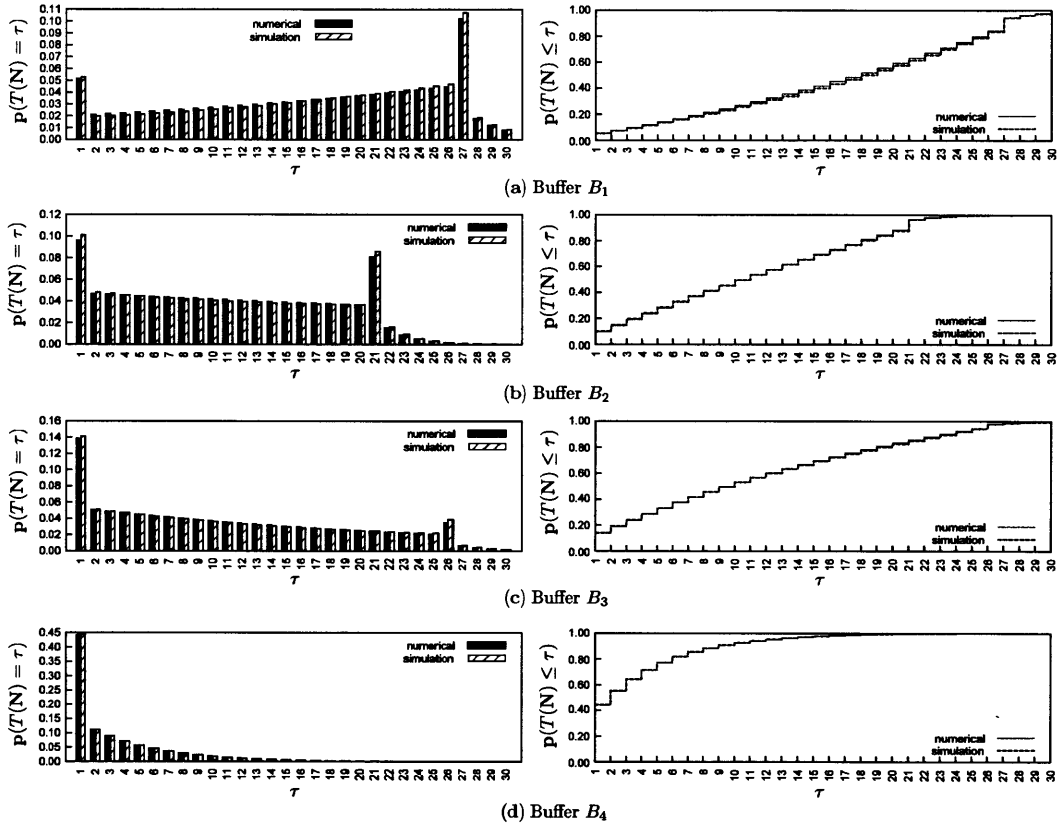
simulation for every τ. When $\mathbf{p}(T(\mathbf{N}) \leq \tau)$ approaches 1, its values from the two approaches are almost identical. It is important to point out that, for the maximum part waiting time constraint, we only care about the cumulative probability $\mathbf{p}(T(\mathbf{N}) \leq W_{\hat{i}})$. This numerical experiment indicates that the analytical approach, which is based on the analytical formulation (7.16) for two-machine lines and decomposition, is very accurate to find $\mathbf{p}(T(\mathbf{N}) \leq W_{\hat{i}})$ in the optimization problem.

In the second experiment, we consider a five-machine four-buffer line. Machine parameters are $r_1 = .4, p_1 = .01, r_2 = .36, p_2 = .009, r_3 = .4, p_3 = .01, r_4 = .45$, and $p_4 = .006$. The sizes of the four buffers are 28, 22, 27 and 26. We evaluate the line with decomposition and derive the pseudo-machine parameters of all building blocks. These parameters are listed in Table 7.4.

The results of $\mathbf{p}(T(\mathbf{N}))$ for all three buffers are illustrated in Figure 7-12. Again, it can be seen that the numerical solution is a very good approximation of the part waiting time distribution of a buffer in a long line. In addition, the CDF $\mathbf{p}(T(\mathbf{N}) \leq \tau)$ from the analytical approach is very close to that from the simulation for every τ.

Table 7.4: Pseudo-machine parameters, Experiment 2

building block	$r^u(i)$	$p^u(i)$	$r^d(i)$	$p^d(i)$
1	.400000	.010000	.376064	.010692
2	.363134	.009850	.493121	.012710
3	.477262	.013736	.400065	.010015
4	.413981	.012636	.450000	.006000



(a) Buffer B_1

(b) Buffer B_2

(c) Buffer B_3

(d) Buffer B_4

Figure 7-12: $\mathbf{p}(T(\mathbf{N}))$ in long lines, Experiment 2

When $\mathbf{p}(T(\mathbf{N}) \leq \tau)$ approaches **1**, its values from the two approaches are again almost identical. This numerical experiment again indicates that the analytical approach is very accurate to find $\mathbf{p}(T(\mathbf{N}) \leq W_i)$ in the optimization problem.

We further study this issue in Section 7.4, where we verify the accuracy of the optimal buffer distribution derived by our algorithm by comparing it with simulation. In particular, we compare the values of $\mathbf{p}(T(\mathbf{N}) \leq W_i)$ from our algorithm and the

simulation for 200 numerical experiments.

7.3 Transformation of the Original Problem

We have mentioned in Section 7.1 and showed in Section 7.2 that the analytical formulation of the part waiting time distribution for two-machine lines is based on iteration. In other words, we do not have a closed form expression for $\mathbf{p}(T(N))$. Therefore, we cannot deal with the constraint $\mathbf{p}(T(\mathbf{N} \leq W_{\hat{i}})) \geq 1-\alpha$ in Problem (7.2) directly by treating N_i as continuous variables. In order to resolve this concern, we transform (7.2) to a transformed problem where the constraint $\mathbf{p}(T(\mathbf{N} \leq W_{\hat{i}})) \geq 1-\alpha$ is replaced by an average part waiting time constraint. In this section, the transformed problem is solved and its solution is checked against $\mathbf{p}(T(\mathbf{N} \leq W_{\hat{i}})) \geq 1-\alpha$ iteratively. We discuss the transformed problem in the remaining of this section.

7.3.1 The Transformed Problem

In the transformed problem, we consider an average part waiting time constraint derived from Little's Law (Little 1961),

$$\frac{\bar{n}_{\hat{i}}}{P(\mathbf{N})} \leq \delta W_{\hat{i}} \tag{7.19}$$

where $\bar{n}_{\hat{i}}/P(\mathbf{N})$ is the average part waiting time in Buffer $B_{\hat{i}}$ according to Little's Law, and δ is a multiplier. We require the average part waiting time to be upper bounded by $\delta W_{\hat{i}}$. This constraint can be also written as

$$\bar{n}_{\hat{i}} \leq \delta W_{\hat{i}} P(\mathbf{N}). \tag{7.20}$$

Therefore, the transformed problem is

$$\begin{aligned} \max_{\mathbf{N}} \quad & J(\mathbf{N}) = AP(\mathbf{N}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i(\mathbf{N}) \\ \text{subject to} \quad & P(\mathbf{N}) \geq \hat{P} \\ & \bar{n}_{\hat{i}} \leq \delta W_{\hat{i}} P(\mathbf{N}) \\ & N_i \geq N_{\min}, \quad \forall i = 1, \cdots, k-1. \end{aligned} \tag{7.21}$$

The replaced constraint only guarantees the average part waiting time to be upper bounded, while the original constraint requires the waiting times of a required percentage of parts to be bounded. To take this into account, we conduct a one-dimensional search over δ and solve Problem (7.21) and check the following condition for each value of δ considered:

$$\mathbf{p}\left(T(\tilde{\mathbf{N}}(\delta)) \leq W_{\hat{i}}\right) \geq 1 - \alpha$$

where $\tilde{\mathbf{N}}(\delta)$ is the solution of the transformed problem for a given δ. We solve the transformed problem iteratively according to the following steps:

- Step 1: Initialize $\delta = \delta_0$ and solve the transformed problem (7.21)[3]. Let the solution be $\tilde{\mathbf{N}}(\delta)$. If $\mathbf{p}\left(T(\tilde{\mathbf{N}}(\delta)) \leq W_{\hat{i}}\right) > 1-\alpha$, go to Step 2. If $\mathbf{p}\left(T(\tilde{\mathbf{N}}(\delta)) \leq W_{\hat{i}}\right) < 1-\alpha$, go to Step 3. If $\mathbf{p}\left(T(\tilde{\mathbf{N}}(\delta)) \leq W_{\hat{i}}\right) = 1-\alpha$, we are done and $\mathbf{N}^\star = \tilde{\mathbf{N}}(\delta)$.
- Step 2: $\mathbf{p}\left(T(\tilde{\mathbf{N}}(\delta)) \leq W_{\hat{i}}\right) > 1 - \alpha$ indicates that either the part waiting time constraint is inactive or the average waiting time constraint in (7.21) is overly restrictive.
  - In the former case, stop and $\mathbf{N}^\star = \tilde{\mathbf{N}}(\delta)$.
  - In the latter case, $\mathbf{p}\left(T(\tilde{\mathbf{N}}(\delta)) \leq W_{\hat{i}}\right) > 1 - \alpha$ implies that δ_0 is too small. Therefore, we conduct a one-dimensional search over $\delta > \delta_0$ and solve the

[3]The algorithm that solves the transformed problem for a given δ is discussed in detail in Section 7.3.2

transformed problem iteratively until we find the smallest $\mathbf{p}\left(T(\tilde{\mathbf{N}}(\delta)) \leq W_{\hat{i}}\right)$ that is larger than $1-\alpha$. Then $\mathbf{N}^{\star} = \tilde{\mathbf{N}}(\delta)$.

- Step 3: $\mathbf{p}\left(T(\tilde{\mathbf{N}}(\delta)) \leq W_{\hat{i}}\right) < 1-\alpha$ implies that δ_0 is too large. In this case, we conduct the one-dimensional search over $\delta < \delta_0$ and solve the transformed problem iteratively until we find the smallest $\mathbf{p}\left(T(\tilde{\mathbf{N}}(\delta)) \leq W_{\hat{i}}\right)$ that is greater than $1-\alpha$. Then $\mathbf{N}^{\star} = \tilde{\mathbf{N}}(\delta)$.

We use $\mathbf{N}^{\star}$ derived according to the procedure above (where we solve the transformed problem for different δs iteratively) as the optimal solution of the original problem (7.2). Although we do not have a proof that this converges to the solution of Problem (7.2), this method has worked on all the numerical experiments we have studied (see Section 7.4). In the numerical experiments, we compare the results from the procedure above with the results from a surface search method. In the surface search, we use the optimal solution from the procedure above as the center point. We search all the points (within a reasonable range of each N_i) around the center point in the $(N_1, N_2, \cdots, N_{k-1})$ space. For every buffer distribution $\mathbf{N}$ within our search scope, we first check if it satisfies both the production rate constraint (i.e., $P(\mathbf{N} \geq \hat{P})$) and the maximum part waiting time probability constraint[4] (i.e., $\mathbf{p}(T(\mathbf{N}) \leq W_{\hat{i}}) \geq 1-\alpha$). If and only if $\mathbf{N}$ satisfies both constraints, it can be considered as a feasible point. After we find all the feasible points, we compute the profits for all those feasible points and choose the one that gives us the maximum profit as the optimal solution of the surface search method. The description above indicates that the surface search method deals with the original problem directly. The accuracy of the optimal buffer distributions found by solving the transformed problem iteratively is verified by comparing with the surface search method. We want to point out that it is worth studying the original problem directly once we have the closed form expression of $\mathbf{p}(T(\mathbf{N}) \leq W_{\hat{i}})$ and compare the solution with our approach here. We outline this as one of the future research directions in Chapter 10.

In what follows, we discuss how to solve the transformed problem, which has a

[4]Note that this is not the average part waiting time constraint derived from Little's Law.

production rate constraint as well as an average part waiting time constraint, for a given δ.

7.3.2 The Algorithm to Solve the Transformed Problem for a Given δ

In this section, we present the algorithm that solves (7.21) for a given δ. Before doing that, it is helpful and necessary to indicate that this problem has five possible cases. They are listed in Table 7.5.

Table 7.5: Five cases for production rate constraint and average part waiting time constraint

	the production rate constraint	the average part waiting time constraint	feasibility of (7.21)
Case 1	conflict with the other	conflict with the other	infeasible
Case 2	active	active	feasible
Case 3	active	inactive	feasible
Case 4	inactive	active	feasible
Case 5	inactive	inactive	feasible

Consider a three-machine two-buffer line with machine parameters $r_1 = .15, p_1 = .01, r_2 = .15, p_2 = .01, r_3 = .09$ and $p_3 = .01$. In addition, the revenue coefficient $A = \$1500$/part. The cost coefficients $b_i = c_i = \$1$/part/time unit, $i = 1, 2$. Consider the five examples of the cases of Table 7.5:

- Case 1: $\hat{P} = .89$ and $\delta W_1 = 2$.
- Case 2: $\hat{P} = .88$ and $\delta W_1 = 7$.
- Case 3: $\hat{P} = .88$ and $\delta W_1 = 15$.
- Case 4: $\hat{P} = .86$ and $\delta W_1 = 6.5$.
- Case 5: $\hat{P} = .86$ and $\delta W_1 = 15$.

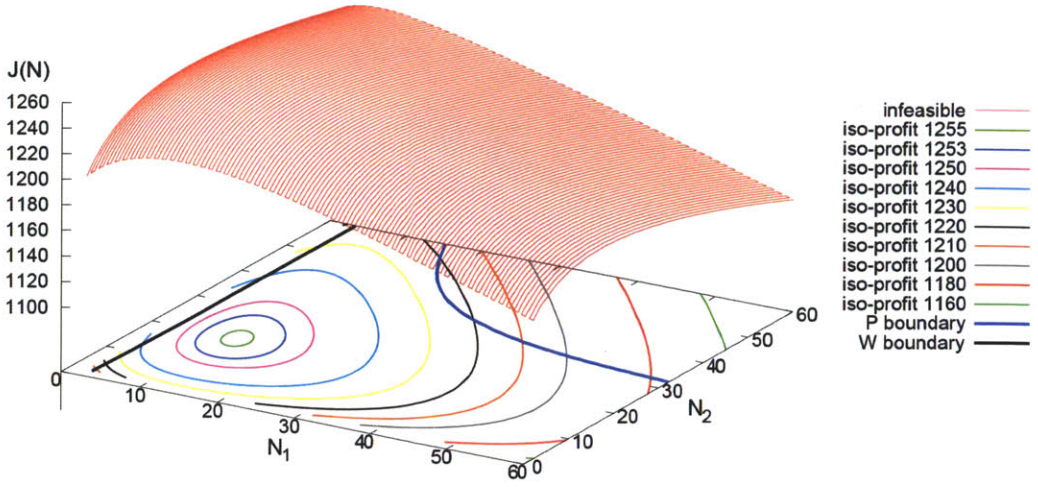


Figure 7-13: Example of the average part waiting time constraint problem, Case 1

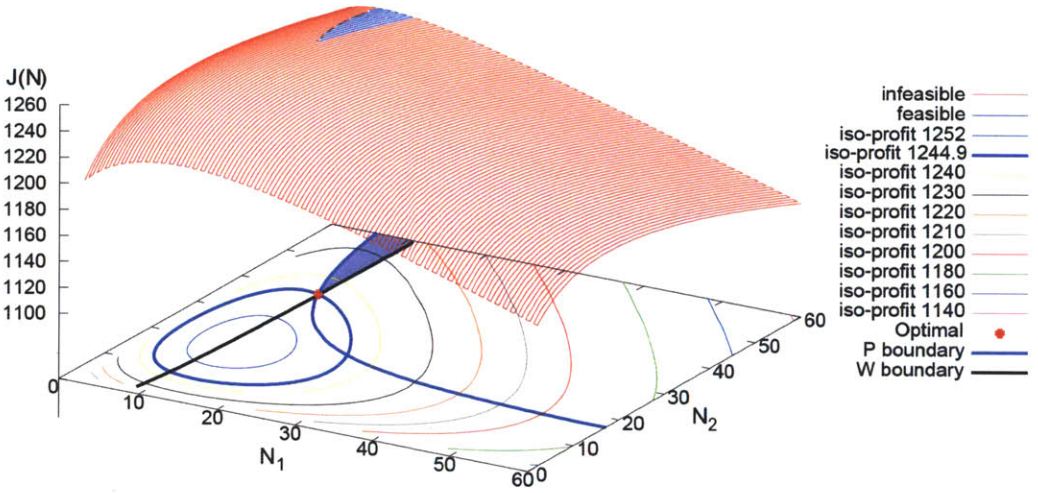


Figure 7-14: Example of the average part waiting time constraint problem, Case 2

Figures 7-13 to 7-17 illustrate these five cases. In all five cases, the profit of the three-machine two-buffer line is plotted as a function of buffer sizes N_1 and N_2. The blue regions on the profit surfaces in Figures 7-14, 7-15, 7-16, and 7-17 are the feasible regions under both the production rate constraint and the average part waiting time constraint. In each of those four cases, the blue region on the profit surface is also projected on $N_1 - N_2$ plane. The red point indicates the optimal solution that maximizes the profit of the line, while satisfying both constraints. It is helpful to indicate that in Case 1, the production rate constraint and the average part waiting time constraint conflict with each other and therefore the problem is

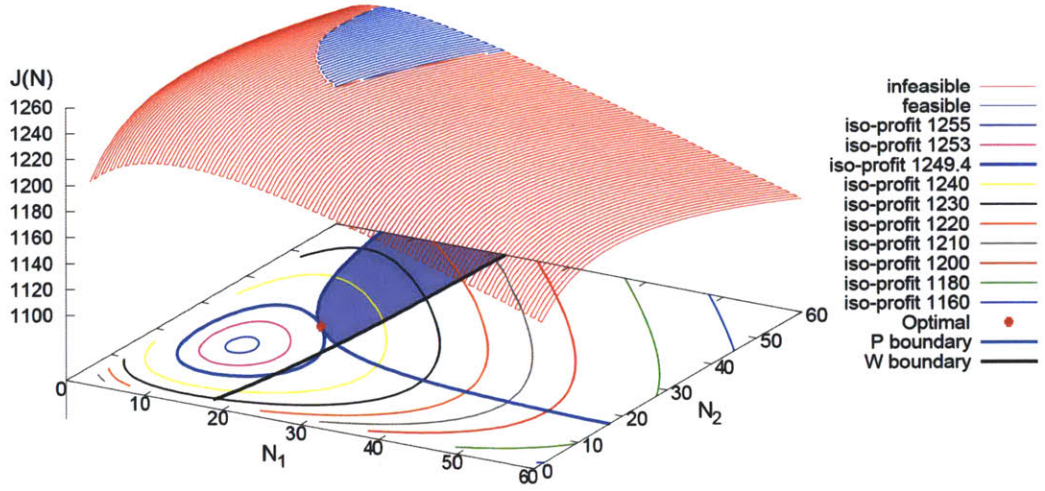


Figure 7-15: Example of the average part waiting time constraint problem, Case 3

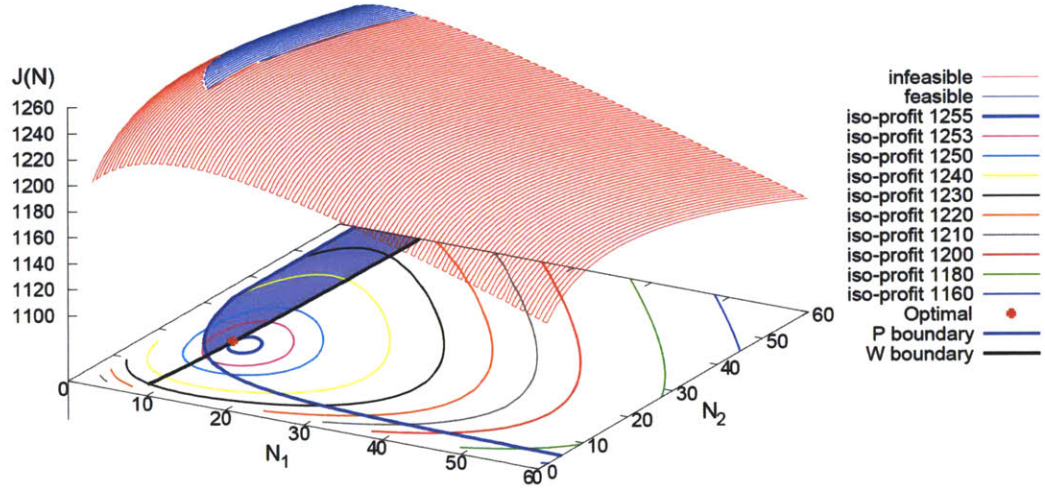


Figure 7-16: Example of the average part waiting time constraint problem, Case 4

infeasible. Therefore, there is no blue region or optimal buffer allocation in this case.

It can be seen that, given the target production rate $\hat{P}$ and the target time constraint $\delta W_{\hat{i}}$, the two constraints may not be both active. If they are not both active, the problem is essentially relaxed to a simpler one with only one constraint (plus those $N_i \geq N_{\min}, i = 1, \cdots, k-1$ constraints). In particular, if the production rate constraint is the only active constraint (i.e., Case 3), the problem is exactly the same as the profit maximization problem studied in Chapter 4. On the other hand, if the average part waiting time constraint is the only active constraint (i.e., Case 4), then an algorithm based on the KKT conditions can be developed

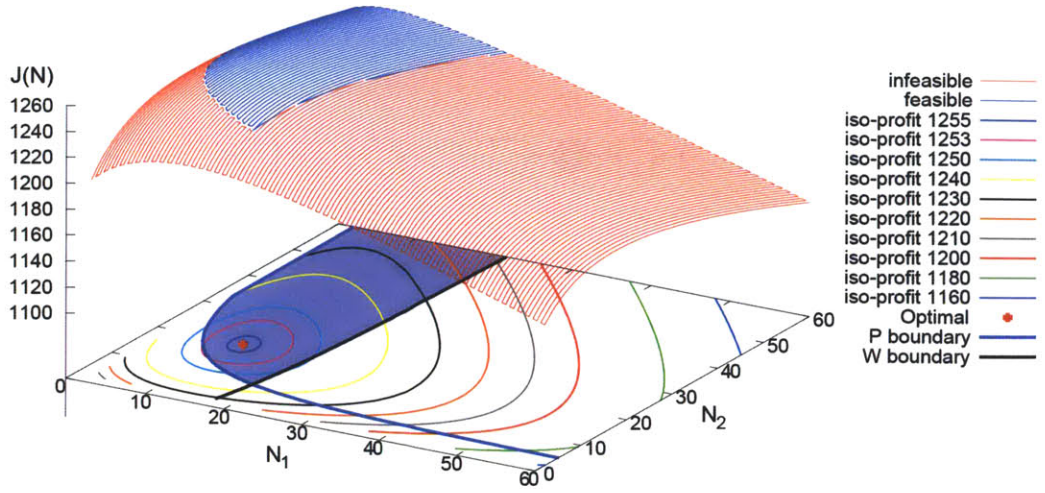


Figure 7-17: Example of the average part waiting time constraint problem, Case 5

to solve the problem. This is because in this case $\nabla\left(\bar{n}_{\hat{i}}(\tilde{\mathbf{N}}_\delta) - \delta W_{\hat{i}} P(\tilde{\mathbf{N}}_\delta)\right)$ and $\nabla\left(\tilde{N}_i(\delta) - N_{\min}\right), \forall i \in B = \left\{i | \tilde{N}_i(\delta) = N_{\min}\right\}$ are linearly independent[5,6]. Therefore, the linear independence constraint qualification guarantees the existence of the Lagrange multipliers (Bertsekas 1999). However, when both constraints are active (i.e., Case 2), the problem becomes harder. In what follows, we discuss how a similar optimization algorithm based on the KKT conditions can be used to solve Case 2 where both the production rate constraint and the average part waiting time constraint are active.

Note that the average part waiting time constraint is nonlinear because both $\bar{n}_{\hat{i}}(\mathbf{N})$ and $P(\mathbf{N})$ are nonlinear functions of the decision variable $\mathbf{N}$. The linear independence constraint qualification of active inequality constraints ensures that there exists Lagrange multipliers for Case 2 to satisfy the KKT conditions (Bertsekas 1999). This is equivalent to requiring that $\nabla\bar{n}_{\hat{i}}(\tilde{\mathbf{N}}(\delta))$ and $\nabla P(\tilde{\mathbf{N}}(\delta))$ are linearly independent[7]. We know that all components of $\nabla P(\tilde{\mathbf{N}}(\delta))$ are positive due to the monotonicity of $P(\mathbf{N})$.

[5]We let $\tilde{\mathbf{N}}(\delta)$ be the optimal solution of the transformed problem for a given δ. Most often, $\tilde{\mathbf{N}}(\delta)$ is an interior solution. Therefore all $N_i \geq N_{\min}$ constraints are inactive and $B = \emptyset$. If $B \neq \emptyset$, we assume that not all optimal buffer sizes satisfy $\tilde{N}_i(\delta) = N_{\min}$, because otherwise the optimal solution is simply $\tilde{N}_1(\delta) = \tilde{N}_2(\delta) = \cdots \tilde{N}_{k-1}(\delta) = N_{\min}$.

[6]Recall that in Chapter 4, we explain that $\mathbf{N}$ can be treated as a vector of continuous variables. Thus, $\bar{n}_{\hat{i}}(\mathbf{N})$ and $P(\mathbf{N})$ can be treated as continuously differentiable functions. Therefore, we can find the corresponding gradients.

[7]We again assume that the optimal solution $\tilde{\mathbf{N}}(\delta)$ is an interior solution and therefore all $N_i \geq N_{\min}$ constraints are inactive. In all our experiments, the optimal solutions have this feature.

Next, we discuss the positivity and negativity of each component in $\nabla \bar{n}_{\hat{i}}(\tilde{\mathbf{N}}(\delta))$.

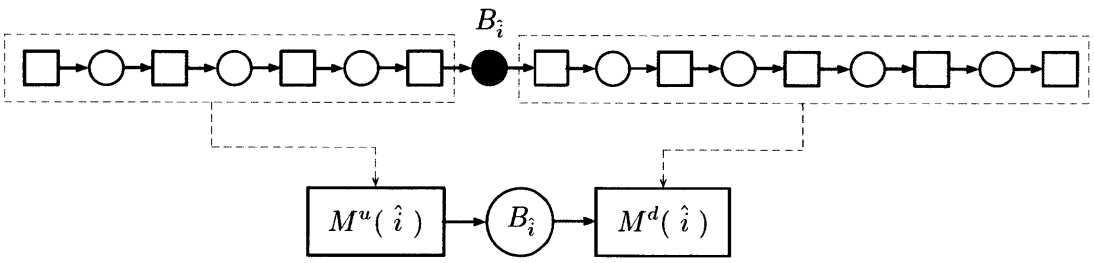


Figure 7-18: Two-machine line representative of a k-machine $k-1$-buffer line

In what follows, we study $\nabla \bar{n}_{\hat{i}}(\mathbf{N})$ in general. The property of $\nabla \bar{n}_{\hat{i}}(\tilde{\mathbf{N}}(\delta))$ can be inferred directly by letting $\mathbf{N} = \tilde{\mathbf{N}}(\delta)$. To study the sign of each component of $\nabla \bar{n}_{\hat{i}}(\mathbf{N})$, we need to understand if $\bar{n}_{\hat{i}}$ increases or decreases as we change the size of another buffer $B_i, \forall i = 1, \cdots, k-1$. To do this, we consider a k-machine $k-1$-buffer line as a two-machine one-buffer line with respect to $B_{\hat{i}}$, since it divides the original line into two segments (see Figure 7-18). In the two-machine one-buffer line $M^u(\hat{i}) - B_{\hat{i}} - M^d(\hat{i})$, the parameters of the two machines are chosen such that the material behavior in $B_{\hat{i}}$ in the two-machine line is the same as that in the original line. We would like to point out that each of the two machines may have more than one failure mode and its repair and failure time probability distributions are not necessarily geometric, since both $M^u(\hat{i})$ and $M^d(\hat{i})$ may represent a set of machines and a set of buffers of the original line. (For this discussion, we are not talking about the decomposition method which approximates a long line with a two-machine one-buffer building block where each machine has a single geometrically distributed failure mode.) For a two-machine one-buffer line where the buffer is finite and $M^u(\hat{i})$ and $M^d(\hat{i})$ are any models of machines, we make the following assumptions for subsequent discussion. These assumptions include:

1. if $M^u(\hat{i})$ or $M^d(\hat{i})$ gets faster, the average production rate of the line increases;

2. if the upstream machine $M^u(\hat{i})$ gets faster, the average inventory level $\bar{n}_{\hat{i}}$ increases;

3. if the downstream machine $M^d(\hat{i})$ gets faster, the average inventory level $\bar{n}_{\hat{i}}$

decreases;

4. and finally, if the size of $B_{\hat{i}}$ increases, the average inventory level $\bar{n}_{\hat{i}}$ increases.

It can be seen that if we change the size of Buffer $B_i, i \neq \hat{i}$, we will change the parameter of either $M^u(\hat{i})$ or $M^d(\hat{i})$. Then, with the assumptions listed above, we will make conclusion on how $\bar{n}_{\hat{i}}$ varies as the size of Buffer $B_i, \forall i$ changes. Therefore, we will be able to identify the sign of $\partial \bar{n}_{\hat{i}}(\mathbf{N})/\partial N_i$ in $\nabla \bar{n}_{\hat{i}}(\mathbf{N})$. Next, we first discuss the signs of $\partial \bar{n}_{\hat{i}}(\mathbf{N})/\partial N_i, \forall \hat{i} = 1, \cdots, k-1$ and $i = 1, \cdots, k-1$. Then we will provide numerical experiments to verify them.

Assume that $\hat{i} = 1$ first. This case is illustrated by a five-machine line in Figure 7-19. We now study the sign of each component of $\nabla \bar{n}_{\hat{i}}(\mathbf{N})$. Instead of considering the original five-machine line, we study the two-machine line $M_1 - B_1 - M^d(1)$. It can be seen that, as we enlarge Buffer B_2, B_3, or B_4, $M^d(1)$ will be faster because Machines M_2, M_3, M_4, and M_5 in the original line will be further decoupled due to a larger B_2, B_3, or B_4. As a result, $M^d(1)$ will pull out parts from Buffer B_1 faster than before. Therefore, the average buffer level $\bar{n}_1$ will decrease due to the enlargement of a downstream buffer. This is equivalent to saying that $\partial \bar{n}_1(\mathbf{N})/\partial N_i < 0, \forall i = 2, \cdots, k-1$. On the other hand, if we increase B_1, $\bar{n}_1(\mathbf{N})$ will increase as well. Thus, $\partial \bar{n}_1(\mathbf{N})/\partial N_1 > 0$.

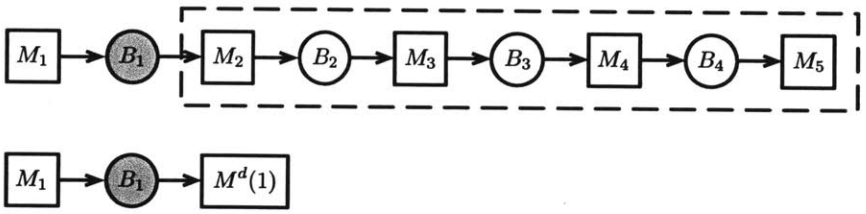


Figure 7-19: Two-machine line representation, $\hat{i} = 1$

Next, assume $1 < \hat{i} < k-1$. Again, we use a five-machine line to illustrate (see Figure 7-20). Without loss of generality, we let $\hat{i} = 2$. (The same argument applies to $\hat{i} = 3$ as well.) We now study the sign of each component of $\nabla \bar{n}_{\hat{i}}(\mathbf{N})$. Here, we group M_1, B_1, and M_2 together, and M_3, M_4, and M_5 as well as B_3 and B_4 together (see Figure 7-20).

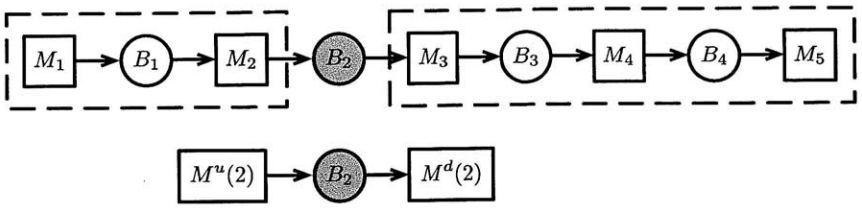


Figure 7-20: Two-machine line representation, $1 < \hat{i} < k-1$ ($\hat{i} = 2$)

It can be seen that, as we enlarge B_3 or B_4, $M^d(2)$ will be faster because M_3, M_4, and M_5 are further decoupled due to a larger B_3 or B_4. As a result, $M^d(2)$ will pull out parts from Buffer B_2 faster than before. Therefore, the average buffer level $\bar{n}_2$ will decrease due to the enlargement of a downstream buffer. This is equivalent to saying that $\partial \bar{n}_2(\mathbf{N})/\partial N_i < 0, \forall i > \hat{i}(= 2)$. On the other hand, if we increase B_1, $M^u(2)$ will be faster. As a result, $M^u(2)$ will put parts into Buffer B_2 faster than before. Therefore, the average buffer level $\bar{n}_2$ will increase due to the enlargement of a upstream buffer. This is equivalent to saying that $\partial \bar{n}_2(\mathbf{N})/\partial N_1 > 0$. Finally, if we increase B_2, $\bar{n}_2(\mathbf{N})$ will increase as well. Consequently, $\partial \bar{n}_2(\mathbf{N})/\partial N_i > 0, \forall i \leq \hat{i}(= 2)$. Therefore, we conclude that for $1 < \hat{i} < k-1$,

$$\begin{cases} \partial \bar{n}_{\hat{i}}(\mathbf{N})/\partial N_i > 0 & \text{if} \quad i = 1, \cdots, \hat{i}, \\ \\ \partial \bar{n}_{\hat{i}}(\mathbf{N})/\partial N_i < 0 & \text{if} \quad i = \hat{i}+1, \cdots, k-1. \end{cases}$$

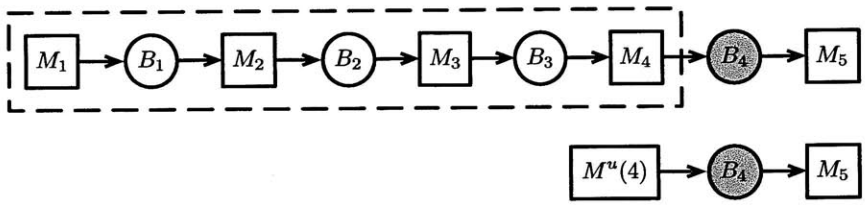


Figure 7-21: Two-machine line representation, $\hat{i} = k-1$ ($\hat{i} = 4$)

Finally, we assume that $\hat{i} = k-1$. Once again, we use a five-machine line to illustrate the point (see Figure 7-21). We group Machines M_1 to M_4 and Buffers B_1 to B_3 together. It can be seen that, as we increase Buffer B_1, B_2, or B_3, $M^u(4)$

will become faster. As a result, $M^u(4)$ will put parts into Buffer B_4 faster than before. Therefore, the average buffer level $\bar{n}_4$ will increase due to the enlargement of a upstream buffer. This is equivalent to saying that $\partial \bar{n}_4(\mathbf{N})/\partial N_1 > 0, \forall i < k-1$. On the other hand, if we increase B_4, $\bar{n}_4(\mathbf{N})$ will increase as well. Consequently, $\partial \bar{n}_4(\mathbf{N})/\partial N_i > 0, \forall i$. This means that all components in $\nabla \bar{n}_{\hat{i}}(\mathbf{N})$ are positive when $\hat{i} = k-1$.

According to the analysis above, we see that $\forall \hat{i} = 1, \cdots, k-1$, the following is true

$$\begin{cases} \partial \bar{n}_{\hat{i}}(\mathbf{N})/\partial N_i > 0 \quad \text{if} \quad i = 1, \cdots, \hat{i}, \\ \\ \partial \bar{n}_{\hat{i}}(\mathbf{N})/\partial N_i < 0 \quad \text{if} \quad i = \hat{i}+1, \cdots, k-1. \end{cases} \tag{7.22}$$

In other words, for $1 \leq \hat{i} < k-1$, $\nabla \bar{n}_{\hat{i}}(\mathbf{N})$ has both negative and positive components, while for $\hat{i} = k-1$, $\nabla \bar{n}_{\hat{i}}(\mathbf{N})$ has only positive components. We also summarize the signs of $\partial \bar{n}_{\hat{i}}(\mathbf{N})/\partial N_i, \forall \hat{i} = 1, \cdots, k-1$ and $i = 1, \cdots, k-1$ in Figure 7-22. We provide numerical experiments to demonstrate the signs of $\partial \bar{n}_{\hat{i}}(\mathbf{N})/\partial N_i, \forall \hat{i} = 1, \cdots, k-1$ and $i = 1, \cdots, k-1$. In particular, the following five five-machine lines are considered (see Table 7.6).

i \ $\hat{i}$	1	2	$\cdots$	j	$\cdots$	$k-1$
1	+	+	$\cdots$	+	$\cdots$	+
2	−	+	$\cdots$	+	$\cdots$	+
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
j	−	−	$\cdots$	+	$\cdots$	+
$j+1$	−	−	$\cdots$	−	$\cdots$	+
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
$k-1$	−	−	$\cdots$	−	$\cdots$	+

Figure 7-22: Summary of the signs of $\partial \bar{n}_{\hat{i}}(\mathbf{N})/\partial N_i$

For each of these five lines, we vary $N_i, i = 1, 2, 3$, and 4 once a time and compute

Table 7.6: Parameters of five five-machine lines

Experiment	1	2	3	4	5
p_1, r_1	.020, .22	.011, .12	.018, .18	.020, .39	.013, .11
p_2, r_2	.015, .15	.040, .33	.033, .31	.037, .33	.033, .38
p_3, r_3	.010, .18	.023, .27	.009, .13	.010, .13	.024, .20
p_4, r_4	.036, .35	.012, .20	.017, .15	.028, .34	.014, .13
p_5, r_5	.021, .39	.017, .19	.022, .40	.010, .15	.014, .17
N_1	5	19	29	44	26
N_2	17	22	58	16	20
N_3	28	28	72	53	39
N_4	34	17	22	9	12

$\partial P(\mathbf{N})/\partial N_i$, $\partial \bar{n}_1(\mathbf{N})/\partial N_i$, $\partial \bar{n}_2(\mathbf{N})/\partial N_i$, $\partial \bar{n}_3(\mathbf{N})/\partial N_i$, and $\partial \bar{n}_4(\mathbf{N})/\partial N_i$. While we vary one N_i, we let other buffers have values shown in the Table 7.6. Since there is no analytical expression of the production rate or the average inventory for long lines, we compute $\partial P(\mathbf{N})/\partial N_i$, $\partial \bar{n}_1(\mathbf{N})/\partial N_i$, $\partial \bar{n}_2(\mathbf{N})/\partial N_i$, $\partial \bar{n}_3(\mathbf{N})/\partial N_i$, and $\partial \bar{n}_4(\mathbf{N})/\partial N_i$ by a forward difference method. Let $\delta N_i = .01$ be the increment of the size of Buffer B_i. Then, the five sets of quantities are calculated by

$$\frac{\partial P(\mathbf{N})}{\partial N_i} = \frac{P(N_1, \cdots, N_i + \delta N_i, \cdots, N_{k-1}) - P(N_1, \cdots, N_i, \cdots, N_{k-1})}{\delta N_i} \qquad i = 1, 2, 3, 4, \tag{7.23}$$

and

$$\frac{\partial \bar{n}_{\hat{i}}(\mathbf{N})}{\partial N_i} = \frac{\bar{n}_{\hat{i}}(N_1, \cdots, N_i + \delta N_i, \cdots, N_{k-1}) - \bar{n}_{\hat{i}}(N_1, \cdots, N_i, \cdots, N_{k-1})}{\delta N_i} \qquad \hat{i} = 1, 2, 3, 4 \text{ and } i = 1, 2, 3, 4. \tag{7.24}$$

The results of these five experiments are summarized in Tables 7.7, 7.8, 7.9, 7.10, and 7.11.

These five experiments demonstrate three points:

1. $\partial P(\mathbf{N})/\partial N_i > 0, \forall i = 1, \cdots, k-1$. This is consistent with the monotonicity of $P(\mathbf{N})$.

2. $\partial \bar{n}_{\hat{i}}(\mathbf{N})/\partial N_i > 0, \forall i = 1, \cdots, \hat{i}$, while $\partial \bar{n}_{\hat{i}}(\mathbf{N})/\partial N_i < 0, \forall i = \hat{i}+1, \cdots, k-1$.

Table 7.7: Five sets of quantities of Experiment 1

i	$\partial P(\mathbf{N})/\partial N_i$	$\partial \bar{n}_1(\mathbf{N})/\partial N_i$	$\partial \bar{n}_2(\mathbf{N})/\partial N_i$	$\partial \bar{n}_3(\mathbf{N})/\partial N_i$	$\partial \bar{n}_4(\mathbf{N})/\partial N_i$
1	.00414064	.50469702	.18368320	.46223830	.09158379
2	.00056910	−.00897171	.13051283	.05962690	.01224530
3	.00003081	−.00091164	−.02739080	.11000819	.00066436
4	.00000003	−.00000100	−.00003065	−.00028952	.00057232

Table 7.8: Five sets of quantities of Experiment 2

i	$\partial P(\mathbf{N})/\partial N_i$	$\partial \bar{n}_1(\mathbf{N})/\partial N_i$	$\partial \bar{n}_2(\mathbf{N})/\partial N_i$	$\partial \bar{n}_3(\mathbf{N})/\partial N_i$	$\partial \bar{n}_4(\mathbf{N})/\partial N_i$
1	.00093878	.68789718	.07571361	.09839447	05803486
2	.00040660	−.02473193	.17126015	.04623048	.02658089
3	.00006988	−.00418794	−.03445800	.10118428	.00441222
4	.00005602	−.00335589	−.02754657	−.18597968	.23038688

Table 7.9: Five sets of quantities of Experiment 3

i	$\partial P(\mathbf{N})/\partial N_i$	$\partial \bar{n}_1(\mathbf{N})/\partial N_i$	$\partial \bar{n}_2(\mathbf{N})/\partial N_i$	$\partial \bar{n}_3(\mathbf{N})/\partial N_i$	$\partial \bar{n}_4(\mathbf{N})/\partial N_i$
1	.00040214	.60000682	.30969401	.58625849	.03262749
2	.00006920	−.01434884	.16666728	.10299528	.00571320
3	.00002935	−.00610637	−.09398285	.41884511	.00182881
4	.00002648	−.00554320	−.09267664	−.32430123	.06846222

Table 7.10: Five sets of quantities of Experiment 4

i	$\partial P(\mathbf{N})/\partial N_i$	$\partial \bar{n}_1(\mathbf{N})/\partial N_i$	$\partial \bar{n}_2(\mathbf{N})/\partial N_i$	$\partial \bar{n}_3(\mathbf{N})/\partial N_i$	$\partial \bar{n}_4(\mathbf{N})/\partial N_i$
1	.00000037	.99970012	.00004039	.00025986	.00000778
2	.00106506	−.03525464	.37710983	.72664696	.02199887
3	.00005185	−.00224009	−.01575971	.22817928	.00093692
4	.00031391	−.01300477	−.08495874	−1.43936738	.26405480

This is consistent with (7.22).

3. When $\hat{i} = k - 1$, $\nabla P(\mathbf{N})$ and $\nabla \bar{n}_{\hat{i}}(\mathbf{N})$ are not proportional.

The three points above are further tested by another 1000 randomly generated lines. In particular, we generate 1000 random lines whose numbers of machines vary

Table 7.11: Five sets of quantities of Experiment 5

i	$\partial P(\mathbf{N})/\partial N_i$	$\partial \bar{n}_1(\mathbf{N})/\partial N_i$	$\partial \bar{n}_2(\mathbf{N})/\partial N_i$	$\partial \bar{n}_3(\mathbf{N})/\partial N_i$	$\partial \bar{n}_4(\mathbf{N})/\partial N_i$
1	.00030768	.66712030	.10055245	.12629490	.01388496
2	.00050094	−.05395109	.67020782	.21958048	.02406870
3	.00031417	−.03310965	−.04046203	.47178788	.01338547
4	.00083551	−.08830334	−.10972111	−.48361869	.28916638

from 4 to 10 according to the case generation method developed by Gershwin (2011). The three points are observed in all those 1000 random lines. We would like to point out an uncommon case where the third point above fails. If the last machine of the line, M_k, is a perfectly reliable machine that does fail, then the average inventory of Buffer B_{k-1} will equal to the production rate of the line. This is because if the sub-line upstream of Buffer B_{k-1} is running, then there will be one part in B_{k-1}; while if the sub-line fails, then there will be no part in B_{k-1}. Therefore, the average inventory level $\bar{n}_{k-1}$ equals to the frequency that the sub-line is running, which is the production rate of the line. Therefore, for this uncommon case, $\nabla P(\mathbf{N})$ and $\nabla \bar{n}_{k-1}(\mathbf{N})$ are the same. However, other than this special case, as long as the last machine of the line is unreliable, we should expect that $\nabla P(\mathbf{N})$ and $\nabla \bar{n}_{k-1}(\mathbf{N})$ are not proportional. Since the production lines studied in this research have all unreliable machines, we safely disregard the special case and conclude that $\nabla P(\mathbf{N})$ and $\nabla \bar{n}_{k-1}(\mathbf{N})$ are not proportional.

The goal for us to study the sign of each component in $\nabla \bar{n}_{\hat{i}}(\mathbf{N})$ is to show that for $\mathbf{N} = \tilde{\mathbf{N}}(\delta)$, $\nabla P(\mathbf{N})$ and $\nabla \bar{n}_{\hat{i}}(\mathbf{N})$ are linearly independent and therefore we can apply the KKT conditions to solve Problem (7.21) when both the production rate constraint and the average part waiting time constraint are active (i.e., Case 2). According to the analysis above, we know that when $\hat{i} = k-1$, $\nabla P(\mathbf{N})$ and $\nabla \bar{n}_{k-1}(\mathbf{N})$ are not proportional. This implies that $\nabla P(\mathbf{N})$ and $\nabla \bar{n}_{k-1}(\mathbf{N})$ are linearly independent, $\forall \mathbf{N}$. Next, we show that for all other $\hat{i} < k-1$, $\nabla P(\mathbf{N})$ and $\nabla \bar{n}_{\hat{i}}(\mathbf{N})$ are also linearly independent, $\forall \mathbf{N}$.

We have shown that $\nabla \bar{n}_{\hat{i}}(\mathbf{N}), \forall \hat{i} < k-1$ has both positive and negative compo-

nents. In addition, all components of $\nabla P(\mathbf{N})$ are positive. Then, it is easy to see that $\nabla \bar{n}_{\hat{i}}(\mathbf{N})$ and $\nabla P(\mathbf{N}), \forall \mathbf{N}$ are linearly independent when $\hat{i} < k-1$. To show this, let $\mathbf{A} = \nabla \bar{n}_{\hat{i}}(\mathbf{N})$ and $\mathbf{B} = \nabla P(\mathbf{N})$. Thus, $\mathbf{A} \in \Re^{k-1}$ and $\mathbf{B} \in \Re^{k-1}$ are vectors and $\mathbf{A}$ has both positive and negative components, while $\mathbf{B}$ has only positive components. By definition, if they are linearly independent, then

$$u_1\mathbf{A} + u_2\mathbf{B} = \mathbf{0}$$

if and only if $u_1 = u_2 = 0$, where u_1 and u_2 are scalars, and the $\mathbf{0} \in \Re^{k-1}$ on the right hand side above is the zero vector. In particular, components A_1 to $A_{\hat{i}}$ in $\mathbf{A}$ are positive, while $\mathbf{A}$'s other components are negative. $u_1\mathbf{A} + u_2\mathbf{B} = \mathbf{0}$ means that

$$u_1 \begin{pmatrix} A_1 \\ \vdots \\ A_{\hat{i}} \\ A_{\hat{i}+1} \\ \vdots \\ A_{k-1} \end{pmatrix} + u_2 \begin{pmatrix} B_1 \\ \vdots \\ B_{\hat{i}} \\ B_{\hat{i}+1} \\ \vdots \\ B_{k-1} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{7.25}$$

Let us analyze if (7.25) can be satisfied with not both u_1 and u_2 being 0. Assume that not both u_1 and u_2 are 0. Since $A_{\hat{i}+1} < 0$ and $B_{\hat{i}+1} > 0$, (7.25) requires u_1 and u_2 to be both positive or both negative. In other words, they have to have the same sign. However, note that $A_{\hat{i}} > 0$ and $B_{\hat{i}} > 0$. As a result, if u_1 and u_2 have the same sign, then $u_1 A_{\hat{i}} + u_2 B_{\hat{i}} \neq 0$, and (7.25) is violated. Therefore, the only way that Equation (7.25) holds is $u_1 = u_2 = 0$. In other words, $\mathbf{A}$ and $\mathbf{B}$ ($\nabla \bar{n}_{\hat{i}}(\mathbf{N})$ and $\nabla P(\mathbf{N}), \forall \mathbf{N}$) are linearly independent.

Based on the analysis above, we conclude that $\nabla P\left(\tilde{\mathbf{N}}(\delta)\right)$ and $\nabla \bar{n}_{\hat{i}}\left(\tilde{\mathbf{N}}(\delta)\right)$ are linearly independent for any $\hat{i} = 1, \cdots, k-1$. Therefore, according to the linear independence constraint qualification (Bertsekas 1999), there exists unique Lagrange multipliers $\mu_i^\star, i = 0, \cdots, k$ for Problem (7.21) to satisfy the KKT conditions:

$$-\nabla J\left(\tilde{\mathbf{N}}(\delta)\right)+\mu_0^\star\nabla\left(\bar{n}_{\hat{i}}\left(\tilde{\mathbf{N}}(\delta)\right)-\delta W_{\hat{i}}P\left(\tilde{\mathbf{N}}(\delta)\right)\right)+\sum_{i=1}^{k-1}\mu_i^\star\nabla\left(N_{\min}-\tilde{N}_i(\delta)\right)$$

$$+\mu_k^\star\nabla\left(\hat{P}-P\left(\tilde{\mathbf{N}}(\delta)\right)\right)=0 \tag{7.26}$$

or

$$-\begin{pmatrix}\frac{\partial J\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_1}\\ \vdots\\ \frac{\partial J\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_{k-1}}\end{pmatrix}+\mu_0^\star\begin{pmatrix}\frac{\partial \bar{n}_{\hat{i}}\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_1}-\delta W_{\hat{i}}\frac{\partial P\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_1}\\ \vdots\\ \frac{\partial \bar{n}_{\hat{i}}\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_{\hat{i}}}-\delta W_{\hat{i}}\frac{\partial P\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_{\hat{i}}}\\ \vdots\\ \frac{\partial \bar{n}_{\hat{i}}\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_{k-1}}-\delta W_{\hat{i}}\frac{\partial P\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_{k-1}}\end{pmatrix} \tag{7.27}$$

$$-\mu_1^\star\begin{pmatrix}1\\ \vdots\\ 0\end{pmatrix}-\cdots-\mu_{k-1}^\star\begin{pmatrix}0\\ \vdots\\ 1\end{pmatrix}-\mu_k^\star\begin{pmatrix}\frac{\partial P\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_1}\\ \vdots\\ \frac{\partial P\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_{k-1}}\end{pmatrix}=\begin{pmatrix}0\\ \vdots\\ 0\end{pmatrix},$$

and

$$\mu_i^\star\geq 0,\forall i=0,\cdots,k, \tag{7.28}$$

$$\mu_0^\star\left(\bar{n}_{\hat{i}}\left(\tilde{\mathbf{N}}(\delta)\right)-\delta W_{\hat{i}}P\left(\tilde{\mathbf{N}}(\delta)\right)\right)=0, \tag{7.29}$$

$$\mu_i^\star\left(N_{\min}-\tilde{N}_i(\delta)\right)=0,\forall i=1,\cdots,k-1 \tag{7.30}$$

$$\mu_k^\star\left(\hat{P}-P\left(\tilde{\mathbf{N}}(\delta)\right)\right)=0, \tag{7.31}$$

where $\tilde{\mathbf{N}}(\delta)$ is the optimal solution of (7.21). Similar to the discussion in Section 4.2.1, by assuming an interior solution, we simplify the KKT conditions to

$$-\begin{pmatrix} \frac{\partial J\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_1} \\ \vdots \\ \frac{\partial J\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_{k-1}} \end{pmatrix} + \mu_0^\star \begin{pmatrix} \frac{\partial \bar{n}_{\hat{i}}\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_1} - \delta W_{\hat{i}} \frac{\partial P\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_1} \\ \vdots \\ \frac{\partial \bar{n}_{\hat{i}}\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_{\hat{i}}} - \delta W_{\hat{i}} \frac{\partial P\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_{\hat{i}}} \\ \vdots \\ \frac{\partial \bar{n}_{\hat{i}}\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_{k-1}} - \delta W_{\hat{i}} \frac{\partial P\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_{k-1}} \end{pmatrix} \tag{7.32}$$

$$-\mu_k^\star \begin{pmatrix} \frac{\partial P\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_1} \\ \vdots \\ \frac{\partial P\left(\tilde{\mathbf{N}}(\delta)\right)}{\partial N_{k-1}} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix},$$

$$\mu_0^\star \left(\bar{n}_{\hat{i}}\left(\tilde{\mathbf{N}}(\delta)\right) - \delta W_{\hat{i}} P\left(\tilde{\mathbf{N}}(\delta)\right)\right) = 0, \tag{7.33}$$

and

$$\mu_k^\star \left(\hat{P} - P\left(\tilde{\mathbf{N}}(\delta)\right)\right) = 0. \tag{7.34}$$

Since $\tilde{\mathbf{N}}(\delta)$ is not the optimal solution of the corresponding unconstrained problem of (7.21), $\nabla J(\tilde{\mathbf{N}}(\delta)) \neq \mathbf{0}$ where $\mathbf{0} \in \Re^{k-1}$ is the zero vector, because otherwise both the production rate constraint and the average part waiting time constraint will be inactive. $\nabla J(\tilde{\mathbf{N}}(\delta)) \neq \mathbf{0}$ means that not all $\partial J(\tilde{\mathbf{N}}(\delta))/\partial N_i, i = 1, \cdots, k-1$ equal 0. Thus, $\mu_0^\star$ and $\mu_k^\star$ cannot be 0 simultaneously, since otherwise condition (7.32) would be violated. Moreover, if either of $\mu_0^\star$ and $\mu_k^\star$ is 0, it means that the constraint it corresponds to is inactive, and the problem (Case 2) is relaxed to a simpler one (Case 3 or Case 4). Therefore, we argue that both $\mu_0^\star$ and $\mu_k^\star$ are positive. By (7.33) and (7.34), the optimal solution $\tilde{\mathbf{N}}(\delta)$ satisfies $P(\tilde{\mathbf{N}}(\delta)) = \hat{P}$ and $\bar{n}_{\hat{i}}(\tilde{\mathbf{N}}(\delta)) = \delta W_{\hat{i}} P(\tilde{\mathbf{N}}(\delta))$. In addition, (7.32) to (7.34) reveal how we could find $\mu_0^\star$, $\mu_k^\star$ and $\tilde{\mathbf{N}}(\delta)$. For every combination of $\mu_0^\star$ and $\mu_k^\star$, (7.32) determines $\tilde{\mathbf{N}}(\delta)$ since there are $k-1$

equations and $k-1$ unknowns. Therefore, we can think of $\tilde{\mathbf{N}}(\delta) = \tilde{\mathbf{N}}(\delta)(\mu_0^\star, \mu_k^\star)$. We search for values of $\mu_0^\star$ and $\mu_k^\star$ such that $P(\tilde{\mathbf{N}}(\delta)(\mu_0^\star, \mu_k^\star)) = \hat{P}$ and $\bar{n}_{\hat{i}}(\tilde{\mathbf{N}}(\delta)(\mu_0^\star, \mu_k^\star)) = \delta W_{\hat{i}} P(\tilde{\mathbf{N}}(\delta)(\mu_0^\star, \mu_k^\star))$. In what follows, we indicate how to find the Lagrange multipliers and the optimal solution of the problem. Replacing $\mu_0^\star$ by $\mu_0 > 0$ and $\mu_k^\star$ by $\mu_k > 0$ in (7.32) gives

$$-\begin{pmatrix} \frac{\partial J(\bar{\mathbf{N}})}{\partial N_1} \\ \vdots \\ \frac{\partial J(\bar{\mathbf{N}})}{\partial N_{k-1}} \end{pmatrix} + \mu_0 \begin{pmatrix} \frac{\partial \bar{n}_{\hat{i}}(\bar{\mathbf{N}})}{\partial N_1} - \delta W_{\hat{i}} \frac{\partial P(\bar{\mathbf{N}})}{\partial N_1} \\ \vdots \\ \frac{\partial \bar{n}_{\hat{i}}(\bar{\mathbf{N}})}{\partial N_{\hat{i}}} - \delta W_{\hat{i}} \frac{\partial P(\bar{\mathbf{N}})}{\partial N_{\hat{i}}} \\ \vdots \\ \frac{\partial \bar{n}_{\hat{i}}(\bar{\mathbf{N}})}{\partial N_{k-1}} - \delta W_{\hat{i}} \frac{\partial P(\bar{\mathbf{N}})}{\partial N_{k-1}} \end{pmatrix} - \mu_k \begin{pmatrix} \frac{\partial P(\bar{\mathbf{N}})}{\partial N_1} \\ \vdots \\ \frac{\partial P(\bar{\mathbf{N}})}{\partial N_{k-1}} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \quad (7.35)$$

where $\bar{\mathbf{N}}$ is the unique solution of (7.35). Note that $\bar{\mathbf{N}}$ is the solution of the following optimization problem

$$\begin{aligned} \min_{\mathbf{N}} \quad -\bar{J}(\mathbf{N}) &= -J(\mathbf{N}) + \mu_0 \left(\bar{n}_{\hat{i}}(\mathbf{N}) - \delta W_{\hat{i}} P(\mathbf{N})\right) - \mu_k \left(\hat{P} - P(\mathbf{N})\right) \\ \text{subject to} \quad N_{\min} - N_i &\leq 0, \forall i = 1, \cdots, k-1, \end{aligned} \quad (7.36)$$

which is equivalent to

$$\begin{aligned} \max_{\mathbf{N}} \quad \bar{J}(\mathbf{N}) &= J(\mathbf{N}) - \mu_0 \left(\bar{n}_{\hat{i}}(\mathbf{N}) - \delta W_{\hat{i}} P(\mathbf{N})\right) + \mu_k P(\mathbf{N}) \\ \text{subject to} \quad N_{\min} - N_i &\leq 0, \forall i = 1, \cdots, k-1, \end{aligned} \quad (7.37)$$

or

$$\begin{aligned} \max_{\mathbf{N}} \quad \bar{J}(\mathbf{N}) &= AP(\mathbf{N}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i(\mathbf{N}) \\ &\quad -\mu_0 \left(\bar{n}_{\hat{i}}(\mathbf{N}) - \delta W_{\hat{i}} P(\mathbf{N})\right) + \mu_k P(\mathbf{N}) \\ \text{subject to} \quad N_{\min} - N_i &\leq 0, \forall i = 1, \cdots, k-1, \end{aligned} \tag{7.38}$$

or

$$\begin{aligned} \max_{\mathbf{N}} \quad \bar{J}(\mathbf{N}) &= \left(A + \mu_0 \delta W_{\hat{i}} + \mu_k\right) P(\mathbf{N}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i(\mathbf{N}) \\ &\quad -\mu_0 \bar{n}_{\hat{i}}(\mathbf{N}) \\ \text{subject to} \quad N_{\min} - N_i &\leq 0, \forall i = 1, \cdots, k-1. \end{aligned} \tag{7.39}$$

Therefore, when both the production rate constraint and the average part waiting time constraint are active (i.e., Case 2), we solve (7.21) as follows. For every $\{\mu_0 > 0, \mu_k > 0\}$, we find the corresponding optimal solution $\bar{\mathbf{N}}$ that satisfies (7.35) by solving (7.39), and we need to find the $\{\mu_0' > 0, \mu_k' > 0\}$ such that the solution to (7.39), denoted as $\mathbf{N}(\mu_0', \mu_k')$, satisfies $P(\mathbf{N}(\mu_0', \mu_k')) = \hat{P}$ and $\bar{n}_{\hat{i}}(\mathbf{N}(\mu_0' > 0, \mu_k' > 0)) = \delta W_{\hat{i}} P(\mathbf{N}(\mu_0', \mu_k'))$. Then, μ_0', μ_k' and $\mathbf{N}(\mu_0', \mu_k')$ satisfy (7.32) to (7.34). Hence, μ_0' and μ_k' are exactly the Lagrange multipliers satisfying the KKT conditions of Problem (7.21), and $\tilde{\mathbf{N}}(\delta) = \mathbf{N}(\mu_0', \mu_k')$ is the optimal solution of Problem (7.21). Consequently, solving (7.21) with both the production rate constraint and the average part waiting time constraint being active (i.e., Case 2) through the procedure above is essentially finding the unique Lagrange multipliers and optimal solution of the problem.

We state the algorithm that solves Problem (7.21) for a given δ. As a reminder, there are five potential cases for the problem. We repeat the five cases here:

1. Case 1: the production rate constraint conflicts with the average part waiting time constraint. There is no feasible solution for the problem.

2. Case 2: both the production rate constraint and the average part waiting time constraint are active.

3. Case 3: the production rate constraint is active, but the average part waiting time constraint is inactive.

4. Case 4: the production rate constraint is inactive, but the average part waiting time constraint is active.

5. Case 5: both the production rate constraint and the average part waiting time constraint are inactive.

Let $\tilde{\mathbf{N}}(\delta)$ be the optimal solution. The algorithm consists of five steps and they are stated below:

1. Check the feasibility of the problem. That is to check

   - if $\hat{P}$ is feasible for the problem without the average part waiting time constraint;
   - if the production rate constraint and the average part waiting time constraint conflict with each other.

   If the problem is feasible, go to Step 2. Otherwise, the problem is a Case 1 problem and it is infeasible.

2. Solve the corresponding unconstrained problem without both constraints. That is to maximize the profit of the production line without the production rate constraint or the average part waiting time constraint. Let $\mathbf{N}^U$ be the solution of the unconstrained problem. Then, check whether both $\bar{n}_{\hat{i}}(\mathbf{N}^U) \leq \delta W_{\hat{i}} P(\mathbf{N}^U)$ and $P(\mathbf{N}^U) \geq \hat{P}$ are satisfied:

   - if both are satisfied, then we are done and $\tilde{\mathbf{N}}(\delta) = \mathbf{N}^U$. Both constraints are ineffective. The problem is a Case 5 problem;
   - if $\bar{n}_{\hat{i}}(\mathbf{N}^U) \leq \delta W_{\hat{i}} P(\mathbf{N}^U)$ but $P(\mathbf{N}^U) < \hat{P}$, we know that the production rate constraint is violated. Go to Step 3;

- if $P(\mathbf{N}^U) \geq \hat{P}$ but $\bar{n}_{\hat{i}}(\mathbf{N}^U) > \delta W_{\hat{i}} P(\mathbf{N}^U)$, we know that the average part waiting time constraint is violated. Go to Step 4;
- if both constraints are not satisfied, then go to Step 5.

3. Solve the profit maximization problem with the production rate constraint being effective. Let $\mathbf{N}^P$ be the solution. Check whether $\bar{n}_{\hat{i}}(\mathbf{N}^P) \leq \delta W_{\hat{i}} P(\mathbf{N}^P)$ is satisfied:
   - if $\bar{n}_{\hat{i}}(\mathbf{N}^P) \leq \delta W_{\hat{i}} P(\mathbf{N}^P)$, then $\mathbf{N}^P$ satisfies both constraints. In particular, the average part waiting time constraint is inactive. $\tilde{\mathbf{N}}(\delta) = \mathbf{N}^P$ and the problem is a Case 3 problem;
   - if $\bar{n}_{\hat{i}}(\mathbf{N}^P) > \delta W_{\hat{i}} P(\mathbf{N}^P)$, then the average part waiting time constraint is violated by $\mathbf{N}^P$. Go to Step 5.

4. Solve the profit maximization problem with the average part waiting time constraint being effective. Let $\mathbf{N}^T$ be the solution. Check whether $P(\mathbf{N}^T) \geq \hat{P}$ is satisfied:
   - if $P(\mathbf{N}^T) \geq \hat{P}$, then $\mathbf{N}^T$ satisfies both constraints. In particular, the production rate constraint is inactive. $\tilde{\mathbf{N}}(\delta) = \mathbf{N}^T$ and the problem is a Case 4 problem;
   - if $P(\mathbf{N}^U) < \hat{P}$, then the average part waiting time constraint is violated by $\mathbf{N}^T$. Go to Step 5.

5. Solve the profit maximization problem with both production rate constraint and average part waiting time constraint as equalities (i.e., being effective). To solve this, apply the technique explained in this section. This requires to conducting a two-dimensional search in μ_0 and μ_k. For each pair of μ_0 and μ_k, solve the unconstrained problem (7.39). Stop the iteration process once $\mu_0^\star$ and $\mu_k^\star$ are found such that $\tilde{\mathbf{N}}(\delta) = \mathbf{N}(\mu_0^\star, \mu_k^\star)$ satisfies both the production rate constraint and the average part waiting time constraint. The problem is a Case 2 problem.

7.4 Numerical Experiments

In this section, we study 200 four-machine three-buffer lines to show the accuracy of the proposed algorithm that solves the transformed problem (7.21) iteratively, to find the optimal solution of the original problem (7.2) (see Section 7.3.1). These lines are constructed according to the method of Gershwin (2011). In all these lines, the isolated efficiency $e_i = r_i/(r_i + p_i)$ of any machine is between .923 and .952 with r_i and p_i generated randomly. In addition, the buffer cost coefficients b_i and c_i for any buffer are also generated randomly. The target production rate $\hat{P}$ is among .86, .87, and .88 parts per time unit for all experiments. The revenue coefficient A is 2000. For each line, the buffer where we impose the average part waiting time constraint is randomly chosen from all three buffers. In addition, the waiting time upper limit is chosen in a way that the average part waiting time constraint and the production rate constraint do not conflict.

We compare the results from the algorithm with surface search and compute three types of errors. They are the profit error, the production rate error, and the maximum buffer size error. As a reminder, in the surface search, we use the optimal solution from the algorithm as the center point. We search around its adjacent area with a reasonable range of each N_i. For every buffer distribution $\mathbf{N}$ in this area, we first check if it satisfies both the production rate constraint $P(\mathbf{N} \geq \hat{P})$ and the maximum part waiting time probability constraint $\mathbf{p}(T(\mathbf{N}) \leq W_i) \geq 1 - \alpha$. If and only if $\mathbf{N}$ satisfies both constraints, it can be considered as a feasible points. After we find all the feasible points in this area, we compute the profits for all those feasible points and choose the one that gives us the maximum profit as the optimal solution of the surface search method. The surface search method deals with the original problem (7.2) directly. The comparison between the solution from our algorithm and the solution from the surface search demonstrates the accuracy of our algorithm in solving the original problem (7.2).

We use $\mathbf{N}^{\star}_{\text{alg}}$ and $\mathbf{N}^{\star}_{\text{SS}}$ to denote the optimal buffer allocations from the algorithm described in Section 7.3.1 and the surface search, respectively. Then, the three types

of errors are computed (which are the same as those considered in Chapter 4) as:

$$J_{\text{err}} = \left| \frac{J(\mathbf{N}^{\star}_{\text{ss}}) - J(\mathbf{N}^{\star}_{\text{alg}})}{J(\mathbf{N}^{\star}_{\text{ss}})} \right| \times 100\%,$$

$$P_{\text{err}} = \left| \frac{P(\mathbf{N}^{\star}_{\text{ss}}) - P(\mathbf{N}^{\star}_{\text{alg}})}{P(\mathbf{N}^{\star}_{\text{ss}})} \right| \times 100\%,$$

and finally

$$N_{\text{err}} = \max_{i=1,\cdots,k-1} \left\{ \left| \frac{\mathbf{N}^{\star}_{\text{ss}}(B_i) - \mathbf{N}^{\star}_{\text{alg}}(B_i)}{\mathbf{N}^{\star}_{\text{ss}}(B_i)} \right| \times 100\% \right\}.$$

In addition to the three types of errors mentioned above, there is one more error to consider. Recall that the analytical solution of part waiting time distribution for two-machine lines is applied to a single buffer in long lines with the help of decomposition. In particular, the pseudo-machine parameters of the two-machine building block that contains the buffer are used to compute the part waiting time distribution of that buffer. Since the decomposition method is an approximation, error may exist between the probability $\mathbf{p}\left(T\left(\mathbf{N}^{\star}\right) \leq W_{\hat{i}}\right)$ we compute this way and the actual probability without an approximation method[8]. Therefore, we compare $\mathbf{p}\left(T\left(\mathbf{N}^{\star}\right) \leq W_{\hat{i}}\right)$ from the algorithm with the one from simulation. A discrete time simulation for production lines is written for this purpose to calculate the part waiting time distribution of any buffer and therefore to compute the desired probability. For each line, with the optimal buffer distribution $\mathbf{N}^{\star}$ derived from the algorithm, the length of each simulation is 11,000,000 time units with the first 1,000,000 time units being the warm up period, and we run the simulation 20 times and use their average as the simulation result. Finally, we set $\alpha = 0.1$ and therefore for each line at least 90% of the parts should have waiting times (in a given buffer) not greater than its upper limit.

The production rate error, the profit error, and the maximum buffer size error for the 200 four-machine lines are illustrated in Figure 7-23. For each type error in

[8]For notation simplicity, we drop the alg subscript and let $\mathbf{N}^{\star}$ denote the optimal solution from the algorithm.

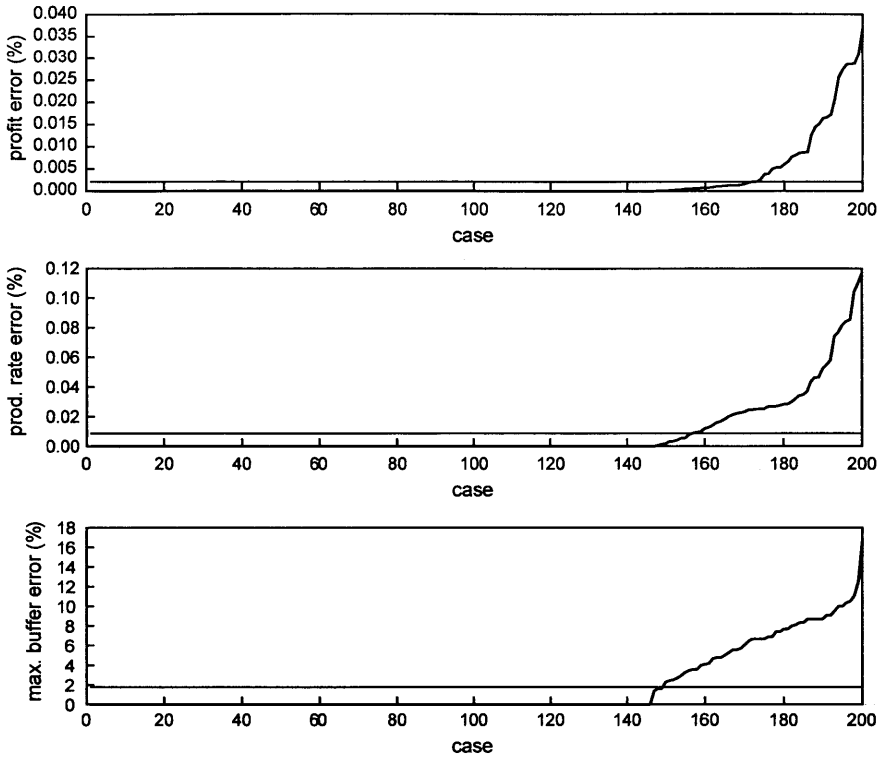


Figure 7-23: Results of two hundred randomly generated deterministic single failure mode four-machine lines, both constraints

Figure 7-23, we rank the three types of error in their corresponding ascending orders respectively. (Therefore, the ith case in the production rate error graph, for instance, may not necessary be the same as the ith case in the profit error graph.) The average error of each type is also provided. In particular, in 146 out of the 200 cases, the optimal buffer sizes from the algorithm and the surface search are the same, and therefore the three types of error in these 146 cases are 0. In addition, the average profit error, the average production rate error, and the average maximum buffer error of these 200 cases are .002%, .01%, and 1.77%, respectively.

Figures 7-24 and 7-25 demonstrate $\mathbf{p}\left(T\left(\mathbf{N}^{\star}\right) \leq W_{\hat{i}}\right)$, which is the probability that the waiting time of a part in Buffer $B_{\hat{i}}$ does not exceed the upper limit $W_{\hat{i}}$. The errors in Figure 7-25 are computed by the probability found by simulation minus the probability found by the analytical solution and ranked according to the descending order. As mentioned previously in this section, the decomposition method is approx-

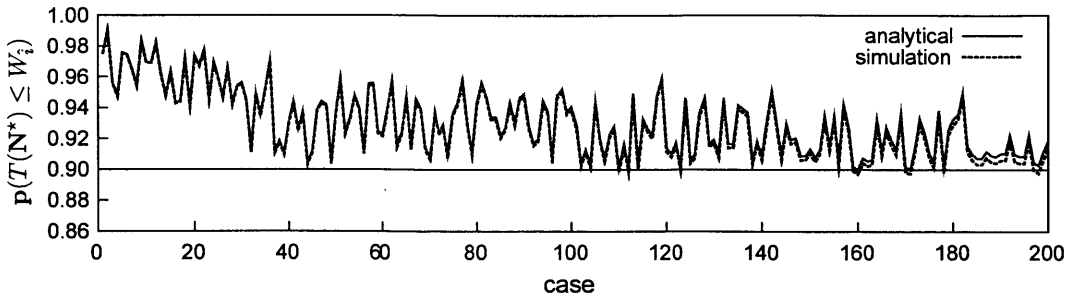


Figure 7-24: $\mathbf{p}\left(T\left(\mathbf{N}^{\star}\right) \leq W_{\hat{i}}\right)$ of the two hundred cases from both the analytical solution and the simulation

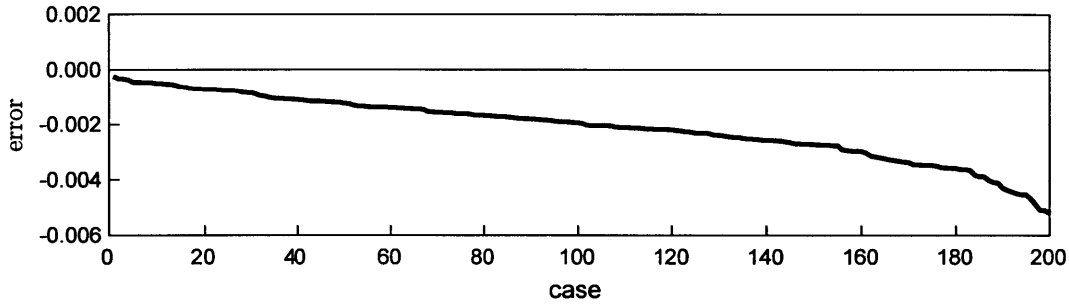


Figure 7-25: Error in $\mathbf{p}\left(T\left(\mathbf{N}^{\star}\right) \leq W_{\hat{i}}\right)$ between the analytical solution and the simulation

imate and therefore we compare $\mathbf{p}\left(T\left(\mathbf{N}^{\star}\right) \leq W_{\hat{i}}\right)$ from the analytical solution with that from simulation. It can be seen from Figure 7-24 that they are very close to each other, which indicates the accuracy of the analytical approach in computing the cumulative distribution function (CDF) of the part waiting time in a buffer in a long line when compared with simulation. Furthermore, Figure 7-25 suggests that the analytical approach, which makes use of the decomposition method to compute the waiting time probability for a buffer in long lines, slightly overestimates the probability as compared to simulation. This is because the errors in all 200 cases are negative. In all these 200 cases, there are 10 cases whose $\mathbf{p}\left(T\left(\mathbf{N}^{\star}\right) \leq W_{\hat{i}}\right)$ from simulation is slightly smaller than the target value of 90%. All other 190 cases satisfy the constraint. In addition, it is helpful to indicate that since for the deterministic production line model, the part waiting time is discrete and therefore its CDF is a

piecewise constant function of $T(N)$. This is why this constraint is in general not satisfied with equality because it is unlikely to have $\mathbf{p}\left(T\left(\mathbf{N}^{\star}\right) \leq W_{\hat{i}}\right) = 0.9$ exactly. Therefore in most of the cases, the constraint is satisfied with $\mathbf{p}\left(T\left(\mathbf{N}^{\star}\right) \leq W_{\hat{i}}\right) > 0.9$, but they do not suggest that the maximum part waiting time constraint is inactive in the problem. The results of these 200 numerical experiments indicate the accuracy and reliability of the algorithm.

Note that if one or more machines downstream of Buffer $B_{\hat{i}}$ are very unreliable with both large up-times and down-times, most of the parts could have very long waiting times in $B_{\hat{i}}$, and this could make the maximum part waiting time constraint always infeasible. This is because once such a downstream machine (say M_j) fails, it will take a very long time to get repaired. During this time, machines upstream of M_j will start getting blocked and eventually $B_{\hat{i}}$ will get full. Thus, parts need to wait in $B_{\hat{i}}$ for a long time due to the failure of M_j. In this case, the constraint $\mathbf{p}(T(\mathbf{N}) \leq W_{\hat{i}}) \geq 1 - \alpha$ could always be infeasible for the line regardless of how small the size of $B_{\hat{i}}$ is. In our approach of solving the transformed problem (7.21) iteratively, we will keep searching δ until $\mathbf{p}\left(T(\tilde{\mathbf{N}}(\delta)) \leq W_{\hat{i}}\right) \geq 1 - \alpha$ is satisfied. However, since the part waiting time constraint is always infeasible, we will not be able to find the desirable value of δ no matter how small it is. This will tell us that the original maximum part waiting time constraint is infeasible for the problem. Or, once the δ is too small, the transformed average part waiting time constraint conflicts with the production rate constraint. Then the problem becomes infeasible as well.

Finally, as desirable future work directions, we could consider developing the closed form of $\mathbf{p}(T(\mathbf{N}) \leq W_{\hat{i}})$, and then solving the original problem (7.2) directly.

Chapter 8

The Segmentation Method for Long Line Optimization

8.1 Motivation

In Chapter 4, we develop an efficient buffer design algorithm for production line profit maximization. As a reminder, the algorithm solves the following problem:

$$\begin{aligned} \max_{\mathbf{N}} \quad & J(\mathbf{N}) = AP(\mathbf{N}) - \sum_{i=1}^{K-1} b_i N_i - \sum_{i=1}^{K-1} c_i \bar{n}_i(\mathbf{N}) \\ \text{subject to} \quad & P(\mathbf{N}) \geq \hat{P}, \\ & N_i \geq N_{\min}, \quad i = 1, \cdots, k-1, \end{aligned} \tag{8.1}$$

where $J(\mathbf{N})$ is the profit of the line, A (\$/part) is the revenue coefficient associated with the production rate $P(\mathbf{N})$, while b_i and c_i (\$/part/time unit) are cost coefficients associated with the buffer space and average inventory of Buffer B_i. $\hat{P}$ is the target production rate. To solve (8.1), the algorithm starts with solving a corresponding unconstrained problem[1]

[1]We have pointed out in Section 4.1.2 that the unconstrained problem is a convenient, although not quite accurate, name for (8.2) since we still have the buffer size constraint.

$$\begin{aligned} \max_{\mathbf{N}} \quad & J(\mathbf{N}) = AP(\mathbf{N}) - \sum_{i=1}^{K-1} b_i N_i - \sum_{i=1}^{K-1} c_i \bar{n}_i(\mathbf{N}) \\ \text{subject to} \quad & N_i \geq N_{\min}, \quad i = 1, \cdots, k-1. \end{aligned} \tag{8.2}$$

If the solution of (8.2), say $\mathbf{N}^u$, satisfies the production rate constraint, then it is also the optimal solution of the original problem (8.1). However, if $\mathbf{N}^u$ does not satisfy the production rate constraint, the algorithm conducts an one-dimensional search over the revenue coefficient $A' > A$ and solves (8.2) iteratively for different A's until it finds a value of A' for which the solution of the unconstrained problem (8.2), denoted by $\mathbf{N}'(A')$, satisfies $P(\mathbf{N}'(A')) = \hat{P}$. Then the optimal solution of (8.1) is $\mathbf{N}^\star = \mathbf{N}'(A')$. The algorithm solves Problem (8.1) efficiently.

However, both Chapter 4 and Shi and Gershwin (2009a) indicate that the computer time of the algorithm increases exponentially with the length of the line when $\mathbf{N}^u$ does not satisfy the production rate constraint and therefore the one-dimensional search over $A' > A$ is adopted to find $\mathbf{N}^\star = \mathbf{N}'(A')$ such that $P(\mathbf{N}'(A')) = \hat{P}$ (see Figure 4-7). This is because there are two factors in the algorithm that determine the computer time. They are the time required for solving (8.2) for a given A' in each iteration and the number of iterations. As the length of the line increases, both factors increase and they lead to a drastic increase in computer time. The long computer time is undesirable in the design and operation of long production lines. Therefore, it is desirable and important to find a method to reduce the computer time in long line optimization, while assuring the accuracy of the optimization algorithm. In this chapter, we propose a segmentation method that is proved to achieve these goals. The method is accurate, reliable, and greatly reduces the computer time when $\mathbf{N}^\star$ of (8.1) is such that the production rate constraint is satisfied with equality. This should be determined as a first step in the method and it does not require very much computer time.

The materials of this chapter are structured as follows. We first demonstrate the segmentation method with two sets of examples on perfectly balanced long lines

as well as unbalanced long lines in Sections 8.2 and 8.3. After that, the method is presented and explained formally in Section 8.4. Two strategies that can be used to improve the accuracy of the segmentation method are also addressed, following by more numerical experiments to show the efficiency of the method in Section 8.5.

8.2 Qualitative Behavior of Perfectly Balanced Lines

We first provide two examples of perfectly balanced long lines to show how the optimal solution of the original line can be found by the segmentation method. The two lines being considered are a 20-machine line and a 30-machine line. As the name suggests, a perfectly balanced line has identical machines and identical buffers[2].

8.2.1 A 20-Machine Line Example

The 20-machine 19-buffer line has 20 identical machines and 19 identical buffers. Machine parameters are $r_i = .1$ and $p_i = .01$, $i = 1, \cdots, 20$. Buffer parameters are $b_i = c_i = 1, i = 1, \cdots, 19$, and the revenue coefficient $A = 10,000$. The target production rate is .88 parts per time unit. Suppose that instead of optimizing the original 20-machine 19-buffer line, we optimize the following three 10-machine 9-buffer lines that are constructed by the machines and buffers of the original line:

- the first 10-machine 9-buffer line is $M_1 - B_1 - M_2 - \cdots - B_9 - M_{10}$;
- the second 10-machine 9-buffer line is $M_6 - B_6 - M_7 - \cdots - B_{14} - M_{15}$; and
- the third 10-machine 9-buffer line is $M_{11} - B_{11} - M_{12} - \cdots - B_{19} - M_{20}$.

The segmentation of the original 20-machine 19-buffer line is illustrated in Figure 8-1.

For these three 10-machine lines, we modify the revenue coefficient A to 5,000. The value of A is modified such that the production rate constraint will be satisfied with equality in all line segments as well. (We further comment on the effect of the

[2]In a perfectly balanced line, all machines have exactly the same repair and failure probabilities and therefore the same isolated production rates. However, machines are still unreliable.

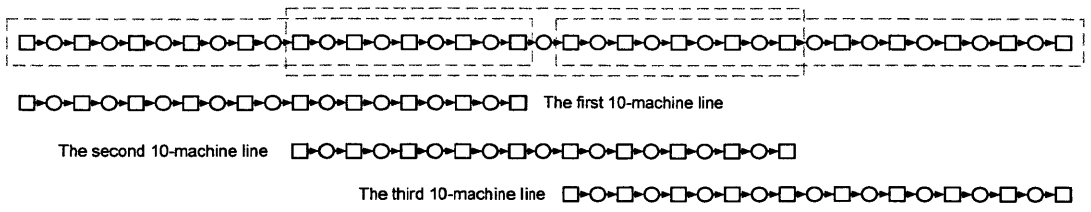


Figure 8-1: The segmentation of a 20-machine 19-buffer line

revenue coefficient A when we explain the segmentation method in Section 8.4.1.) Let $\mathbf{N}_1^\star$, $\mathbf{N}_2^\star$, and $\mathbf{N}_3^\star$ be the optimal buffer distributions of the three lines. They are provided in Table 8.1[3] as well as Figure 8-2. For illustration purpose, we keep the original buffer indices. For instance, the nine buffers in the second 10-machine line are labelled as Buffers B_6, B_7, $\cdots$, B_{14}. The optimal buffer distributions of the three 10-machine lines are exactly the same despite the indices of buffers, because the three lines are identical.

Table 8.1: The optimal buffer distribution of a perfectly balanced line, Example 1

The first line	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9
$\mathbf{N}_1^\star$	59.00	83.89	92.16	94.63	95.20	94.97	93.63	89.15	73.12
The second line	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}
$\mathbf{N}_2^\star$	59.00	83.89	92.16	94.63	95.20	94.97	93.63	89.15	73.12
The third line	B_{11}	B_{12}	B_{13}	B_{14}	B_{15}	B_{16}	B_{17}	B_{18}	B_{19}
$\mathbf{N}_3^\star$	59.00	83.89	92.16	94.63	95.20	94.97	93.63	89.15	73.12

We take some results from each segment and construct the approximate solution $\mathbf{N}_{\text{seg}}$ for the original line who has 19 buffers as follows (and we further explain the construction in Section 8.4):

$$\begin{aligned}\mathbf{N}_{\text{seg}} = \Big(&\mathbf{N}_1^\star(B_1), \mathbf{N}_1^\star(B_2), \cdots, \mathbf{N}_1^\star(B_7), \mathbf{N}_2^\star(B_8), \mathbf{N}_2^\star(B_9), \cdots, \mathbf{N}_2^\star(B_{12}), \\ &\mathbf{N}_3^\star(B_{13}), \mathbf{N}_3^\star(B_{14}), \cdots, \mathbf{N}_3^\star(B_{19})\Big).\end{aligned} \tag{8.3}$$

[3]We have argued in Chapter 4 that buffer sizes can be treated as continuous variables in the algorithm. Therefore, $\mathbf{N}_1^\star$, $\mathbf{N}_2^\star$, and $\mathbf{N}_3^\star$ are not integers. We keep them as non-integers for illustration purpose.

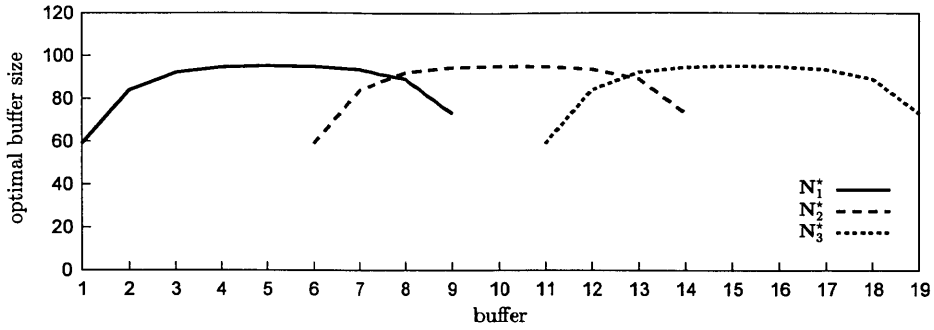


Figure 8-2: The optimal buffer distributions of the three 10-machine lines, a perfectly balanced line, Example 1

For comparison, we optimize the original 20-machine 19-buffer line directly and let $\mathbf{N}^\star$ be the optimal buffer distribution. We display $\mathbf{N}^\star$ and $\mathbf{N}_{\text{seg}}$ in Figure 8-3.

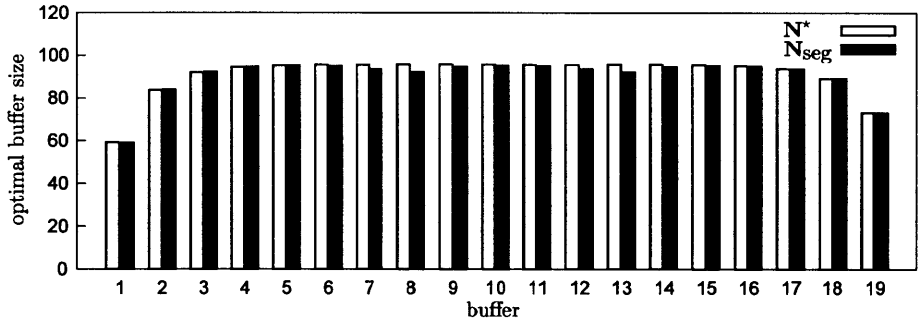


Figure 8-3: Comparison between $\mathbf{N}^\star$ and $\mathbf{N}_{\text{seg}}$, a perfectly balanced line, Example 1

Figure 8-3 shows that $\mathbf{N}_{\text{seg}}$ is very close to $\mathbf{N}^\star$. The computer time required to optimize the original 20-machine 19-buffer line directly is 483.74 seconds, while the computer time for the segmentation method is 125.00 seconds. It can be seen that the segmentation method reduces the computer time dramatically. The production rate $P(\mathbf{N}^\star)$ is .8800 parts per time unit, while $P(\mathbf{N}_{\text{seg}})$ is slightly smaller at .8798 parts per time unit. The profit rate $J(\mathbf{N}^\star)$ is \$6259.11 per time unit, while $J(\mathbf{N}_{\text{seg}})$ is \$6270.34 per time unit. Note that $J(\mathbf{N}_{\text{seg}}) > J(\mathbf{N}^\star)$ does not indicate that $\mathbf{N}^\star$ is not the optimal solution. However, this is because the production rate associate with $\mathbf{N}_{\text{seg}}$ is smaller than $\hat{P}$. Thus, if we optimize the original line directly, $\mathbf{N}_{\text{seg}}$ will be an infeasible solution to the problem. With $\mathbf{N}_{\text{seg}}$, the production rate of the line is

slightly lower than $\hat{P}$, while the buffer space cost and the inventory holding cost are also smaller. The combined effect of the revenue and the cost leads to a slightly higher profit rate than $J(\mathbf{N}^\star)$. (We observe $J(\mathbf{N}_{\text{seg}}) > J(\mathbf{N}^\star)$ in the following examples as well.) We will further study the difference between $\mathbf{N}^\star$ and $\mathbf{N}_{\text{seg}}$, the lengths of the line segments, and the computer time in detail in Section 8.4.

Before leaving this example, we want to repeat the important fact that in the optimization for the original 20-machine line and those three segmented 10-machine lines, the production rate constraint $P(\mathbf{N}) \geq \hat{P}$ is always satisfied with equality. In other words, all $\mathbf{N}^\star$, $\mathbf{N}_1^\star$, $\mathbf{N}_2^\star$, $\mathbf{N}_3^\star$ satisfy $P(\mathbf{N}^\star) = \hat{P}$, $P(\mathbf{N}_1^\star) = \hat{P}$, $P(\mathbf{N}_2^\star) = \hat{P}$, and $P(\mathbf{N}_3^\star) = \hat{P}$. As we show in Section 8.4.1, the fact that the production rate constraint is satisfied with equality is a crucial prerequisite of the segmentation method.

8.2.2 A 30-Machine Line Example

Next we consider a perfectly balanced 30-machine 29-buffer line. The parameters of each machine and each buffer are the same as those in the previous case. The revenue coefficient is $A = 15,000$. The target production rate is again .88 parts per time unit. We segment the original line into five 10-machine lines:

- the first 10-machine 9-buffer line is $M_1 - B_1 - M_2 - \cdots - B_9 - M_{10}$;
- the second 10-machine 9-buffer line is $M_6 - B_6 - M_7 - \cdots - B_{14} - M_{15}$;
- the third 10-machine 9-buffer line is $M_{11} - B_{11} - M_{12} - \cdots - B_{19} - M_{20}$;
- the fourth 10-machine 9-buffer line is $M_{16} - B_{16} - M_{17} - \cdots - B_{24} - M_{25}$; and
- the fifth 10-machine 9-buffer line is $M_{21} - B_{21} - M_{22} - \cdots - B_{29} - M_{30}$.

For these five 10-machine lines, we use a revenue coefficient of 5,000. Let $\mathbf{N}_1^\star$, $\mathbf{N}_2^\star$, $\mathbf{N}_3^\star$, $\mathbf{N}_4^\star$, and $\mathbf{N}_5^\star$ denote the optimal buffer distributions of these five lines. Again, for illustration purpose, we keep the original buffer indices. The optimal buffer distributions of the five 10-machine lines are illustrate in Figure 8-4. It is helpful to indicate that $\mathbf{N}_1^\star$, $\mathbf{N}_2^\star$, $\mathbf{N}_3^\star$, $\mathbf{N}_4^\star$, and $\mathbf{N}_5^\star$ are the same since the five 10-machine lines

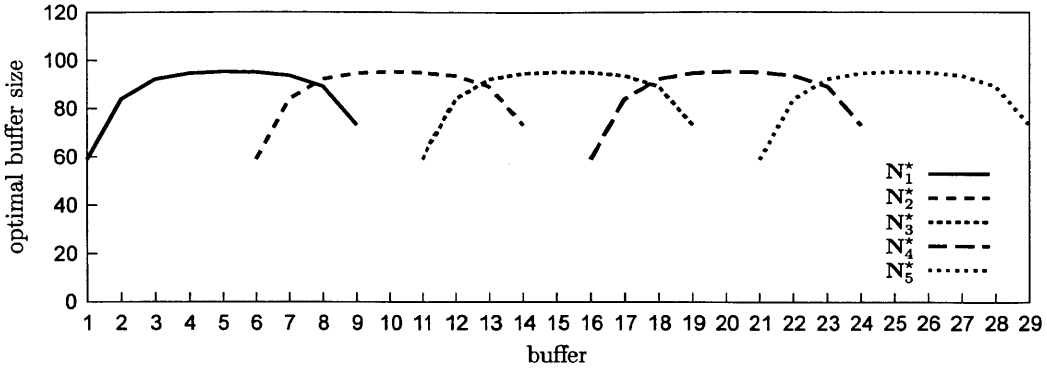


Figure 8-4: The optimal buffer distributions of the five 10-machine lines, a perfectly balanced line, Example 2

are identical. For values, see Table 8.1 since the parameters of any of these five 10-machine lines are the same as the parameters of any of those three 10-machine lines in the previous example. We take some results from each segment and construct the approximate solution $\mathbf{N}_{\text{seg}}$ for the original line who has 29 buffers as follows

$$\begin{aligned}\mathbf{N}_{\text{seg}} = \Big(&\mathbf{N}_1^\star(B_1), \mathbf{N}_1^\star(B_2), \cdots, \mathbf{N}_1^\star(B_7), \mathbf{N}_2^\star(B_8), \mathbf{N}_2^\star(B_9), \cdots, \mathbf{N}_2^\star(B_{12}), \\ &\mathbf{N}_3^\star(B_{13}), \mathbf{N}_3^\star(B_{14}), \cdots, \mathbf{N}_3^\star(B_{17}), \mathbf{N}_4^\star(B_{18}), \mathbf{N}_4^\star(B_{19}), \cdots, \mathbf{N}_4^\star(B_{22}), \\ &\mathbf{N}_5^\star(B_{23}), \mathbf{N}_5^\star(B_{24}), \cdots, \mathbf{N}_5^\star(B_{29})\Big).\end{aligned} \tag{8.4}$$

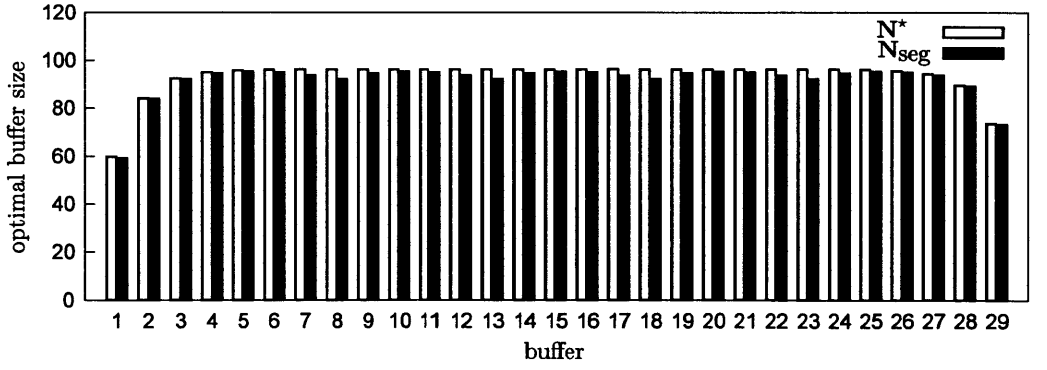


Figure 8-5: Comparison between $\mathbf{N}^\star$ and $\mathbf{N}_{\text{seg}}$, a perfectly balanced line, Example 2

For comparison, we optimize the original 30-machine 29-buffer line directly and display $\mathbf{N}^\star$ and $\mathbf{N}_{\text{seg}}$ in Figure 8-5. It demonstrates that $\mathbf{N}_{\text{seg}}$ is very close to $\mathbf{N}^\star$. With the segmentation method, the computer time to optimize the original 30-machine 29-buffer line is reduced dramatically from 2137.65 seconds (if optimizing the original line directly) to only 290.55 seconds. The production rate $P(\mathbf{N}^\star)$ is .8800 parts per time unit, while $P(\mathbf{N}_{\text{seg}})$ is slightly smaller at .8797 parts per time. The profit rate $J(\mathbf{N}^\star)$ is \$9237.20 per time unit, while $J(\mathbf{N}_{\text{seg}})$ is \$9276.35 per time unit.

The two examples in Sections 8.2.1 and 8.2.2 show how we can find an approximation of the optimal buffer distribution of the original line by the segmentation method for perfectly balanced long lines. In Section 8.3, we discuss unbalanced lines.

8.3 Qualitative Behavior of Unbalanced Lines

We study three examples of unbalanced long lines. The lines being considered are two 20-machine lines and one 30-machine line. As opposite to perfectly balanced lines, a unbalanced line does not have all identical machines or all identical buffers.

8.3.1 A 20-Machine Line Example

The 20-machine 19-buffer line has 20 machines falling into two groups and 19 identical buffers. The parameters for the first 10 machines are $r_i = .1$ and $p_i = .01$, $i = 1, \cdots, 10$, while the parameters for the second 10 machines are $r_i = .12$ and $p_i = .01$, $i = 11, \cdots, 20$. Buffer parameters are $b_i = c_i = 1, i = 1, \cdots, 19$. The revenue coefficient of $A = 10,000$. The target production rate is .88 parts per time unit. As before, we segment the original line into three 10-machine lines and optimize them separately and get $\mathbf{N}_1^\star$, $\mathbf{N}_2^\star$, and $\mathbf{N}_3^\star$ (see Table 8.2 and Figure 8-6). We construct $\mathbf{N}_{\text{seg}}$ according to Equation (8.3).

We also optimize the original 20-machine 19-buffer line directly and compare $\mathbf{N}^\star$ and $\mathbf{N}_{\text{seg}}$ in Figure 8-7. Figure 8-7 demonstrates that $\mathbf{N}_{\text{seg}}$ is also a very good approximation of $\mathbf{N}^\star$. The segmentation method reduces the computer time from 233.18 seconds to 101.06 seconds. The production rate $P(\mathbf{N}^\star)$ is .8800 parts per time

Table 8.2: The optimal buffer distribution of a unbalanced line, Example 1

The first line	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9
$\mathbf{N}_1^\star$	59.00	83.89	92.16	94.63	95.20	94.97	93.63	89.15	73.12
The second line	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}
$\mathbf{N}_2^\star$	59.27	83.32	89.84	86.65	68.19	53.35	47.72	43.47	34.05
The third line	B_{11}	B_{12}	B_{13}	B_{14}	B_{15}	B_{16}	B_{17}	B_{18}	B_{19}
$\mathbf{N}_3^\star$	25.76	39.39	44.17	45.71	46.12	46.01	45.20	42.54	33.62

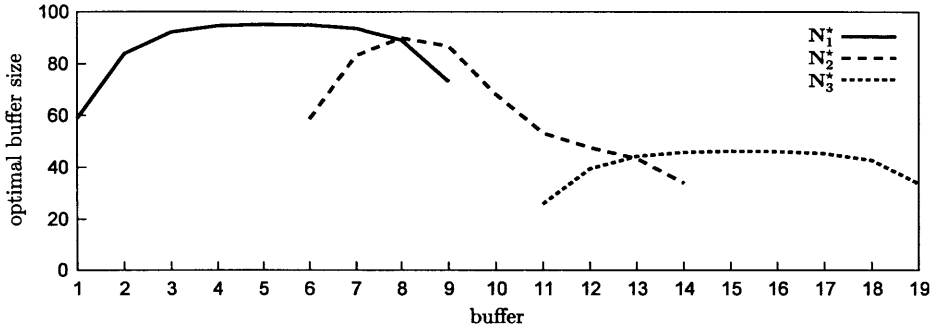


Figure 8-6: The optimal buffer distributions of the three 10-machine lines, a unbalanced line, Example 1

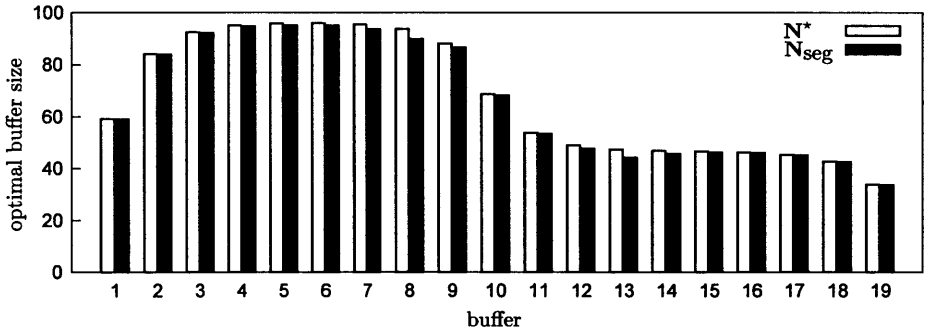


Figure 8-7: Comparison between $\mathbf{N}^\star$ and $\mathbf{N}_{\text{seg}}$, a unbalanced line, Example 1

unit, while $P(\mathbf{N}_{\text{seg}})$ is slightly smaller at .8798 parts per time. The profit rate $J(\mathbf{N}^\star)$ is \$6931.68 per time unit, while $J(\mathbf{N}_{\text{seg}})$ is \$6941.88 per time unit.

8.3.2 Another 20-Machine Line Example

In this example, we let machines be the same but buffers be very different in terms of their coefficients. The parameters for all machines are $r_i = .1$ and $p_i = .01$,

$i = 1, \cdots, 20$. Buffer parameters are listed in Table 8.3. The target production rate is .88 parts per time unit and the revenue coefficient is $A = 10,000$. As before, we segment the original line into three 10-machine 9-buffer lines in the way stated previously in Section 8.2.1, optimize them separately, and derive $\mathbf{N}_1^\star$, $\mathbf{N}_2^\star$, and $\mathbf{N}_3^\star$ (see Table 8.4 and Figure 8-8). We construct $\mathbf{N}_{\text{seg}}$ with $\mathbf{N}_1^\star$, $\mathbf{N}_2^\star$, and $\mathbf{N}_3^\star$ according to Equation (8.3).

Table 8.3: Buffer cost coefficients

b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
0.26	1.28	0.56	1.92	0.32	1.92	1.62	0.86	1.60	1.32
b_{11}	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}	b_{17}	b_{18}	b_{19}	
1.70	1.36	1.50	1.32	1.42	0.56	0.20	1.40	1.92	
c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
1.84	0.20	1.10	1.94	1.96	0.98	0.30	1.84	1.92	0.08
c_{11}	c_{12}	b_{13}	c_{14}	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}	
1.88	1.52	0.80	0.36	0.08	0.10	1.66	0.64	0.08	

Table 8.4: The optimal buffer distribution of a unbalanced line, Example 2

The first line	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9
$\mathbf{N}_1^\star$	60.73	83.69	109.69	67.27	129.34	85.21	96.43	94.83	66.47
The second line	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}
$\mathbf{N}_2^\star$	56.54	88.25	96.92	83.80	119.60	79.04	94.71	90.20	75.93
The third line	B_{11}	B_{12}	B_{13}	B_{14}	B_{15}	B_{16}	B_{17}	B_{18}	B_{19}
$\mathbf{N}_3^\star$	54.55	86.66	93.20	98.82	85.91	123.35	99.47	83.00	62.60

We optimize the original 20-machine 19-buffer line directly and compare $\mathbf{N}^\star$ and $\mathbf{N}_{\text{seg}}$ in Figure 8-9. Figure 8-9 demonstrates that for this unbalanced line where buffers are very different, $\mathbf{N}_{\text{seg}}$ approximates $\mathbf{N}^\star$ very well. The segmentation method reduces the computer time from 603.33 seconds to 199.80 seconds. The production rate $P(\mathbf{N}^\star)$ is .8800 parts per time unit, while $P(\mathbf{N}_{\text{seg}})$ is slightly smaller at .8798 parts per time. The profit rate $J(\mathbf{N}^\star)$ is \$5904.10 per time unit, while $J(\mathbf{N}_{\text{seg}})$ is \$5922.54 per time unit.

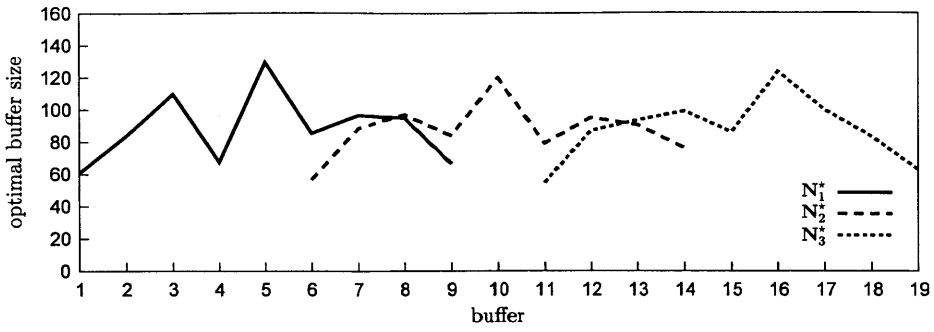


Figure 8-8: The optimal buffer distributions of the three 10-machine lines, a unbalanced line, Example 2

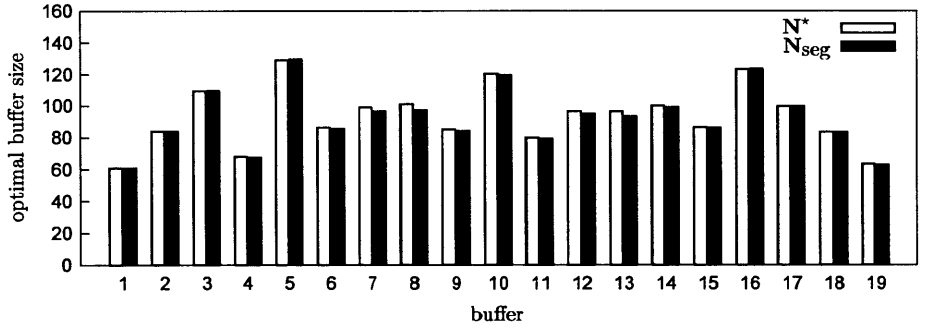


Figure 8-9: Comparison between $\mathbf{N}^\star$ and $\mathbf{N}_{\mathbf{seg}}$, a unbalanced line, Example 2

8.3.3 A 30-Machine Line Example

Finally, a unbalanced 30-machine 29-buffer line is studied. In particular, machine repair probabilities are very different (see Table 8.5) but buffers are identical. Failure probabilities for all machines are identical at $p_i = .01, i = 1, \cdots, 30$. The revenue coefficient of the line is $A = 15,000$. The target production rate is again .88 parts per time unit. We segment the original line into five 10-machine lines in the way stated previously in Section 8.2.2, optimize them, and derive $\mathbf{N}_1^\star$, $\mathbf{N}_2^\star$, $\mathbf{N}_3^\star$, $\mathbf{N}_4^\star$, and $\mathbf{N}_5^\star$ (see Table 8.6 and Figure 8-4).

We construct $\mathbf{N}_{\mathbf{seg}}$ with $\mathbf{N}_1^\star$, $\mathbf{N}_2^\star$, and $\mathbf{N}_3^\star$ according to Equation (8.4) and optimize the original 30-machine 29-buffer line directly. We compare $\mathbf{N}^\star$ and $\mathbf{N}_{\mathbf{seg}}$ in Figure 8-11, which demonstrates that for this unbalanced line where machines are very different, $\mathbf{N}_{\mathbf{seg}}$ approximates $\mathbf{N}^\star$ very well. The segmentation method dramatically

Table 8.5: Machine repair probabilities, 30-machine unbalanced line

r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}	r_{13}	r_{14}	r_{15}
.11	.09	.12	.14	.13	.11	.09	.088	.095	.106	.11	.11	.09	.09	.13
r_{16}	r_{17}	r_{18}	r_{19}	r_{20}	r_{21}	r_{22}	r_{23}	r_{24}	r_{25}	r_{26}	r_{27}	r_{28}	r_{29}	r_{30}
.12	.09	.098	.11	.105	.11	.12	.15	.14	.13	.094	.085	.11	.12	.104

Table 8.6: The optimal buffer distribution of a unbalanced line, Example 3

The first line	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9
$\mathbf{N}_1^\star$	52.04	66.85	47.48	43.50	55.15	90.80	141.05	133.06	82.40
The second line	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}
$\mathbf{N}_2^\star$	67.48	133.74	136.32	101.89	80.03	77.04	94.95	120.79	61.58
The third line	B_{11}	B_{12}	B_{13}	B_{14}	B_{15}	B_{16}	B_{17}	B_{18}	B_{19}
$\mathbf{N}_3^\star$	46.71	84.45	121.54	83.93	63.89	79.17	103.96	81.42	58.56
The fourth line	B_{16}	B_{17}	B_{18}	B_{19}	B_{20}	B_{21}	B_{22}	B_{23}	B_{24}
$\mathbf{N}_4^\star$	51.04	95.66	84.09	72.99	65.97	52.37	36.67	28.94	22.72
The fifth line	B_{21}	B_{22}	B_{23}	B_{24}	B_{25}	B_{26}	B_{27}	B_{28}	B_{29}
$\mathbf{N}_5^\star$	26.77	32.63	33.06	41.31	64.90	122.27	107.41	66.44	50.77

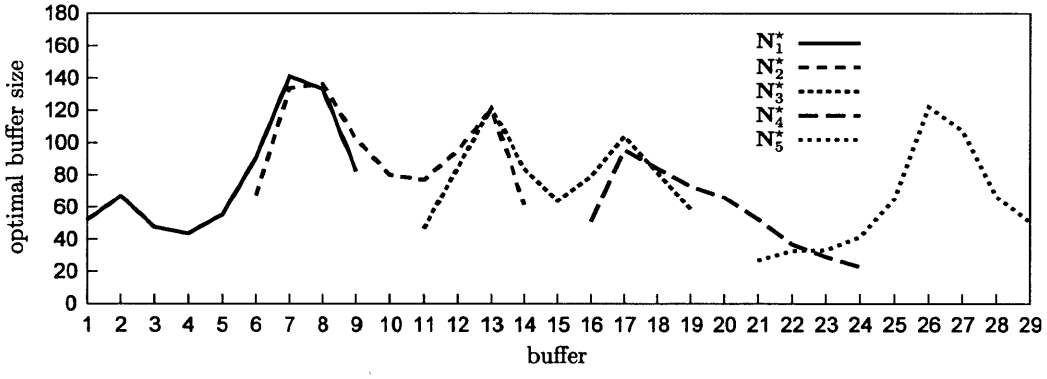


Figure 8-10: The optimal buffer distributions of the five 10-machine lines, a unbalanced line, Example 3

reduces the computer time from 2665.99 seconds to 751.29 seconds. The production rate $P(\mathbf{N}^\star)$ is .8800 parts per time unit, while $P(\mathbf{N}_{\text{seg}})$ is slightly smaller at .8798 parts per time. The profit rate $J(\mathbf{N}^\star)$ is \$9867.40 per time unit, while $J(\mathbf{N}_{\text{seg}})$ is \$9895.70 per time unit.

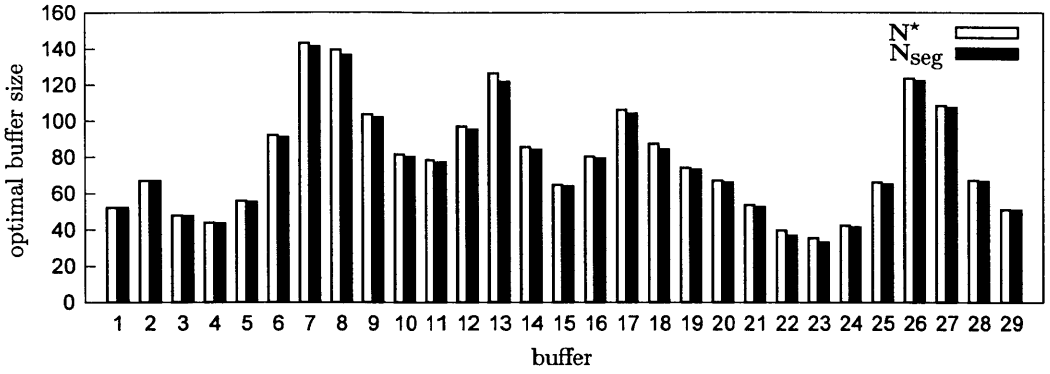


Figure 8-11: Comparison between $\mathbf{N}^\star$ and $\mathbf{N}_{\mathbf{seg}}$, a unbalanced line, Example 3

8.4 The Segmentation Method

According to all those examples in Sections 8.2 and 8.3 on both perfectly balanced lines and unbalanced lines, it can be seen that the buffer distribution derived by the segmentation method approximates the actual optimal buffer distribution of the original very well. However, we observe small discrepancy between $\mathbf{N}_{\mathbf{seg}}$ and $\mathbf{N}^\star$, especially in buffers shared by two adjacent segments. For instance, in the perfectly balanced 20-machine 19-buffer line example, both the first and the second segmented 10-machine lines contain Buffers B_6, B_7, B_8, and B_9. They are buffers shared by two adjacent segments. Figure 8-3 shows that errors between $\mathbf{N}_{\mathbf{seg}}$ and $\mathbf{N}^\star$ on B_7 and B_8 (as well as B_{12} and B_{13}) are the biggest compared to errors on other buffers.

These errors are due to the segmentation, which changes the variabilities at one or both ends of a line segment. In what follows, we first heuristically explain and then formally state the segmentation method. After that, we provide some preliminary study about two ways that can be used to improve the accuracy of the segmentation method.

8.4.1 Heuristic Explanation

We use the perfectly balanced 20-machine 19-buffer line of Section 8.2.1 to explain the segmentation method. The optimal buffer distributions of the original 20-machine line and the first 10-machine line segment $\mathbf{N}^\star$ and $\mathbf{N}_1^\star$ are illustrated in Figure 8-12.

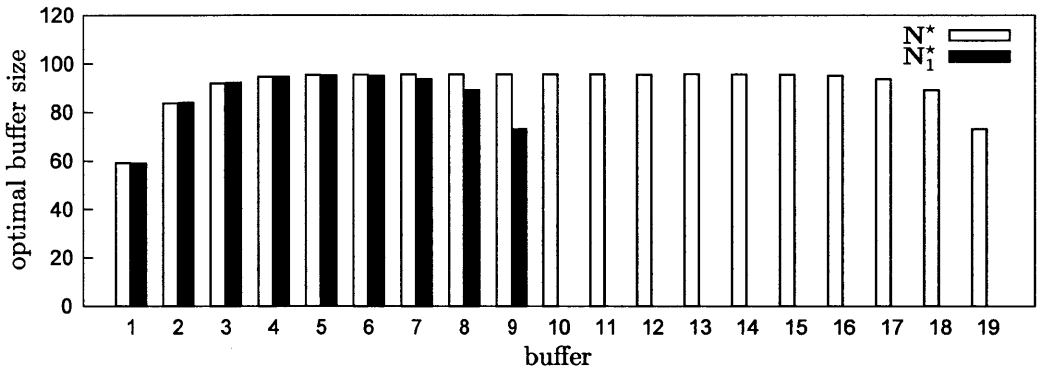


Figure 8-12: Comparison between $\mathbf{N}$ and $\mathbf{N}^\star$

Comparing the first nine components of $\mathbf{N}^\star$ with $\mathbf{N}_1^\star$, we observe increasing differences in Buffers B_6, B_7, B_8, and B_9. The optimal buffer sizes for those four buffers in $\mathbf{N}_1^\star$ are smaller than those in $\mathbf{N}^\star$. This can be explained by looking at the original 20-machine line and the first 10-machine line segment. In the original line, there are ten machines and ten buffers downstream of M_{10}, while in the first line segment there is nothing downstream of M_{10}. Therefore, in the original line M_{10} can be blocked if Buffer B_{10} gets full because downstream machine failures, while in the first line segment M_{10} will not be blocked. From the variability standpoint, the variability downstream of M_{10} in the original line is not 0, while the variability downstream of M_{10} in the first line segment is 0. As a result, $\mathbf{N}^\star(B_9)$ is larger than $\mathbf{N}_1^\star(B_9)$ to absorb the variability. Buffers B_8, B_7, and B_6 in the first line segment are also affected and reduced because of the zero variability downstream of M_{10}.

Figure 8-12 also indicates that there is no visible difference in Buffers B_1, B_2, B_3, B_4, and B_5 in the original line and the first line segment. This is because the benefits (in terms of reduced buffer space) brought by zero variability downstream of M_{10} in the first line segment have been consumed by B_6, B_7, B_8, and B_9. If we place an

operator at Buffer B_5 in either the original line or the first line segment, the material outflow behaviors the operator will observe will be almost the same statistically. Therefore, to achieve the required target production rate, the two lines require the same set of Buffers B_1, B_2, B_3, B_4, and B_5, or $\mathbf{N}^\star(B_i) = \mathbf{N}_1^\star(B_i), i = 1, 2, 3, 4, 5$.

The analysis above emphasizes the importance of the condition that the production rate constraint has to be active for the line segments. It is the active production rate constraint that ensures that the buffer sizes in $\mathbf{N}_i^\star$ that are not affected by the zero variability to be (approximately) the same as those in $\mathbf{N}^\star$. This also explains why we might not use the revenue coefficients A of the original line directly but to choose a smaller A to optimize the line segments. This is because it is possible that the A for the original line is too large for the line segments. According to Chapter 4, the production rate constraint will be inactive for the line segments if A is too large. The optimal sizes of buffers will be larger than necessary such that the production rate associated with the buffers is larger than $\hat{P}$ to enjoy the large revenue coefficient towards the goal of achieving a higher profit (rate). In this case, the segmentation method would not work. Consequently, we may need to choose the revenue coefficient when we optimize those line segments such that the production rate constraint is satisfied with equality in all segments.

In addition, the analysis related to the zero downstream variability reveals the source of inaccuracy of the segmented buffer distribution $\mathbf{N}_{\text{seg}}$ as compared to the $\mathbf{N}^\star$. To explain this, we refer to the benefits brought by the zero upstream or downstream variability to buffers at each end of a line segment as the *edge effect*. For example, the smaller B_6, B_7, B_8, and B_9 in the first line segment as compared to those in the original line in Figure 8-12 are due to the edge effect. Now we explain how the edge effect becomes a source of inaccuracy of $\mathbf{N}_{\text{seg}}$.

Consider the example of Section 8.2.1 again. Figure 8-13 shows the optimal buffer distributions for the original line and the first two segments. The bars show $\mathbf{N}_1^\star$ and $\mathbf{N}_2^\star$, while the curve shows $\mathbf{N}^\star$. We focus on the buffers that are common to the two segments (i.e., B_6, B_7, B_8, and B_9). We refer to them as buffers on the boundary of the two segments. The edge effects that we must correct can be found at the right

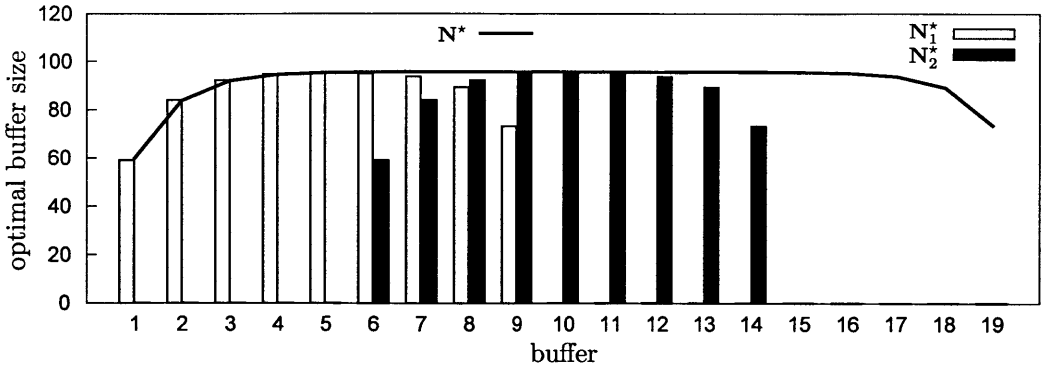


Figure 8-13: Explanation of the edge effect

end of the first segment and at both ends of the second segment. Due to the edge effect, the optimal sizes of B_6, B_7, B_8, and B_9 in both segments are smaller than their corresponding ones in the original line. Therefore, for these four buffers, we choose $\max\{\mathbf{N}_1^\star(B_i), \mathbf{N}_2^\star(B_i)\}$ as the size of $B_i, i = 6, 7, 8, 9$ in $\mathbf{N}_{\text{seg}}$. As a result, the first nine components of $\mathbf{N}_{\text{seg}}$ are $\mathbf{N}_1^\star(B_1)$, $\mathbf{N}_1^\star(B_2)$, $\mathbf{N}_1^\star(B_3)$, $\mathbf{N}_1^\star(B_4)$, $\mathbf{N}_1^\star(B_5)$, $\mathbf{N}_1^\star(B_6)$, $\mathbf{N}_1^\star(B_7)$, $\mathbf{N}_2^\star(B_8)$, and $\mathbf{N}_2^\star(B_9)$, and this setting mitigates the edge effort the most and gives the best approximation of the first nine components of $\mathbf{N}^\star$ with $\mathbf{N}_1^\star$ and $\mathbf{N}_2^\star$. However, we still observe differences between $\mathbf{N}_1^\star(B_7)$ and $\mathbf{N}^\star(B_7)$, as well as between $\mathbf{N}_2^\star(B_8)$ and $\mathbf{N}^\star(B_8)$. These differences result in the error in $\mathbf{N}_{\text{seg}}$.

The edge effect can be further mitigated by increasing the number of buffers common to two or more segments. To achieve this, we can

- increase the number of segments (while maintaining the length of each segment), or
- increase the length of each segment (while maintaining or slightly reducing the number of segments).

We provide a preliminary study about these approaches in Sections 8.4.3 and 8.4.4.

8.4.2 The Method

We state formally the segmentation method here. To solve Problem (8.1) for a K-machine $K-1$-buffer line[4], the segmentation method consists of the following steps.

0. Solve the corresponding unconstrained problem (8.2). If the solution of (8.2) satisfies the production rate constraint, then it is also the solution of (8.1) and we are done. If the solution of (8.2) does not satisfy the production rate constraint, then go to Step 1.

1. Choose the length of the line segments, say k. k should be large enough such that the number of buffers that are not apparently impacted by the edge effect in each segment is large enough to construct $\mathbf{N}_{\text{seg}}$. On the other hand, k should be kept small enough to keep the computer time spent on those segments small. Given the trade-off between the computer time and the accuracy of the segmentation method, we have not found a optimal way to choose k (but for a discussion about the trade-off, see Section 8.4.4). According to the examples studied in this chapter, 10 appears to be a good choice.

2. Choose the number of segments, say s, such that every two adjacent segments should share a set of buffers. The number of buffers contained by both segments should be large enough (e.g. four buffers in the examples discussed in Sections 8.2 and 8.3) to mitigate the edge effect. (Note that it is possible for buffers to be shared by more than two segments. In this case, more than two segments partially overlap with each other.)

3. Optimize each line segment and let $\mathbf{N}_i^\star$ be the optimal buffer distribution of the ith segment, $i = 1, 2, \cdots, s$.

4. Construct $\mathbf{N}_{\text{seg}}$ from $\mathbf{N}_i^\star$s. If a buffer is contained in more than one segment, the largest value of that buffer from any segment should be used in $\mathbf{N}_{\text{seg}}$ to

[4]By convention, the lower case k is used for a k-machine $k-1$-buffer line. However, we use the upper case K here for the original line, while saving the lower case k for the line segments.

eliminate the edge effect. For instance, if segments $i_1, i_2, \cdots, i_n$ contain B_m, then $\mathbf{N}_{\text{seg}}(B_m) = \max(\mathbf{N}^\star_{i_1}(B_m), \mathbf{N}^\star_{i_2}(B_m), \cdots, \mathbf{N}^\star_{i_n}(B_m))$.

5. Verify that $P(\mathbf{N}_{\text{seg}}) \approx \hat{P}$ and compute the profit of the line by $J(\mathbf{N}_{\text{seg}})$.

In the following two sections, we examine two strategies that can be considered to reduce the edge effect and to improve the accuracy of the segmentation method. For the following discussion, define $\mathbf{N}_{\text{seg}}(s,k)$ as the approximate buffer distribution for the original line resulted from the segmentation method with s k-machine line segments. In addition, define the error of the segmentation method as $e_{\text{seg}}(s,k) = \mathbf{N}_{\text{seg}}(s,k) - \mathbf{N}^\star$.

8.4.3 Discussion on the Number of the Line Segments

To show the effect of the number of line segments on the accuracy of the segmentation method, we reconsider those three 20-machine 19-buffer lines discussed in Sections 8.2.1, 8.3.1, and 8.3.2. In each of those three examples, we segment a 20-machine line with three 10-machine line segments. In this section, we segment the original line with five 10-machine lines.

Consider the line of Section 8.2.1 first. The configurations (in terms of the machines and buffers in the original line) of these five segments are:

1. the first segment: $M_1 - B_1 - M_2 - \cdots - B_9 - M_{10}$;
2. the second segment: $M_4 - B_4 - M_5 - \cdots - B_{12} - M_{13}$;
3. the third segment: $M_6 - B_6 - M_7 - \cdots - B_{14} - M_{15}$;
4. the four segment: $M_8 - B_8 - M_9 - \cdots - B_{16} - M_{17}$; and
5. the fifth segment: $M_{11} - B_{11} - M_{12} - \cdots - B_{19} - M_{20}$.

Let $\mathbf{N}^\star_1$, $\mathbf{N}^\star_2$, $\mathbf{N}^\star_3$, $\mathbf{N}^\star_4$, and $\mathbf{N}^\star_5$ be the optimal buffer distributions of these five segmented 10-machine lines. We construct $\mathbf{N}_{\text{seg}}$ according to Step 4 in Section 8.4.2

as follows:

$$\mathbf{N}_{\text{seg}} = \Big(\mathbf{N}_1^\star(B_1), \mathbf{N}_1^\star(B_2), \cdots, \mathbf{N}_1^\star(B_6), \mathbf{N}_2^\star(B_7), \mathbf{N}_2^\star(B_8), \mathbf{N}_2^\star(B_9), \mathbf{N}_3^\star(B_{10}), \mathbf{N}_3^\star(B_{11}), \mathbf{N}_4^\star(B_{12}), \mathbf{N}_4^\star(B_{13}), \mathbf{N}_5^\star(B_{14}), \cdots, \mathbf{N}_5^\star(B_{19})\Big). \tag{8.5}$$

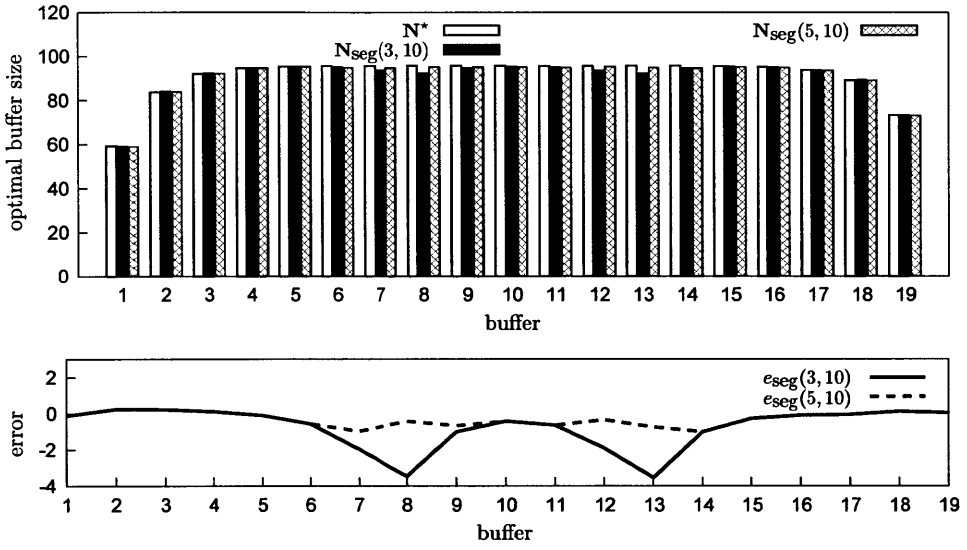


Figure 8-14: Effect of the number of the segments on the accuracy of the segmentation method, Example 1

To study the effect of the number of the segments on the accuracy of the segmentation method, we compare the optimal distribution of the original line $\mathbf{N}^\star$, the solution constructed from three 10-machine line segments $\mathbf{N}_{\text{seg}}(3, 10)$, and the solution constructed from five 10-machine line segments $\mathbf{N}_{\text{seg}}(5, 10)$ (see Figure 8-14). The errors of the two constructed solutions are also included in Figure 8-14. Figure 8-14 reveals that the inaccuracy due to the edge effect is much more obvious when we approximate $\mathbf{N}^\star$ with three 10-machine lines. However, with five segments, the solution from the segmentation method is more accurate.

Table 8.7 compares the three alternatives that can be used to find the optimal buffer distribution for the original line: optimizing the original line by the original method of Chapter 4, segmenting the original line into three 10-machine lines, and segmenting the line into five 10-machine lines. We see that segmenting the line

Table 8.7: Result summary, effect of the number of the line segments, Example 1

	$P(\mathbf{N})$	$J(\mathbf{N})$	compter time (sec.)	max. buffer difference
$\mathbf{N}^{\star}$	.8800	6259.11	483.74	—
$\mathbf{N}_{\text{seg}}(3,10)$	.8798	6270.34	125.00	−3.56
$\mathbf{N}_{\text{seg}}(5,10)$	.8799	6263.13	164.86	−1.02

with five 10-machine lines provides a more accurate approximation of the optimal buffer distribution at a cost of longer computer time because more segments are optimized. As there are more segments, the number of buffers shared by any two adjacent segments becomes larger, and therefore more buffers affected by the edge effect will not be chosen to construct $\mathbf{N}_{\text{seg}}$. This mitigates the edge effect. With the segmentation method, we reduce the computer time dramatically. Similarly results can be observed on the other two unbalanced 20-machine lines of Sections 8.3.1 and 8.3.2 (see Figures 8-15 and 8-16). The comparison is summarized in Tables 8.8 and 8.9 as well.

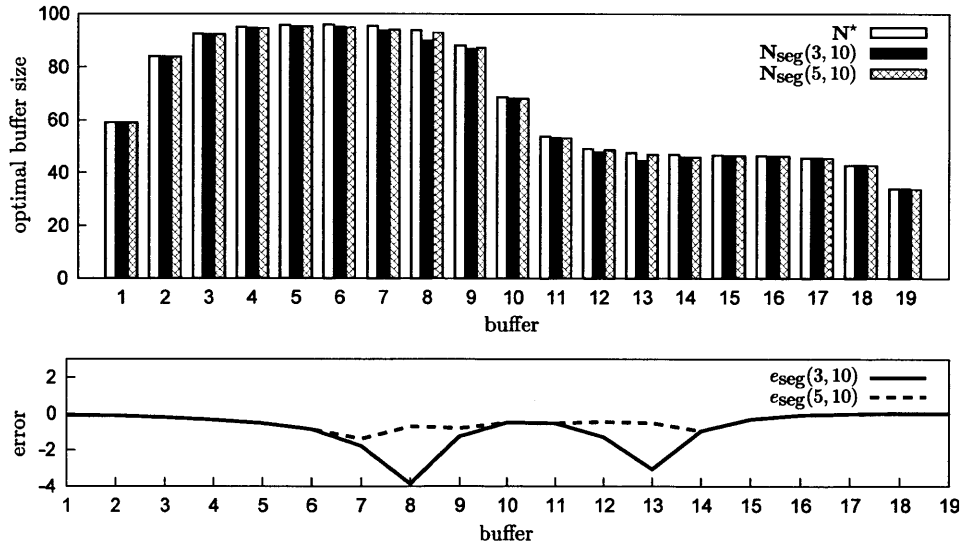


Figure 8-15: Effect of the number of the segments on the accuracy of the segmentation method, Example 2

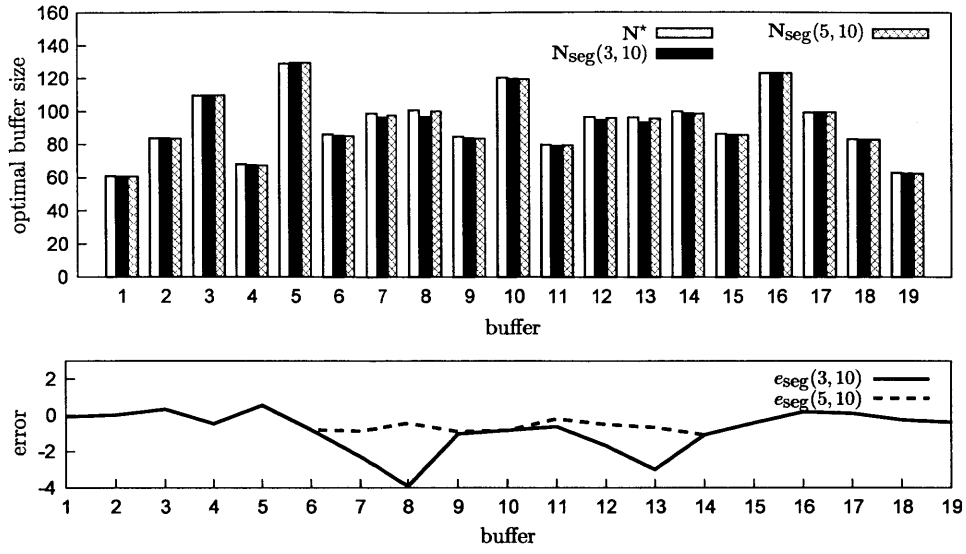


Figure 8-16: Effect of the number of the segments on the accuracy of the segmentation method, Example 3

Table 8.8: Result summary, effect of the number of line segments, Example 2

	$P(\mathbf{N})$	$J(\mathbf{N})$	compter time (sec.)	max. buffer difference
$\mathbf{N}^\star$	.8800	6931.68	233.18	—
$\mathbf{N}_{\text{seg}}(3,10)$	.8798	6941.88	101.06	−3.86
$\mathbf{N}_{\text{seg}}(5,10)$	.8799	6938.54	120.05	−1.39

Table 8.9: Result summary, effect of the number of line segments, Example 3

	$P(\mathbf{N})$	$J(\mathbf{N})$	compter time (sec.)	max. buffer difference
$\mathbf{N}^\star$	.8800	5904.10	603.33	—
$\mathbf{N}_{\text{seg}}(3,10)$	.8798	5922.54	199.80	−3.91
$\mathbf{N}_{\text{seg}}(5,10)$	.8799	5912.86	274.91	−1.08

8.4.4 Discussion on the Length of the Line Segments

In this section, with the three 20-machine 19-buffer lines of Sections 8.2.1, 8.3.1, and 8.3.2 as well as the two 30-machine 29-buffer lines of Sections 8.2.2 and 8.3.3, we study the effect of the length of the line segments. Recall that we have discussed

the segmentation method with 10-machine lines. In addition to that, for 20-machine lines, we consider 15-machine line segments; while for 30-machine lines, we consider both 15-machine line segments and 20-machine line segments.

Consider the perfectly balanced 20-machine 19-buffer line of Section 8.2.1 first. The configurations of the two 15-machine line segments are:

1. the first segment: $M_1 - B_1 - M_2 - \cdots - B_{14} - M_{15}$; and

2. the second segment: $M_6 - B_6 - M_7 - \cdots - B_{19} - M_{20}$.

Let $\mathbf{N}_1^\star$ and $\mathbf{N}_2^\star$ be the optimal buffer distributions of the two 15-machine line segments. We construct $\mathbf{N}_{\text{seg}}$ as follows:

$$\mathbf{N}_{\text{seg}} = \left(\mathbf{N}_1^\star(B_1), \mathbf{N}_1^\star(B_2), \cdots, \mathbf{N}_1^\star(B_{10}), \mathbf{N}_2^\star(B_{11}), \mathbf{N}_2^\star(B_{12}), \cdots, \mathbf{N}_2^\star(B_{19})\right). \quad (8.6)$$

As a reminder, we let $\mathbf{N}_{\text{seg}}(s,k)$ be the constructed buffer distribution from the segmentation method with s k-machine line segments, and let $e_{\text{seg}}(s,k) = \mathbf{N}_{\text{seg}}(s,k) - \mathbf{N}^\star$ be the error. We plot $\mathbf{N}^\star$, $\mathbf{N}_{\text{seg}}(3,10)$, and $\mathbf{N}_{\text{seg}}(2,15)$ as well as the errors of the two constructed solutions in Figure 8-17. Figure 8-17 reveals that with two 15-machine line segments, the solution from the segmentation method improves obviously and the error in buffer is very close to zero. Table 8.10 compares these three alternatives.

Table 8.10: Result summary, effect of the length of line segments, Example 1

	$P(\mathbf{N})$	$J(\mathbf{N})$	compter time (sec.)	max. buffer difference
$\mathbf{N}^\star$	.8800	6259.11	483.74	—
$\mathbf{N}_{\text{seg}}(3,10)$	.8798	6270.34	125.00	−3.56
$\mathbf{N}_{\text{seg}}(2,15)$	.8800	6257.03	340.71	0.19

Segmenting the line with two 15-machine line segments provides a more accurate approximation of the optimal distribution at a cost of larger computer time. This is because the time to compute $\mathbf{N}_{\text{seg}}(s,k)$ is roughly linear in s and exponential

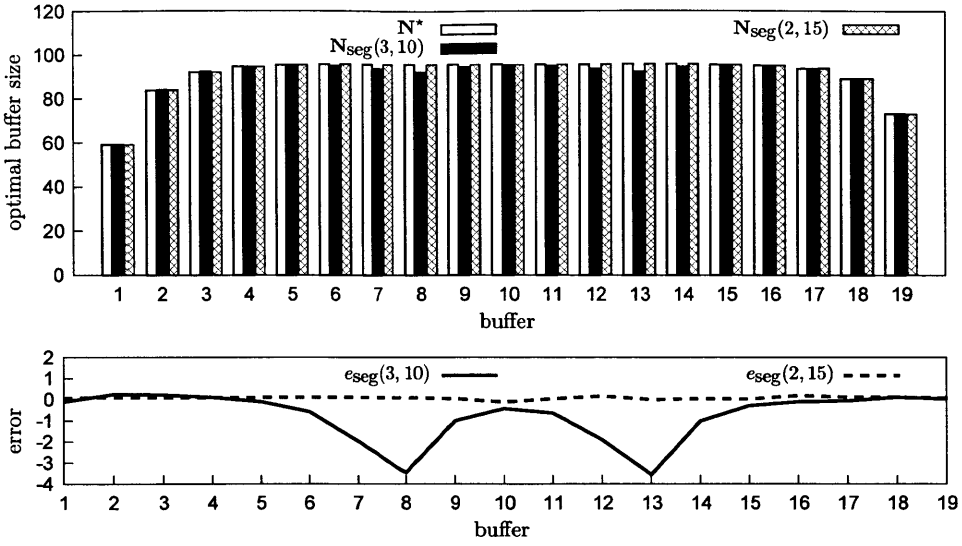


Figure 8-17: Effect of the length of line segments on the accuracy of the segmentation method, Example 1

in k. Therefore, it takes more time to find $\mathbf{N}_{\text{seg}}(2, 15)$ as longer line segments are optimized even though the number of segments is reduced. As the segments get longer, the number of buffers shared by any two adjacent segments becomes larger, and therefore more buffers affected by the edge effect will not be chosen to construct $\mathbf{N}_{\text{seg}}$. This mitigates the edge effect. Furthermore, we see that even with two 15-machine line segments, we still reduce the computer time from 483.74 seconds to 340.71 seconds, although this is not very drastic. However the solution from the segmentation method is very accurate. Similarly results can be observed on the other two unbalanced lines of Sections 8.3.1 and 8.3.2 (see Figures 8-18 and 8-19, as well as Tables 8.11 and 8.12.)

Table 8.11: Result summary, effect of the length of line segments, Example 2

	$P(\mathbf{N})$	$J(\mathbf{N})$	compter time (sec.)	max. buffer difference
$\mathbf{N}^\star$	.8800	6931.68	233.18	—
$\mathbf{N}_{\text{seg}}(3, 10)$	.8798	6941.88	101.06	−3.86
$\mathbf{N}_{\text{seg}}(2, 15)$	.8800	6936.24	168.31	−0.64

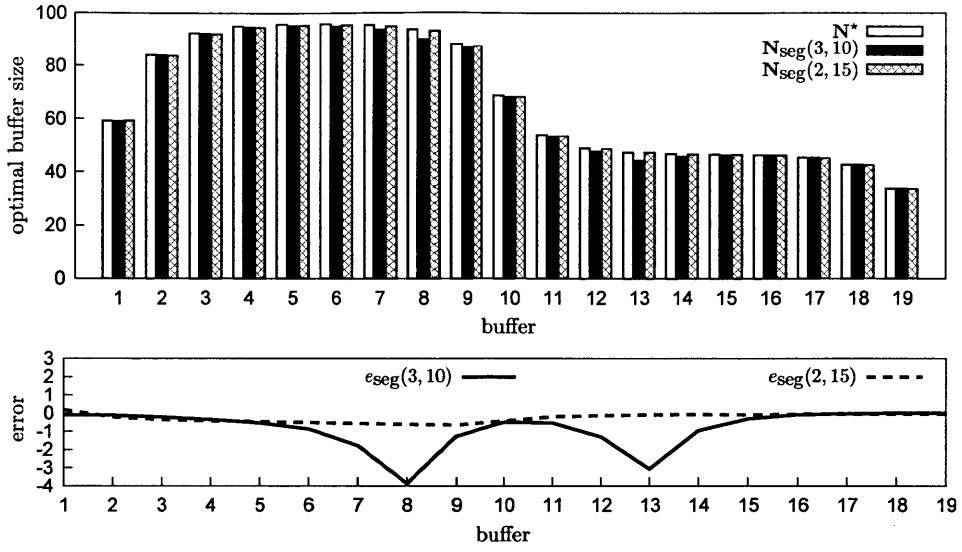


Figure 8-18: Effect of the length of line segments on the accuracy of the segmentation method, Example 2

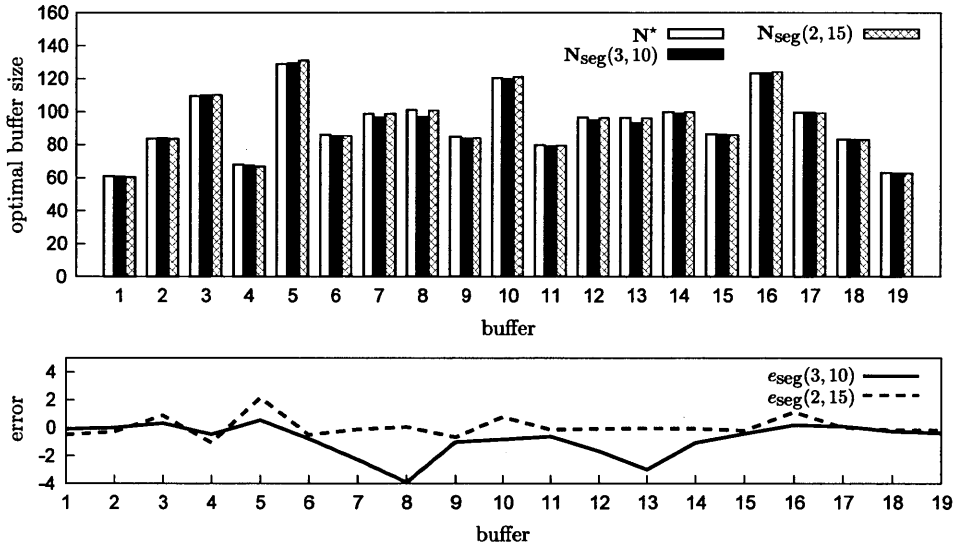


Figure 8-19: Effect of the length of line segments on the accuracy of the segmentation method, Example 3

Next, we study the two 30-machine lines of Sections 8.2.2 and 8.3.3. In particular, we study the segmentation method with three 15-machine lines as well as two 20-machine lines.

First, we segment an original 30-machine line with three 15-machine lines:

Table 8.12: Result summary, effect of the length of line segments, Example 3

	$P(\mathbf{N})$	$J(\mathbf{N})$	compter time (sec.)	max. buffer difference
$\mathbf{N}^\star$	.8800	5904.10	603.33	—
$\mathbf{N}_{\text{seg}}(3,10)$	.8798	5922.54	199.80	−3.91
$\mathbf{N}_{\text{seg}}(2,15)$	.8800	5909.90	488.36	2.14

1. the first line: $M_1 - B_1 - M_2 - \cdots - B_{14} - M_{15}$;

2. the second line: $M_8 - B_8 - M_9 - \cdots - B_{21} - M_{22}$; and

3. the third line: $M_{16} - B_{16} - M_{17} - \cdots - B_{29} - M_{30}$.

Alternatively, we segment the original line with two 20-machine lines. The two segments are

1. the first line: $M_1 - B_1 - M_2 - \cdots - B_{19} - M_{20}$; and

2. the third line: $M_{11} - B_{11} - M_{12} - \cdots - B_{29} - M_{30}$.

We compare $\mathbf{N}^\star$, $\mathbf{N}_{\text{seg}}(5,10)$, $\mathbf{N}_{\text{seg}}(3,15)$, and $\mathbf{N}_{\text{seg}}(2,20)$. The results are shown in Figures 8-20 and 8-21. It can be seen that the accuracy of the segmentation method with 15-machine segments and 20-machine segments is better than the accuracy with 10-machine segments. The comparison is summarized in Tables 8.13 and 8.14 as well. The segmented solution $\mathbf{N}_{\text{seg}}(3,15)$ is accurate enough as compared to either the actual solution $\mathbf{N}^\star$ or the segmented solution $\mathbf{N}_{\text{seg}}(2,20)$. On the other hand, since the computer time increases exponentially with the length of the segment, using a segment with 20 machines seems not to be a good idea from a standpoint of computer time saving. From the discussion in this section, it appears that segments of 10 or 15 machines can be considered good choices.

8.4.5 Proposed Improvement Strategies

We discuss preliminarily, in this section, two strategies to reduce the optimization inaccuracy of the segmentation method brought by the edge effect. To further inves-

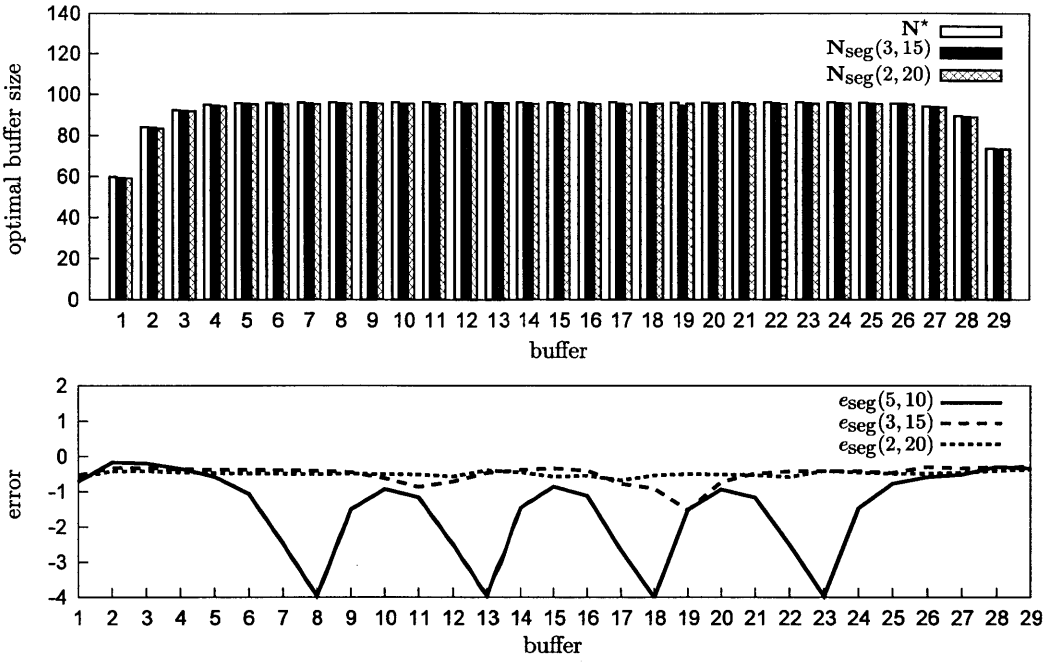


Figure 8-20: Comparsion of different lengths of line segments, Example 4

Table 8.13: Result summary, effect of the length of line segments, Example 4

	$P(\mathbf{N})$	$J(\mathbf{N})$	compter time (sec.)	max. buffer difference
$\mathbf{N}^*$	.8800	9237.20	2137.65	—
$\mathbf{N}_{\text{seg}}(5,10)$	.8797	9276.35	290.55	−4.01
$\mathbf{N}_{\text{seg}}(3,15)$	.8800	9254.66	569.29	−1.53
$\mathbf{N}_{\text{seg}}(2,20)$	.8799	9256.98	1075.72	−0.68

Table 8.14: Result summary, effect of the length of line segments, Example 5

	$P(\mathbf{N})$	$J(\mathbf{N})$	compter time (sec.)	max. buffer difference
$\mathbf{N}^*$	.8800	9867.41	2665.99	—
$\mathbf{N}_{\text{seg}}(5,10)$	.8798	9895.70	751.29	−4.72
$\mathbf{N}_{\text{seg}}(3,15)$	.8800	9869.05	1244.60	−1.01
$\mathbf{N}_{\text{seg}}(2,20)$	.8800	9872.28	1802.52	−1.42

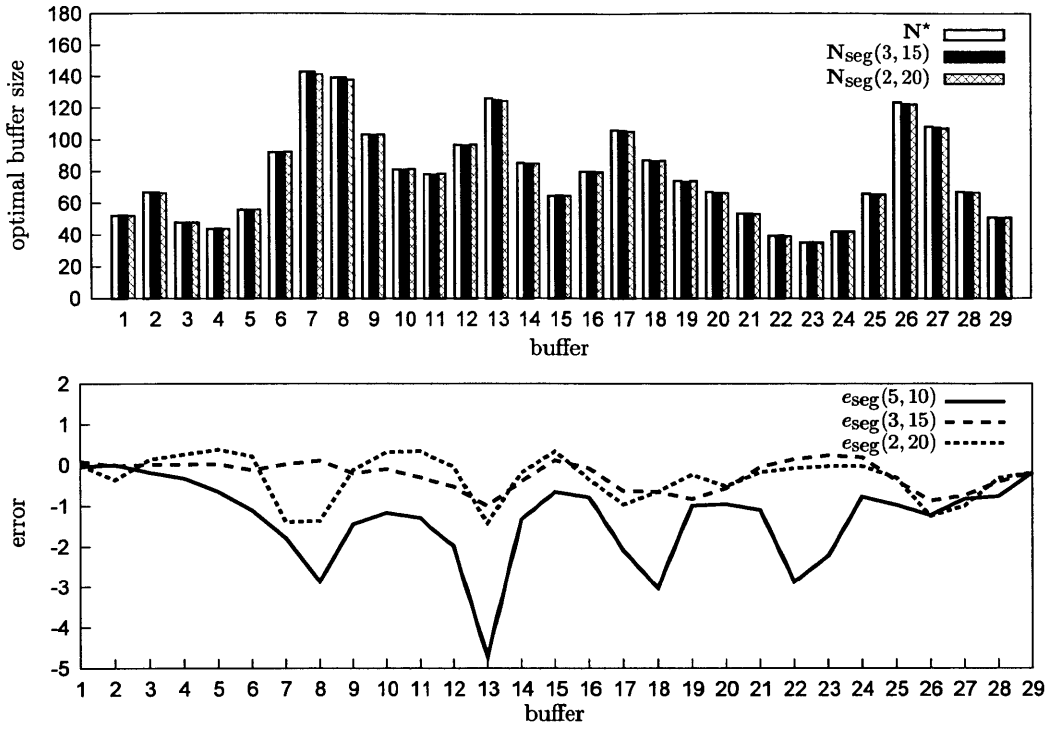


Figure 8-21: Comparsion of different lengths of line segments, Example 5

tigate these two strategies in the future, it is helpful to study the length limit of a line segment (Anthony 2011). For example, for an original 20-machine line, we can start the segmentation method with 20 five-machine four-buffer line segments, and increase the length of each segment (while reducing the number of segments when necessary) gradually. This will allow us to further study the trade-off between the computer time and the optimization accuracy of the segmentation method.

In addition, we notice that Step 0 of the segmentation method (see Section 8.4.2) requires solving the corresponding unconstrained problem first. The computer time of solving the unconstrained problem when the original line is long can contribute a big portion of the total computer time of the segmentation method. For instance, for the 30-machine line of Section 8.2.2, the computer time for solving the unconstrained 30-machine line problem is 197.82 seconds. The total computer time of the segmentation method with three 10-machine line segments is 290.55 seconds. The time spent on solving the unconstrained 30-machine line problem accounts for over

2/3 of the total computer time. Therefore, it is desirable if Step 0 can be skipped in the segmentation method (Graves 2011). The segmentation method without Step 0 finds the approximate optimal buffer distribution with the assumption that the production rate constraint is active for the original line. Therefore, if we get rid of Step 0, we need to find other ways to check if the original problem has this property or not. This can be a future research direction of the segmentation method.

8.5 Numerical Experiments

Finally, we provide more numerical experiments to show the efficiency and accuracy of the segmentation method. First, numerical results for 100 randomly generated 20-machine 19-buffer lines are provided. After that, we consider some cases of extremely long lines, such as lines that have 50 machines, 60 machines, and 70 machines.

8.5.1 More Numerical Examples

We study 100 20-machine 19-buffer lines that are randomly generated according to the method of Gershwin (2011). Both machine parameters and buffer cost coefficients are randomly chosen. The isolated production rate of each machine is greater than .89 parts per time unit. The target production rate is .87 parts per time unit, which proves to be active in all cases when the revenue coefficient is $A = 10,000$. For each 20-machine 19-buffer line, we segment it into three 10-machine 9-buffer lines starting with Machines M_1, M_6, and M_{11} of the original line, respectively.

Figure 8-22 displays the computer time for optimizing the 100 lines by the direct method of conducting the one-dimensional search over $A' > A$ (labeled Time, direct) with that for optimizing them via the segmentation method (labeled Time, segmentation). We rank these 100 cases in a descending order of the computer time for optimizing them by the direct method. The figure also shows that the average computer time for optimizing these 100 lines by the direct method is 696.07 seconds, while the average computer time by the segmentation method is only 286.69 seconds. The three types of errors considered are the profit error (J_{err}), the production rate

error (P_{err}), and the maximum buffer size error (N_{err}):

$$J_{\text{err}} = \left| \frac{J(\mathbf{N}^\star) - J(\mathbf{N}_{\text{seg}})}{J(\mathbf{N}^\star)} \right| \times 100\%,$$

$$P_{\text{err}} = \left| \frac{P(\mathbf{N}^\star) - P(\mathbf{N}_{\text{seg}})}{P(\mathbf{N}^\star)} \right| \times 100\%,$$

and

$$N_{\text{err}} = \max_{i=1,\cdots,19} \left\{ \left| \frac{\mathbf{N}^\star(B_i) - \mathbf{N}_{\text{seg}}(B_i)}{\mathbf{N}^\star(B_i)} \right| \times 100\% \right\},$$

and they are shown in Figure 8-23 for these 100 lines.

Figure 8-23 reveals that the average profit error of the 100 lines is .05%, the average production rate error is .03%, and the average maximum buffer size error is 5.38%. These results demonstrate the accuracy and efficiency of the segmentation method.

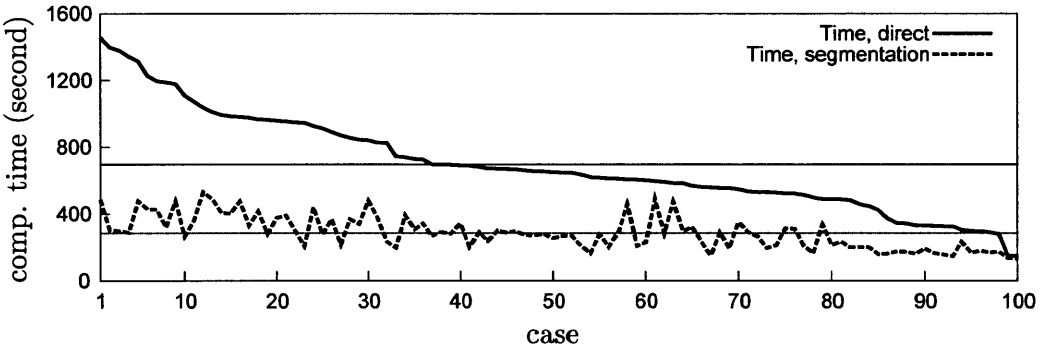


Figure 8-22: Comparison of the computer times for 100 randomly generated 20-machine 19-buffer lines

8.5.2 Extremely Long Lines

Finally, we study three extremely lone lines: a 50-machine 49-buffer line, a 60-machine 59-buffer line, and a 70-machine 69-buffer line. The purpose of these three examples is to show how much computer time can be saved with the segmentation method. However, we have to mention that such long lines (in terms of number of buffers) are not common in reality. In particular, all these lines are perfectly balanced lines.

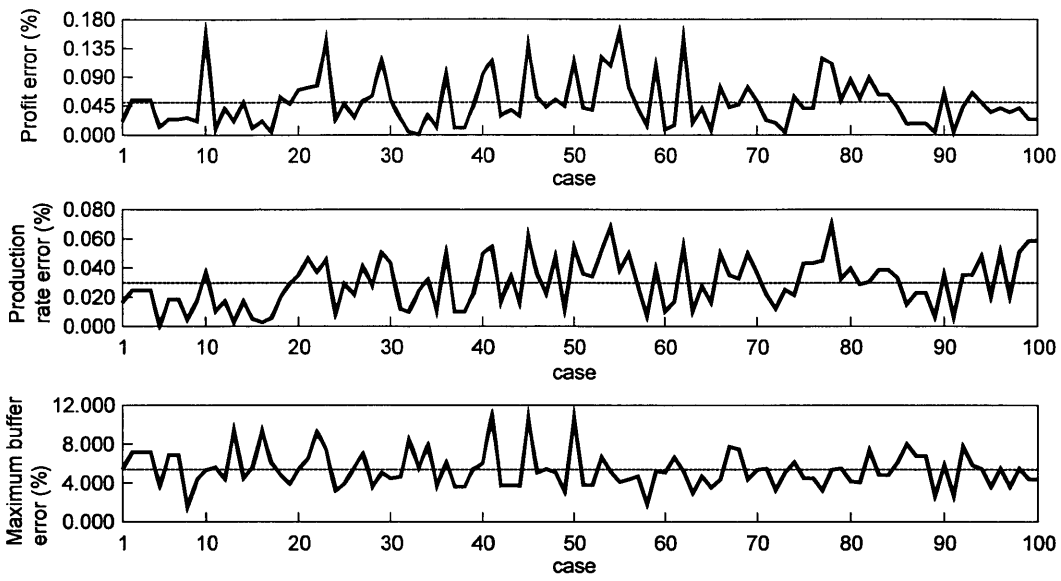


Figure 8-23: Production rate errors, profit errors, and maximum buffer errors of the segmentation method for 100 randomly generated 20-machine 19-buffer lines

All machines have parameters $r_i = .1$ and $p_i = .01$. All buffers have parameters $b_i = c_i = 1$. The revenue coefficients for these three examples are $A = 25000$, 30000, and 35000, respectively. The target production rate is $\hat{P} = .88$ for all three examples.

For the 50-machine 49-buffer line, we segment it into 14 10-machine 9-buffer line segments. They start with Machines M_1, M_4, M_7, $\cdots$, M_{37}, and M_{41} of the original line, respectively. We compare $\mathbf{N}_{\text{seg}}$ and $\mathbf{N}^\star$ in Figure 8-24, which shows that $\mathbf{N}_{\text{seg}}$ is a good approximation of $\mathbf{N}^\star$. The key measures of this example are summarized in Table 8.15. We see that with the segmentation method, the computer time for optimizing the 50-machine 49-buffer line is reduced dramatically from 8967.80 seconds to 1316.51 seconds. If we further realize that the 14 10-machine 9-buffer segments are indeed identical and we need only optimize one of them rather than all of them, we can even reduce the computer time to merely 895.01 seconds.

Similarly, we segment the 60-machine 59-buffer line with 18 10-machine 9-buffer line segments, and the 70-machine 69-buffer line with 21 10-machine 9-buffer line segments. The results for these two lines are summarized in Figures 8-25 and 8-26, as well as Tables 8.16 and 8.17. With the segmentation method, we reduce the computer time for optimizing the 60-machine 59-buffer line from 17908.25 seconds to 1501.88

seconds. Similarly, for the 70-machine 69-buffer line, the computer time is reduced from 20753.75 seconds to 1839.43 seconds.

Table 8.15: Result summary for a 50-machine 49-buffer line

	$P(\mathbf{N})$	$J(\mathbf{N})$	computer time (sec.)	max. buffer difference
$\mathbf{N}^\star$	.8800	15218.95	8967.80	—
$\mathbf{N}_{\text{seg}}$	.8799	15261.12	1316.51	−2.34

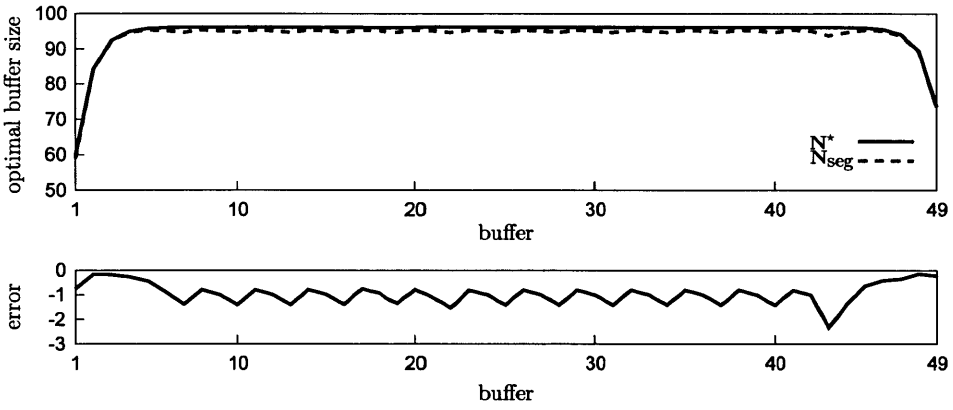


Figure 8-24: The segmentation method for a 50-machine 49-buffer line

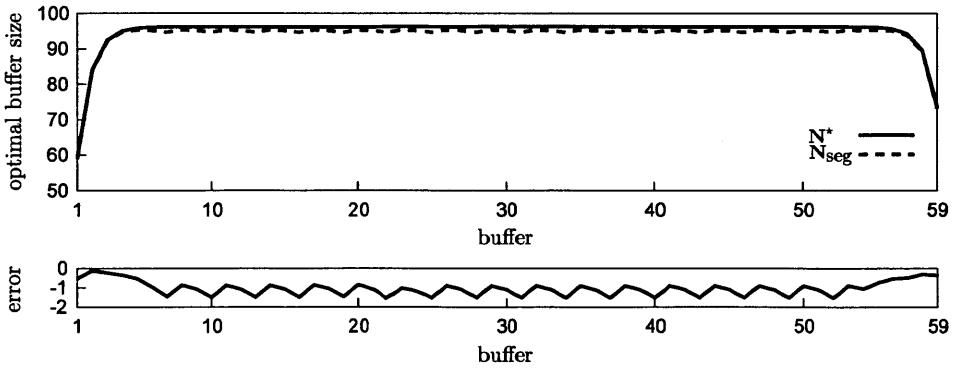


Figure 8-25: The segmentation method for a 60-machine 59-buffer line

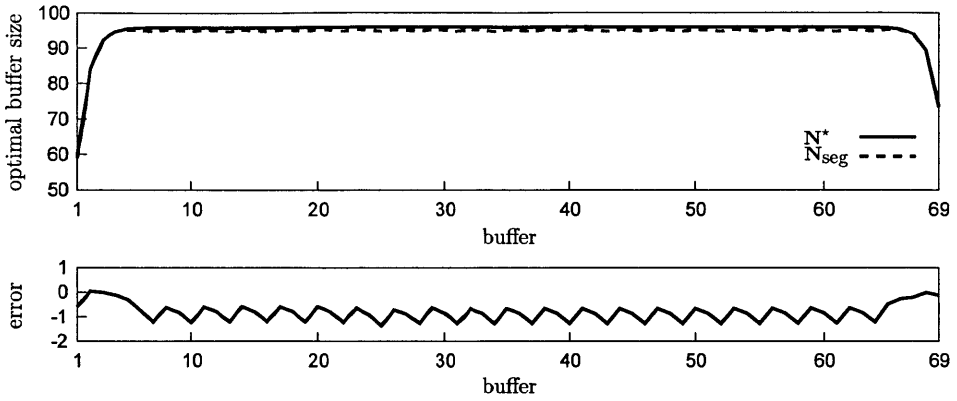


Figure 8-26: The segmentation method for a 70-machine 69-buffer line

Table 8.16: Result summary for a 60-machine 59-buffer line

	$P(\mathbf{N})$	$J(\mathbf{N})$	computer time (sec.)	max. buffer difference
$\mathbf{N}^*$	.8800	18209.24	17908.25	—
$\mathbf{N}_{\mathbf{seg}}$	.8799	18260.85	1501.88	−1.57

Table 8.17: Result summary for a 70-machine 69-buffer line

	$P(\mathbf{N})$	$J(\mathbf{N})$	computer time (sec.)	max. buffer difference
$\mathbf{N}^*$	.8800	21217.08	20753.75	—
$\mathbf{N}_{\mathbf{seg}}$	.8799	21260.15	1839.43	−1.36

8.6 Summary

In this chapter, we study the segmentation method for long line optimization. Instead of optimizing the original long line, the segmentation method divides it into several short line segments, optimizes these short line segments separately, and combines the optimal buffer distributions to find an approximate optimal buffer distribution of the original line.

With demonstrative numerical experiments, we first show that the segmentation method works well for both perfectly balanced lines and unbalanced lines. After ex-

plaining the method in Section 8.4.1, two strategies that improve the accuracy of the segmentation method are studied. Based on the discussion, the accuracy of the segmentation method can be improved by increasing the number of line segments or by increasing the length of each line segment. Both methods improve the accuracy at a cost of longer computer time. With the segmentation method, the total computer time required to find the optimal solution for the original long line can be reduced dramatically. Eventually, 100 numerical experiments are provided to show the accuracy and efficiency of the segmentation method.

Chapter 9

The Additive Property in Long Line Optimization

9.1 Overview

In this chapter, we make use of the algorithm of Chapter 4 for production lines to study an additive property in long line optimization. The property shows that the effect of a set of local bottleneck machines on the optimal buffer distribution, which maximizes the profit of the production line defined by Equation (4.1) subject to a production rate constraint, is approximately the same as the sum of the effects of each local bottleneck machine by itself. A similar property is observed with a mixture of local bottleneck machine and local anti-bottleneck machines[1]. The additive property provides valuable insight in the design and optimization of long lines.

The goal of this chapter is to report on a phenomenon we have observed. We report only on numerical experiments and we describe a preliminary hypothesis to explain it. We do not provide precise conditions under which it will occur, analytical bounds on its accuracy, or potential computational benefits. We also do not provide an algorithm for using it in system design.

This chapter is organized as follows. We first demonstrate the qualitative behavior of the additive property with examples in Section 9.2. The heuristic explanations

[1]The definition of a local anti-bottleneck machine is provided in Section 9.2.1.

about the additive property are provided in Section 9.3. Some cases where the additive property is less accurate are also examined in Section 9.4. We summarize this chapter in Section 9.5.

9.2 Qualitative Demonstration of the Additive Property

We use a 30-machine 29-buffer line to illustrate the additive property in long line optimization. In particular, we start with a base line that contains 30 identical machines and 29 identical buffers (in terms of the buffer space cost and average inventory cost). The optimal buffer allocation that maximizes the profit of the base line is first determined by the algorithm presented in Chapter 4. Then, we vary the parameters of a set of machines and show the additive property.

9.2.1 The Base Line – a 30-Machine Line with Identical Machines and Identical Buffers

In the base line, the parameters of Machine M_i are $r_i = .1$ and $p_i = .01$, $i = 1, \cdots, 30$. Therefore, the isolated production rate of M_i is $P_i = .909$. The buffer space cost coefficient and average inventory cost coefficient of Buffer B_i are $b_i = c_i = \$1$/part/time unit, $i = 1, \cdots, 29$. The revenue coefficient $A = \$15,000$/part and the target production rate is .88 parts per time unit. For convenience in the discussion about the additive property in the remaining of this chapter, we let the base line be called line $L^\star$, and the optimal buffer distribution for the base line $L^\star$ be called $\mathbf{N}^\star$. $\mathbf{N}^\star$ is illustrated in Figure 9-1.

In what follows, we study several cases by replacing machines of the base line $L^\star$ with local bottleneck and/or local anti-bottleneck machines. A machine M_i is called an *anti-bottleneck machine* if its isolated production rate P_i is higher than those original machines in the base line whose isolated production rate is .909. Similarly, M_i is called a *local* anti-bottleneck machine if its isolated production rate is higher than

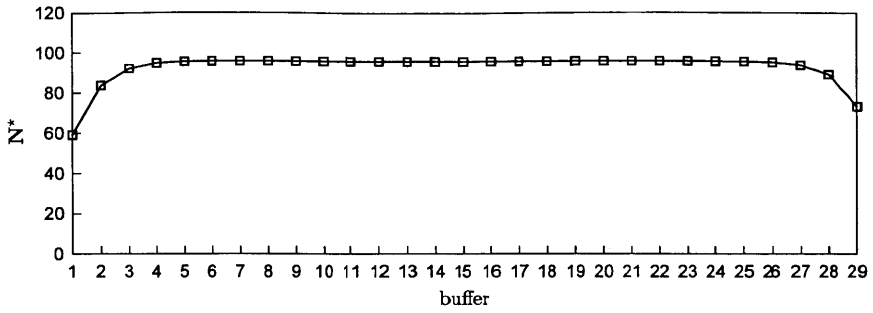


Figure 9-1: The optimal buffer allocation for the base line

those machines in its neighborhood. For simplicity, we refer to local bottleneck machines and local anti-bottleneck machines as bottleneck machines and anti-bottleneck machines, respectively, for the rest of this chapter. The new line with Machines M_{i_1}, $M_{i_2}, \cdots, M_{i_n}$ being replaced is called $L(i_1, i_2, \cdots, i_n)$. These newly replaced machines are called *cause machines*. The optimal buffer distribution of line $L(i_1, i_2, \cdots, i_n)$ is denoted by $\mathbf{N}(i_1, i_2, \cdots, i_n)$. For instance, if Machines M_5 and M_{25} of the base line are replaced by other machines, then the new line is called $L(5, 25)$ and its optimal buffer distribution is $\mathbf{N}(5, 25)$. Moreover, we define $\mathbf{D}(i_1, i_2, \cdots, i_n) = \mathbf{N}(i_1, i_2, \cdots, i_n) - \mathbf{N}^\star$, the difference between the optimal buffer distribution for $L(i_1, i_2, \cdots, i_n)$ and the optimal buffer distribution for the base line $L^\star$.

9.2.2 Case 1: Two Bottleneck Machines

In this section, we consider the first case where two machines of $L^\star$ are replaced by two bottleneck machines. In particular, we choose Machines M_5 and M_{25} to be two bottleneck machines (cause machines) with parameters $r_5 = r_{25} = .08$ and $p_5 = p_{25} = .01$. Therefore, their isolated production rates are $P_5 = P_{25} = .889$. Other machines as well as buffer costs remain unchanged. The line is called $L(5, 25)$ and its optimal buffer distribution is $\mathbf{N}(5, 25)$ and $\mathbf{D}(5, 25) = \mathbf{N}(5, 25) - \mathbf{N}^\star$. On the other hand, instead of considering two bottlenecks, we consider only one bottleneck machine at a time and optimize $L(5)$ and $L(25)$ separately. Let $\mathbf{D}(5) = \mathbf{N}(5) - \mathbf{N}^\star$ and $\mathbf{D}(25) = \mathbf{N}(25) - \mathbf{N}^\star$ where $\mathbf{N}(5)$ is the optimal buffer distribution for the line

with only Machine M_5 being the bottleneck machine, and $\mathbf{N}(25)$ is the optimal buffer distribution for the line with only Machine M_{25} being the bottleneck machine. We display $\mathbf{D}(5)$ and $\mathbf{D}(25)$ in Figure 9-2.

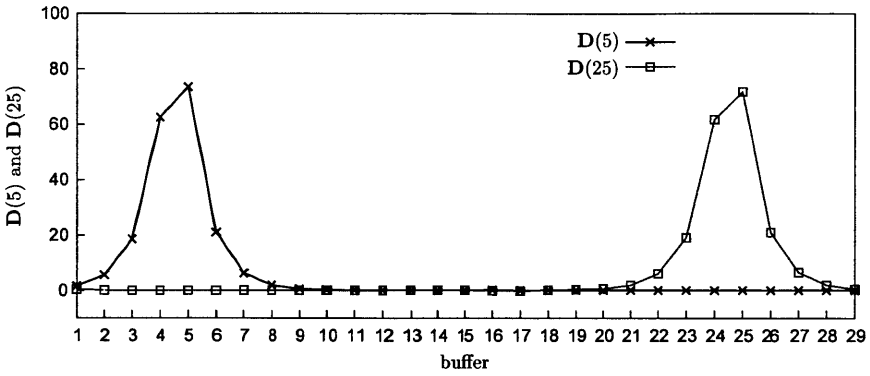


Figure 9-2: $\mathbf{D}(5)$ and $\mathbf{D}(25)$, Case 1

It can be seen from Figure 9-2 that $\mathbf{D}(5)$ looks like a pulse centered at Buffer B_5, while $\mathbf{D}(25)$ looks like a pulse centered at Buffer B_{25}. More importantly, $\mathbf{D}(5)$ shows that the bottleneck machine M_5 has no observable impact on Buffer B_{10} to Buffer B_{29}, since the components of these buffers in $\mathbf{D}(5)$ are almost 0. Similarly, $\mathbf{D}(25)$ shows that the bottleneck machine M_{25} has no observable impact on Buffer B_1 to Buffer B_{20}.

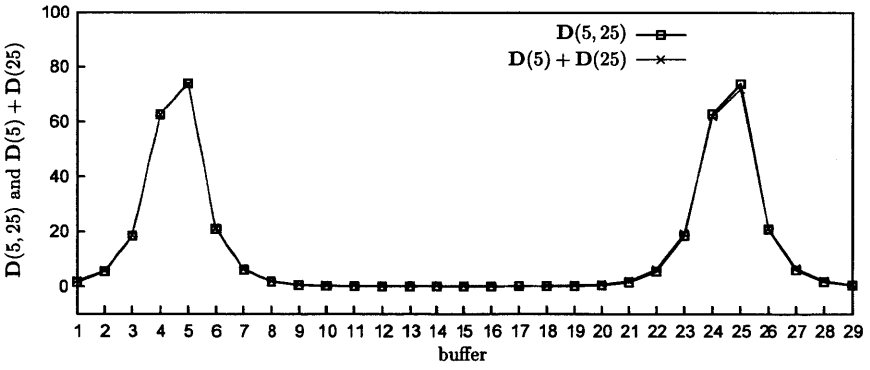


Figure 9-3: $\mathbf{D}(5,25)$ and $\mathbf{D}(5)+\mathbf{D}(25)$, Case 1

Now, we compare $\mathbf{D}(5,25)$ and $\mathbf{D}(5)+\mathbf{D}(25)$. This is shown in Figure 9-3. As a reminder, $\mathbf{D}(5,25) = \mathbf{N}(5,25) - \mathbf{N}^{\star}$. It is the effect brought to $\mathbf{N}^{\star}$ by the two bot-

tleneck machines M_5 and M_{25} together. It can be seen from Figure 9-3 that $\mathbf{D}(5,25)$ is approximately equal to $\mathbf{D}(5)+\mathbf{D}(25)$. This is because, according to the observation made from Figure 9-2, the impacts of bottleneck machines M_5 and M_{25} on the optimal buffer distribution $\mathbf{N}^\star$ of the base line $L^\star$ are approximately **independent**, and therefore their effects are approximately **additive**. In other words, optimizing line $L(5,25)$ with bottleneck machines M_5 and M_{25} is approximately equivalent to optimizing $L(5)$ and $L(25)$ separately and adding their effects. This explains why $\mathbf{D}(5,25) \approx \mathbf{D}(5)+\mathbf{D}(25)$. We refer to the fact $\mathbf{D}(5,25) \approx \mathbf{D}(5)+\mathbf{D}(25)$ as the *additive property* in long line optimization. The argument above can be also expressed with mathematical notations.

$$\begin{aligned}
\mathbf{N}(5,25) &\approx \mathbf{N}(5)+\mathbf{D}(25) \\
&= \mathbf{N}(5)+\mathbf{N}(25)-\mathbf{N}^\star \\
&= \Big(\mathbf{N}(5)-\mathbf{N}^\star\Big)+\Big(\mathbf{N}(25)-\mathbf{N}^\star\Big)+\mathbf{N}^\star \\
&= \mathbf{D}(5)+\mathbf{D}(25)+\mathbf{N}^\star.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbf{N}(5,25) &\approx \mathbf{D}(5)+\mathbf{D}(25)+\mathbf{N}^\star. \\
\mathbf{N}^\star+\mathbf{D}(5,25) &\approx \mathbf{D}(5)+\mathbf{D}(25)+\mathbf{N}^\star. \\
\mathbf{D}(5,25) &\approx \mathbf{D}(5)+\mathbf{D}(25).
\end{aligned}$$

9.2.3 Case 2: Two Anti-bottleneck Machines

In Case 2, we consider an opposite situation from Case 1. In particular, M_5 and M_{25} in the base line $L^\star$ are replaced by two anti-bottleneck machines with parameters $r_5 = .13$, $r_{25} = .12$, and $p_5 = p_{25} = .01$. Therefore, their isolated production rates are $P_5 = .929$ and $P_{25} = .923$. We study $\mathbf{D}(5,25)$ and $\mathbf{D}(5)+\mathbf{D}(25)$ as before. Figure

9-4 shows $\mathbf{D}(5)$ and $\mathbf{D}(25)$. Note that these are inverse pulses centered at Buffers B_5 and B_{25}, respectively. This is because in this case these two cause machines are anti-bottleneck machines, and therefore their corresponding adjacent buffers are reduced due to smaller variabilities of the anti-bottleneck machines.

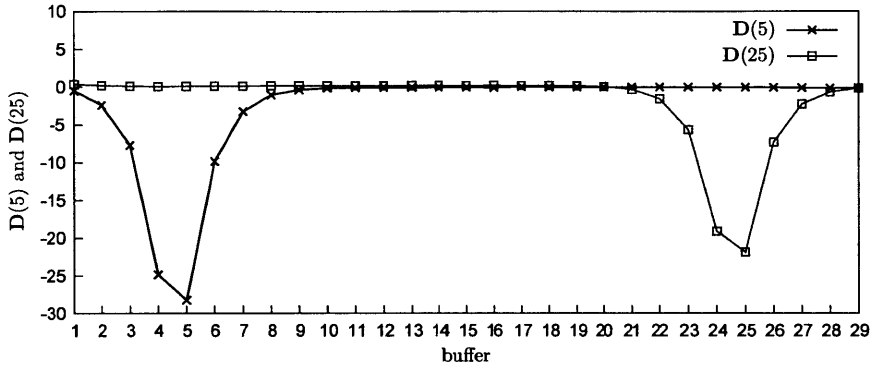


Figure 9-4: $\mathbf{D}(5)$ and $\mathbf{D}(25)$, Case 2

It can be seen from Figure 9-5 that $\mathbf{D}(5,25) \approx \mathbf{D}(5) + \mathbf{D}(25)$. As before, this result indicates that the effect of the anti-bottleneck machine M_5 is approximately independent of the effect of the anti-bottleneck machine M_{25} on $\mathbf{N}^*$. Therefore, the two effects are approximately additive.

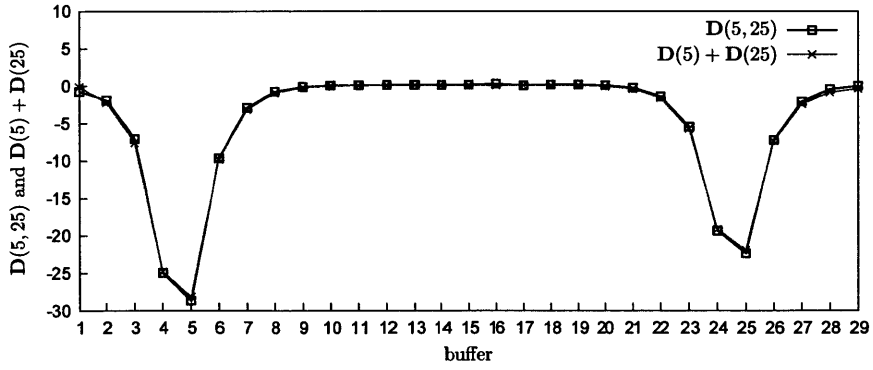


Figure 9-5: $\mathbf{D}(5,25)$ and $\mathbf{D}(5) + \mathbf{D}(25)$, Case 2

9.2.4 Case 3: One Bottleneck Machine and One Anti-bottleneck Machine

In Case 3 we modify the base line $L^\star$ with a bottleneck machine M_5 and an anti-bottleneck machine M_{25}. Their parameters are $r_5 = .08$, $r_{25} = .12$, and $p_5 = p_{25} = .01$. Thus, $P_5 = .889$ and $P_{25} = .923$. Other machines as well as buffer costs remain unchanged. We study $\mathbf{D}(5, 25)$ and $\mathbf{D}(5) + \mathbf{D}(25)$ as before. Figure 9-6 shows $\mathbf{D}(5)$ and $\mathbf{D}(25)$. $\mathbf{D}(5)$ looks like a pulse centered at Buffer B_5 while $\mathbf{D}(25)$ looks like an inverse pulse centered at Buffer B_{25}.

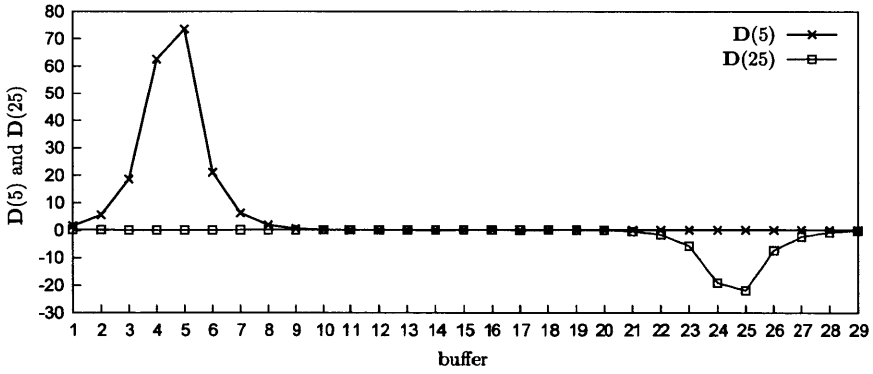


Figure 9-6: $\mathbf{D}(5)$ and $\mathbf{D}(25)$, Case 3

Figure 9-7 shows that $\mathbf{D}(5, 25) \approx \mathbf{D}(5) + \mathbf{D}(25)$. This result indicates that the effect of the bottleneck machine M_5 is approximately independent of the effect of the anti-bottleneck machine M_{25}. Therefore, the two effects are approximately additive.

9.2.5 More General Cases

The three cases mentioned in Sections 9.2.2, 9.2.3, and 9.2.4 are three basic variations of the base line $L^\star$ and therefore three basic situations from which we observe the additive property in long line optimization. In this section, we study more general cases that are derived from these three basic situations. Specifically, we study the following two questions:

1. Whether the additive property holds if we reduce the distance between the two

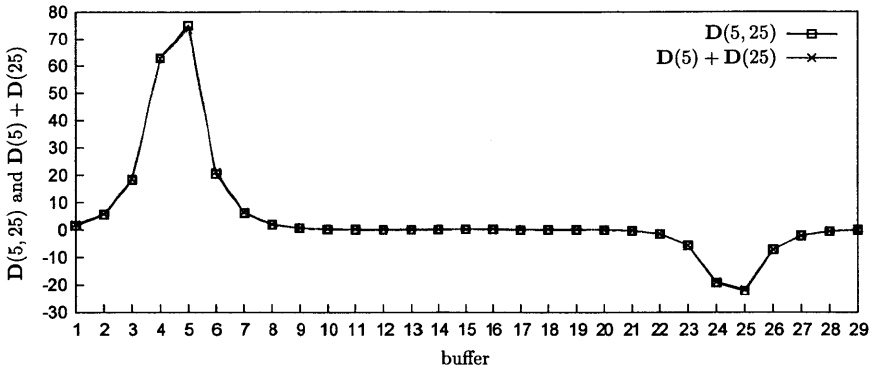


Figure 9-7: $\mathbf{D}(5,25)$ and $\mathbf{D}(5) + \mathbf{D}(25)$, Case 3

cause machines;

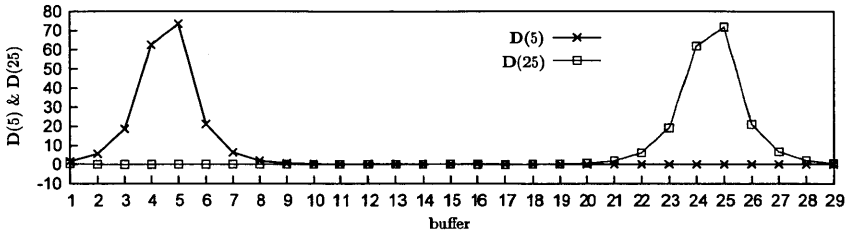
2. Whether the additive property holds if there are more than two cause machines.

1. The effect of the distance between the two cause machines

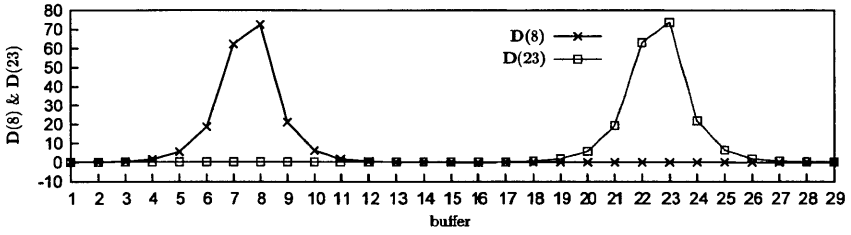
In the three basic situations, we observe that the effect of M_5 and the effect of M_{25} on the optimal buffer distribution of the base line are approximately independent and additive. This is because the cluster of buffers affected by M_5 and the cluster of buffers affected by M_{25} do not overlap. The distance between the two cause machines (M_5 and M_{25}) are 19 (in terms of the number of machines between them). However, as we reduce the distance between the two cause machine, the two clusters of affected buffers move towards each other. If the two cause machines are close enough, part of the two clusters may overlap. We are interested in whether the additive property still holds as we reduce the distance between the two cause machines. We describe some experiments below.

Again, we start with two bottleneck machines. In particular, we first let Machines M_5 and M_{25} be the two cause machines and then move the two cause machines towards each other. As a result, the four lines considered are $L(5,25)$, $L(8,23)$, $L(11,20)$, and $L(14,17)$. As the two cause machines move closer, the two clusters of affected buffers move towards each other as well. This can be seen from Figure 9-8, which shows the individual effect of each cause machine. When the two cause machines are M_{14} and

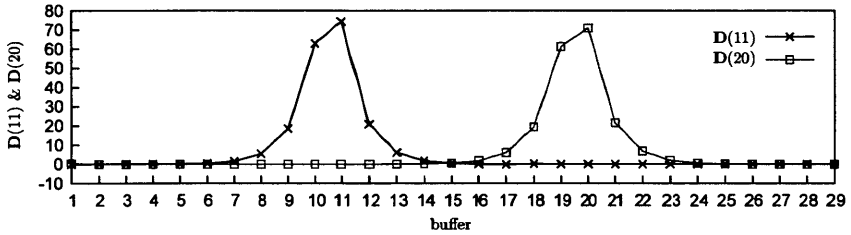
M_{17}, the two clusters overlap partially. The additive property in these four lines are demonstrated in Figure 9-9. Even in $L(14, 17)$ where the two clusters overlap, the additive property still holds as $\mathbf{D}(14, 17) \approx \mathbf{D}(14) + \mathbf{D}(17)$.



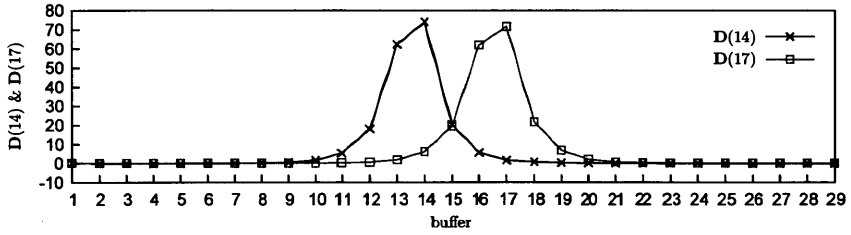
(a) Machines M_5 and M_{25} are bottleneck machines



(b) Machines M_8 and M_{23} are bottleneck machines

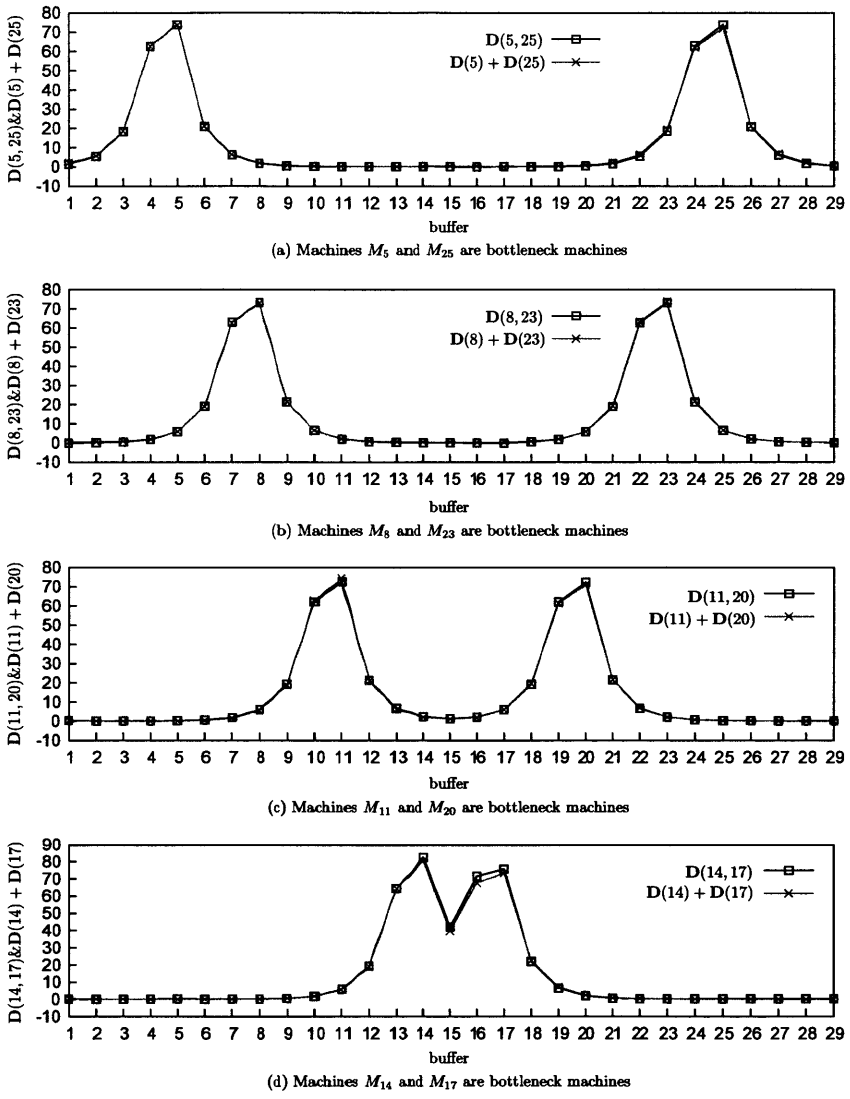


(c) Machines M_{11} and M_{20} are bottleneck machines



(d) Machines M_{14} and M_{17} are bottleneck machines

Figure 9-8: Individual effect of each cause machine, two bottleneck machines, Example 1



(a) Machines M_5 and M_{25} are bottleneck machines

(b) Machines M_8 and M_{23} are bottleneck machines

(c) Machines M_{11} and M_{20} are bottleneck machines

(d) Machines M_{14} and M_{17} are bottleneck machines

Figure 9-9: Effect of the distance between the two cause machines, two bottleneck machines, Example 1

Next, we consider another example that has two different bottleneck machines. The repair probabilities of the two bottleneck machines are .08 and .09, respectively. The four lines being considered are still $L(5,25), L(8,23), L(11,20)$, and $L(14,17)$. The individual effect of each bottleneck machine in these four lines and the additive property in these four lines are illustrated in Figure 9-10 and Figure 9-11, respectively. The additive property exits in all four lines.

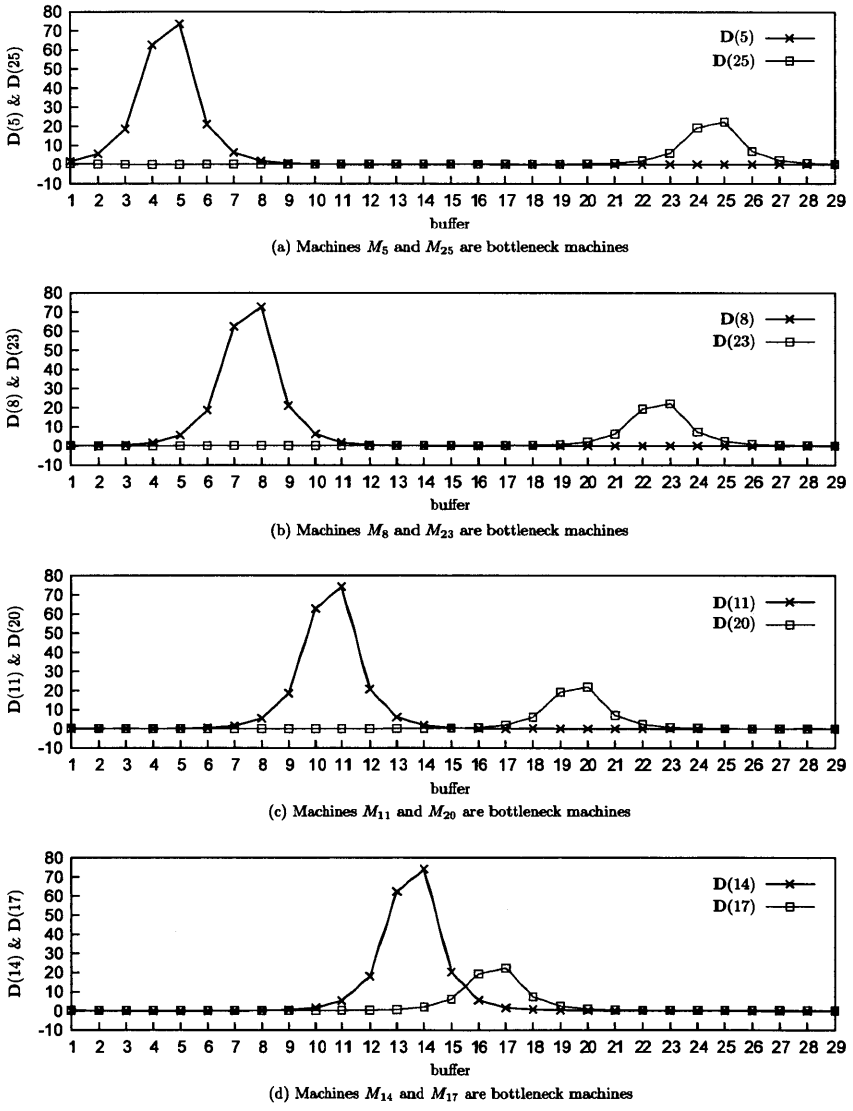
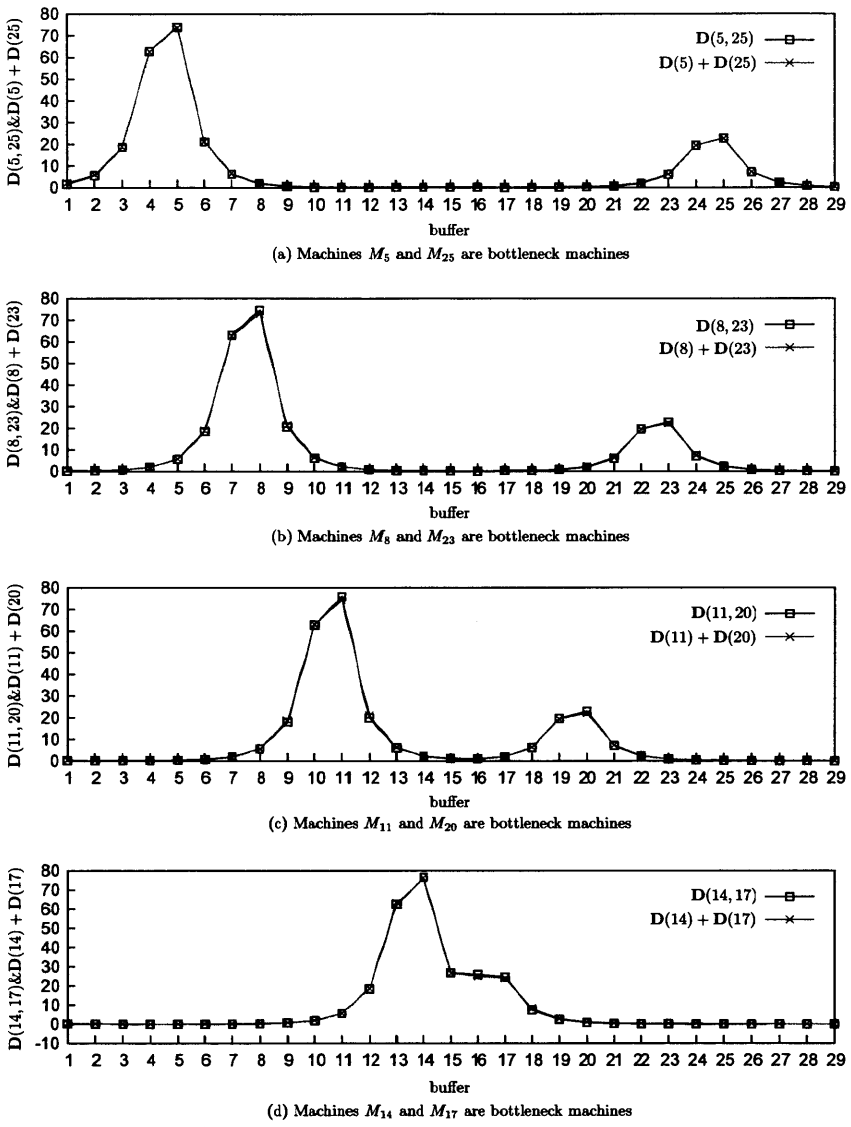


Figure 9-10: Individual effect of each cause machine, two bottleneck machines, Example 2



(a) Machines M_5 and M_{25} are bottleneck machines

(b) Machines M_8 and M_{23} are bottleneck machines

(c) Machines M_{11} and M_{20} are bottleneck machines

(d) Machines M_{14} and M_{17} are bottleneck machines

Figure 9-11: Effect of the distance between the two cause machines, two bottleneck machines, Example 2

Next, we study an example of Case 2 where two machines in the base line $L^\star$ are replaced by two anti-bottleneck machines. The repair probabilities of the two anti-bottleneck machines are .13 and .12, respectively. The four lines being considered are again $L(5,25), L(8,23), L(11,20)$, and $L(14,17)$. The individual effect of each bottleneck machine in these four lines and the additive property in these four lines are illustrated in Figure 9-12 and Figure 9-13, respectively. The additive property exits in all four lines.

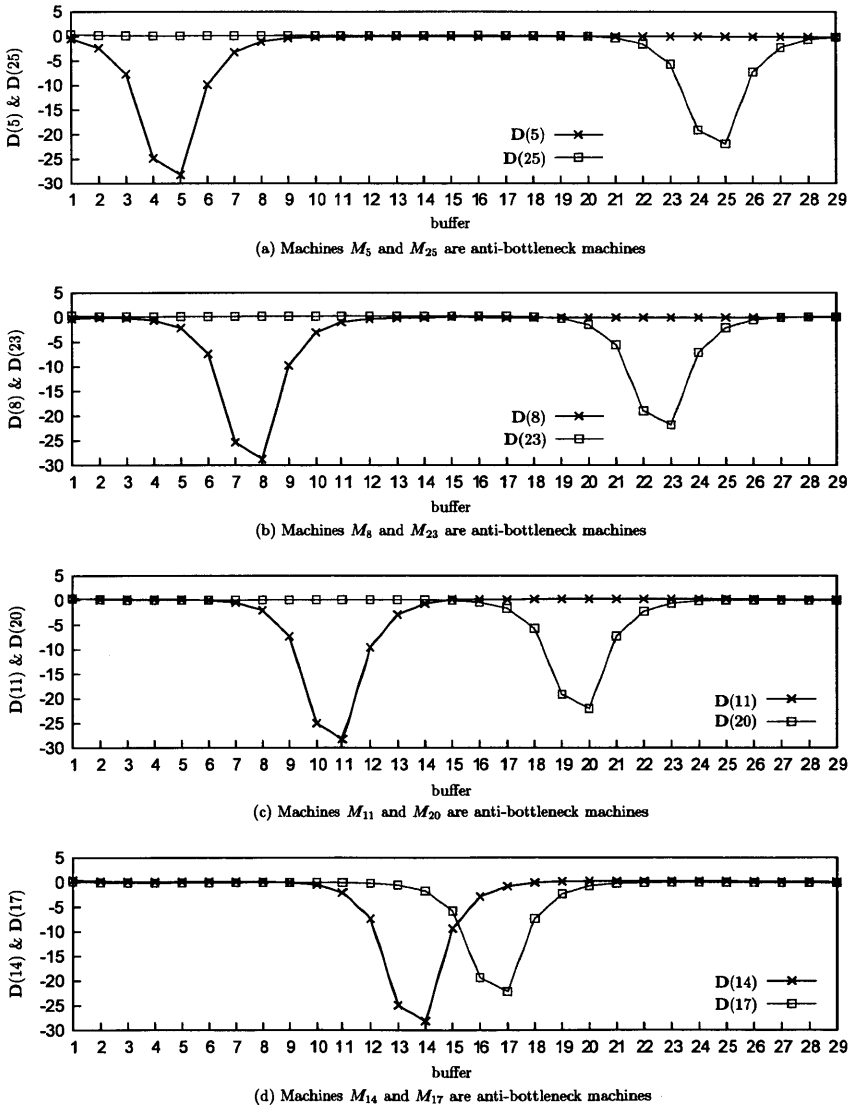
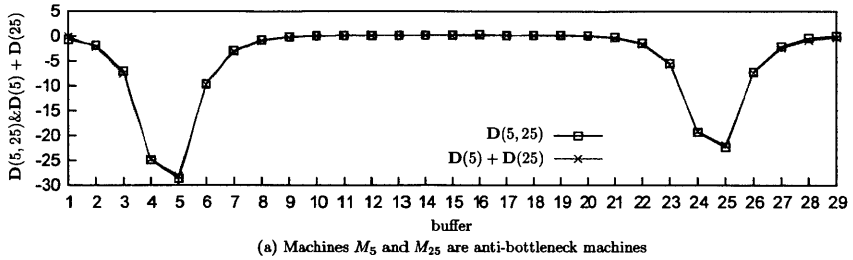
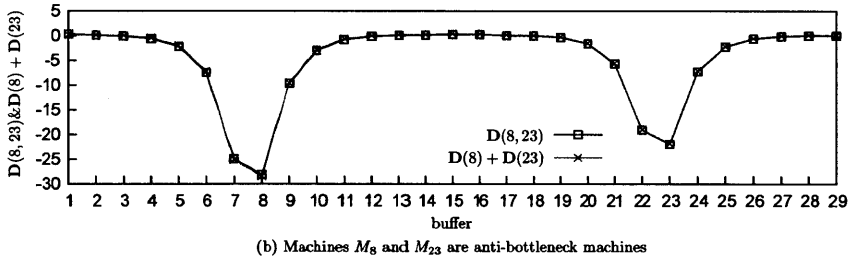


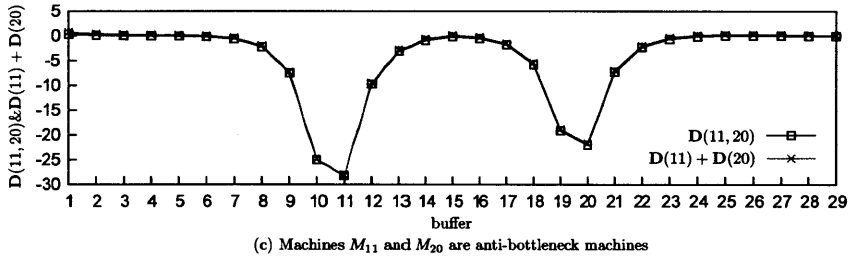
Figure 9-12: Individual effect of each cause machine, two anti-bottleneck machines



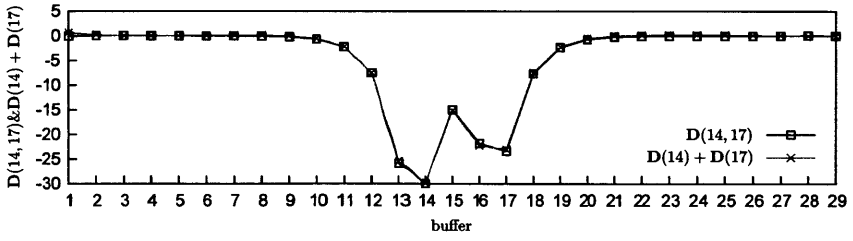
(a) Machines M_5 and M_{25} are anti-bottleneck machines



(b) Machines M_8 and M_{23} are anti-bottleneck machines



(c) Machines M_{11} and M_{20} are anti-bottleneck machines



(d) Machines M_{14} and M_{17} are anti-bottleneck machines

Figure 9-13: Effect of the distance between the two cause machines, two anti-bottleneck machines

Finally, we consider an example of Case 3 where two machines in the base line $L^\star$ are replaced by one bottleneck machine and one anti-bottleneck machine. The repair probabilities of the two machines are .08 and .12, respectively. The four lines being considered are $L(5,25)$, $L(8,23)$, $L(11,20)$ and $L(14,17)$. The individual effect of each bottleneck machine in these four lines and the additive property in these four lines are illustrated in Figures 9-14 and 9-15, respectively. The additive property exits in all four lines.

The four examples above indicate that the additive property is almost insensitive to the distance between the two cause machines. However, we have to point out clearly that, when two cause machines are close enough, the additive property is less accurate. We show this in Section 9.4.

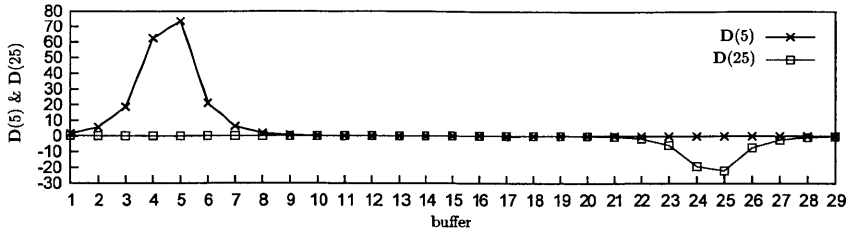
2. Effect of the number of cause machines

In this section, we discuss whether the additive property holds if there are more than two cause machines. Before running any experiments, we expect that the additive property holds, since the additive property is almost insensitive to the distance between two adjacent cause machines. Therefore, as long as any two adjacent cause machines are not too close together, we expect the additive property to hold regardless of the number of cause machines. The set of examples below verify our expectation.

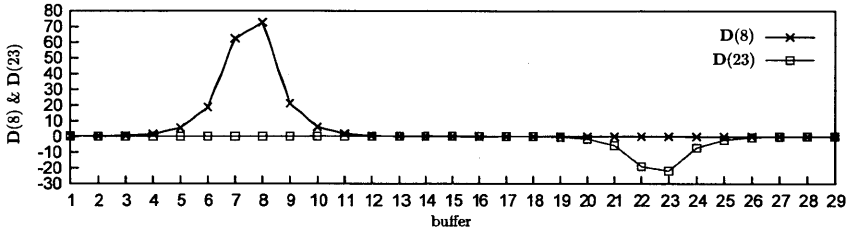
We first modify the base line $L^\star$ with three bottleneck machines M_{i_1}, M_{i_2} and M_{i_3}. Four sets of M_{i_1}, M_{i_2} and M_{i_3} are considered. The additive property holds in all these four lines and it is shown in Figure 9-16.

In the second example, we consider five cause machines spaced evenly in the 30-machine 29-buffer line. In other words, Machines M_5, M_{10}, M_{15}, M_{20}, and M_{25} in $L^\star$ are first replaced by five bottleneck machines whose repair probabilities are .09, and then replaced by five anti-bottleneck machines whose repair probabilities are .12. These results are shown in Figure 9-17, which depicts that $\mathbf{D}(5,10,15,20,15) \approx \mathbf{D}(5)+\mathbf{D}(10)+\mathbf{D}(15)+\mathbf{D}(20)+\mathbf{D}(25)$ in both scenarios. Consequently, the additive property still holds.

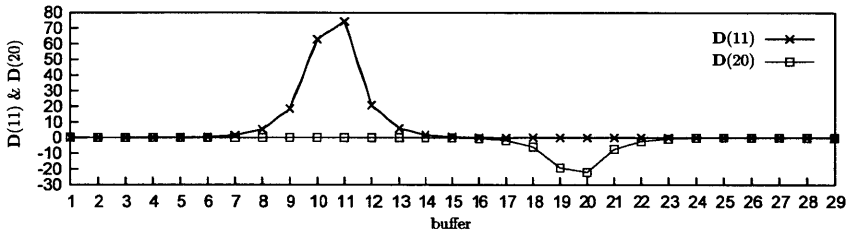
In the third example, we consider seven cause machines spaced evenly. Machines



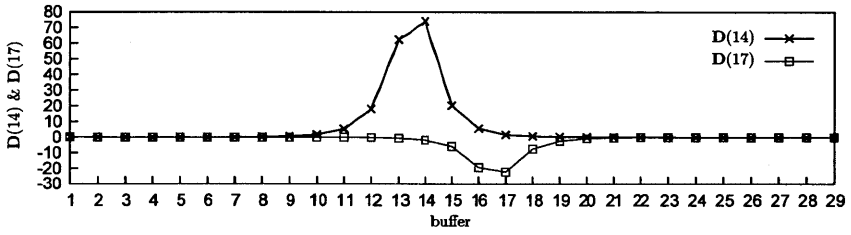
(a) Machine M_5 is the bottleneck; Machine M_{25} is the anti-bottleneck



(b) Machine M_8 is the bottleneck; Machine M_{23} is the anti-bottleneck



(c) Machine M_{11} is the bottleneck; Machine M_{20} is the anti-bottleneck



(d) Machine M_{14} is the bottleneck; Machine M_{17} is the anti-bottleneck

Figure 9-14: Individual effect of each cause machine, one bottleneck machine and one anti-bottleneck machine

M_3, M_7, M_{11}, M_{15}, M_{19}, M_{23}, and M_{27} in $L^\star$ are first replaced by seven bottleneck machines whose repair probabilities are .09, and then replaced by seven anti-bottleneck machines whose repair probabilities are .12. These results are shown in Figure 9-18, which depicts that $\mathbf{D}(3,7,11,15,19,23,27) \approx \mathbf{D}(3)+\mathbf{D}(7)+\mathbf{D}(11)+\mathbf{D}(15)+\mathbf{D}(19)+\mathbf{D}(23)+\mathbf{D}(27)$ in both scenarios. Consequently, the additive property holds.

In the fourth example, we consider nine cause machines spaced evenly. Machines

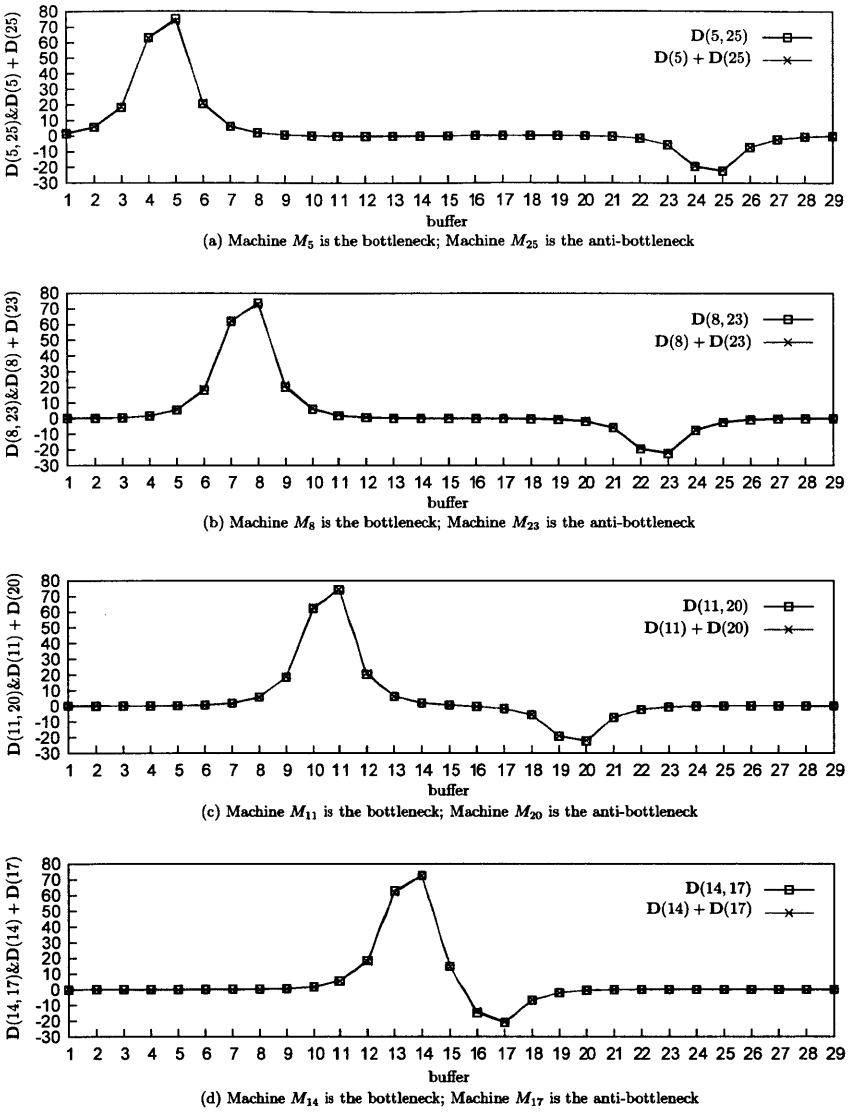


Figure 9-15: Effect of the distance between the two cause machines, one bottleneck machine and one anti-bottleneck machine

M_3, M_6, M_9, M_{12}, M_{15}, M_{18}, M_{21}, M_{24}, and M_{27} in $L^\star$ are first replaced by nine bottleneck machines whose repair probabilities are $r_3 = .09$, $r_6 = .09$, $r_9 = .085$, $r_{12} = .08$, $r_{15} = .08$, $r_{18} = .08$, $r_{21} = .085$, $r_{24} = .09$, and $r_{27} = .09$. Then these nine machines are replaced by another nine anti-bottleneck machines whose repair probabilities are $r_3 = .11$, $r_6 = .11$, $r_9 = .12$, $r_{12} = .12$, $r_{15} = .13$, $r_{18} = .12$, $r_{21} = .12$, $r_{24} = .11$, and $r_{27} = .11$. The results are shown in Figure 9-19, which demonstrates

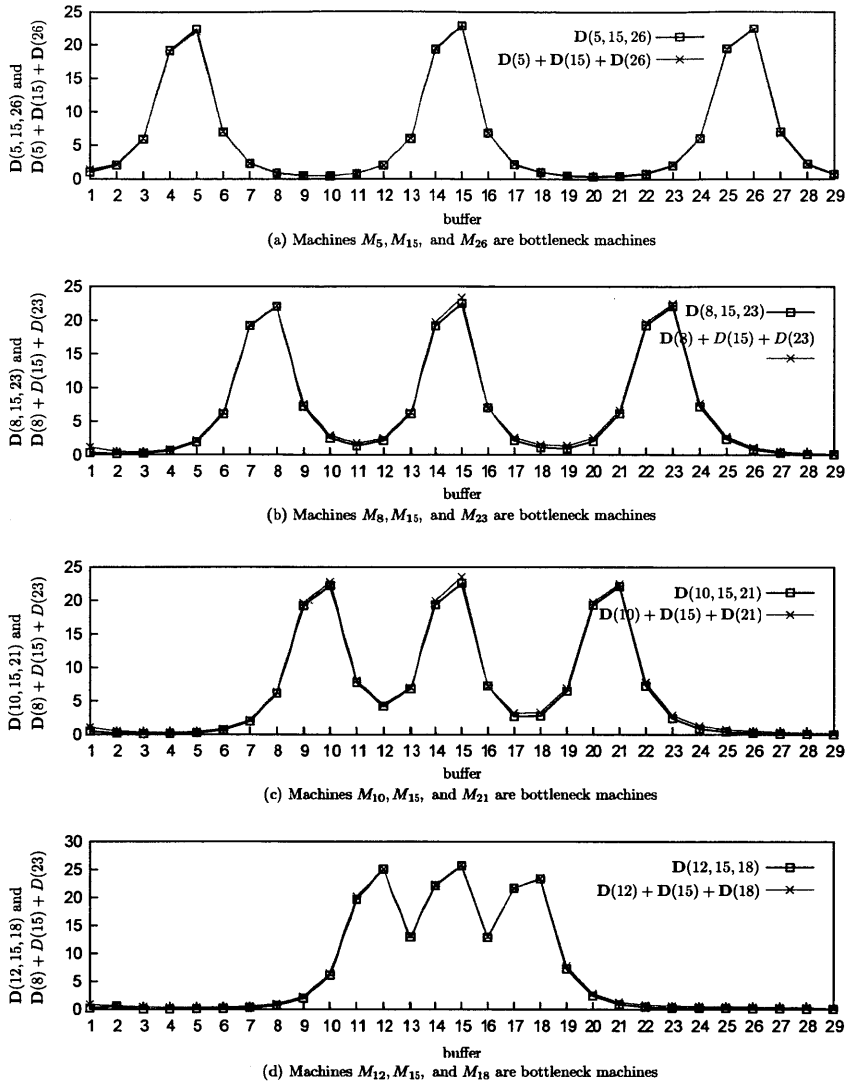
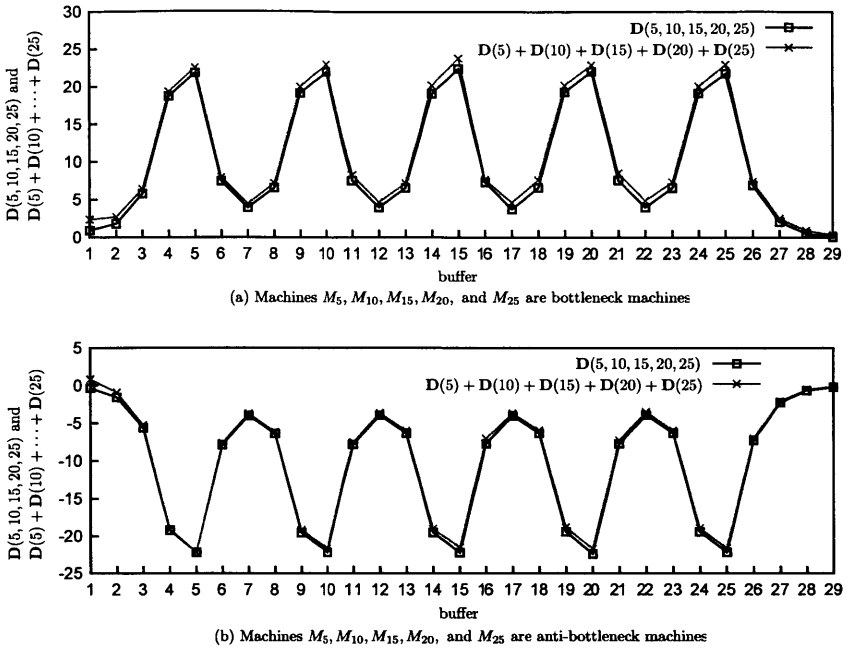


Figure 9-16: Effect of the number of cause machines, Example 1

that $\mathbf{D}(3,6,9,12,15,18,21,24,27) \approx \mathbf{D}(3)+\mathbf{D}(6)+\mathbf{D}(9)+\mathbf{D}(12)+\mathbf{D}(15)+\mathbf{D}(18)+\mathbf{D}(21)+\mathbf{D}(24)+\mathbf{D}(27)$ in both scenarios. Thus, the additive property holds.

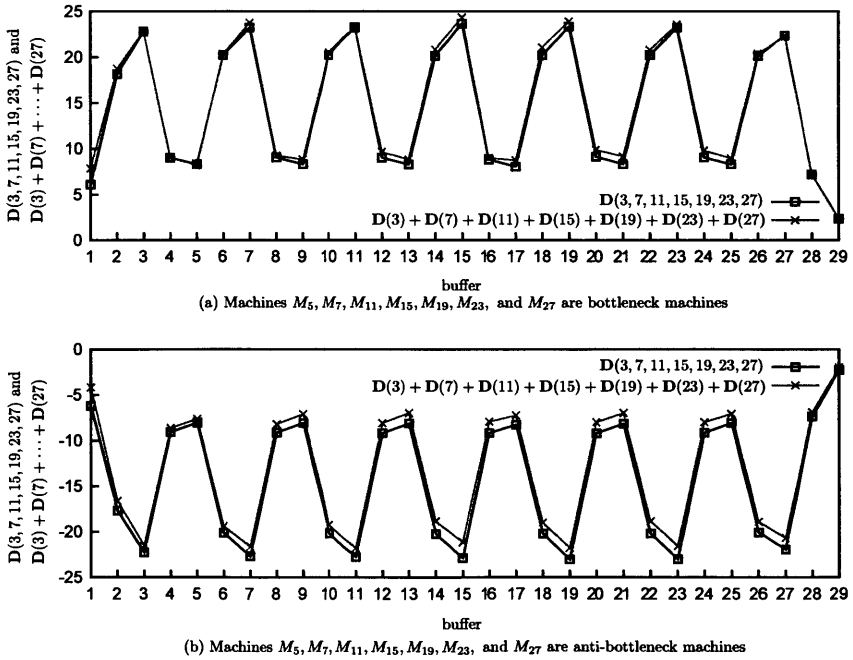
Finally, in the fifth example we consider nine cause machines again. However, some of them are bottleneck machines while the others are anti-bottleneck machines. The repair probabilities of these cause machines are $r_3 = .12$, $r_6 = .09$, $r_9 = .12$, $r_{12} = .09$, $r_{15} = .12$, $r_{18} = .09$, $r_{21} = .12$, $r_{24} = .09$, and $r_{27} = .12$. The result is shown in Figure 9-20, which demonstrates that $\mathbf{D}(3,6,9,12,15,18,21,24,27) \approx$



(a) Machines $M_5, M_{10}, M_{15}, M_{20}$, and M_{25} are bottleneck machines

(b) Machines $M_5, M_{10}, M_{15}, M_{20}$, and M_{25} are anti-bottleneck machines

Figure 9-17: Effect of the number of cause machines, Example 2



(a) Machines $M_5, M_7, M_{11}, M_{15}, M_{19}, M_{23}$, and M_{27} are bottleneck machines

(b) Machines $M_5, M_7, M_{11}, M_{15}, M_{19}, M_{23}$, and M_{27} are anti-bottleneck machines

Figure 9-18: Effect of the number of cause machines, Example 3

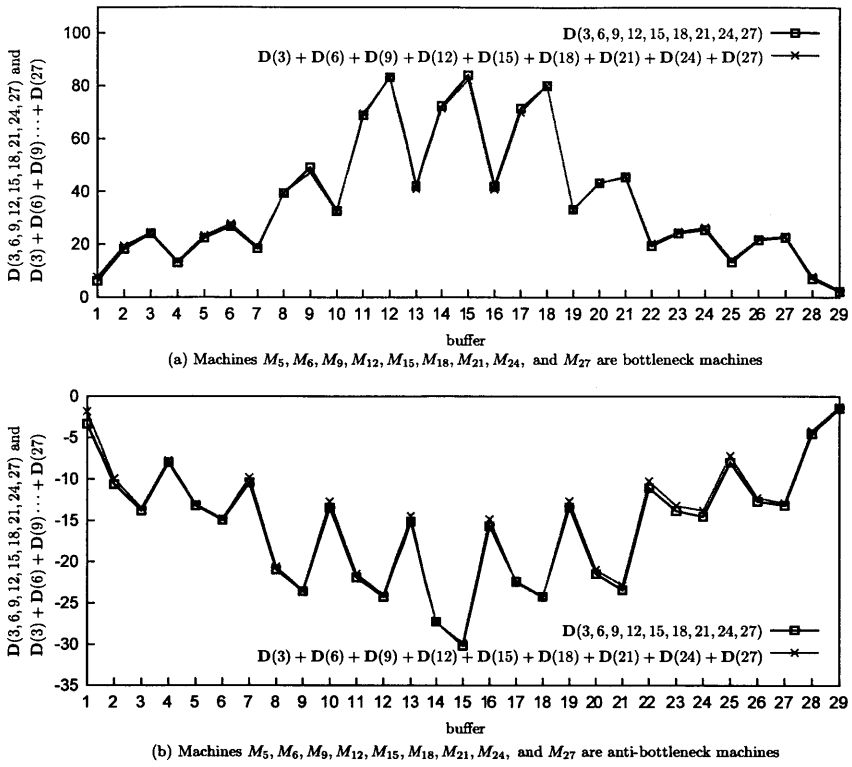


Figure 9-19: Effect of the number of cause machines, Example 4

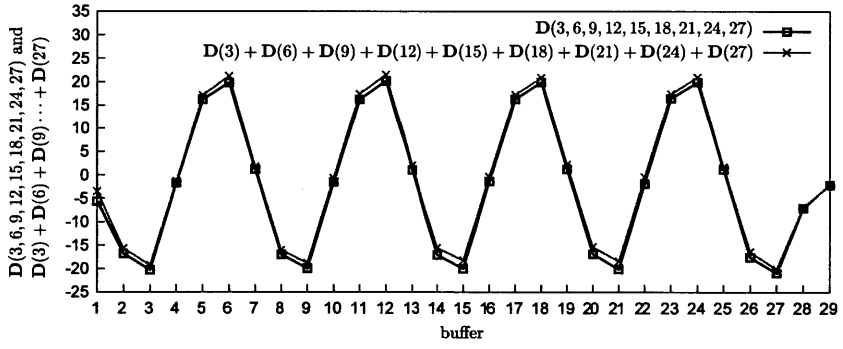


Figure 9-20: Effect of the number of cause machines, Example 5

$\mathbf{D(3) + D(6) + D(9) + D(12) + D(15) + D(18) + D(21) + D(24) + D(27)}$. Thus, the additive property holds.

These five examples above demonstrate that the additive property holds when there are more than two cause machines. As long as the effects on the optimal buffer

distribution of the base line of any two cause machines are approximately independent, the additive property holds regardless of the number of cause machines in the line.

9.3 Explanation

In this section, we explain intuitively why the additive property holds. To show this, we first modify the base line $L^\star$ by replacing a machine by a bottleneck machine. Consider line $L(11)$ where Machine M_{11} in the base line is replaced by a bottleneck machine whose repair probability is .08. Then we compare $\mathbf{N}^\star$ and $\mathbf{N}(11)$ in Figure 9-21.

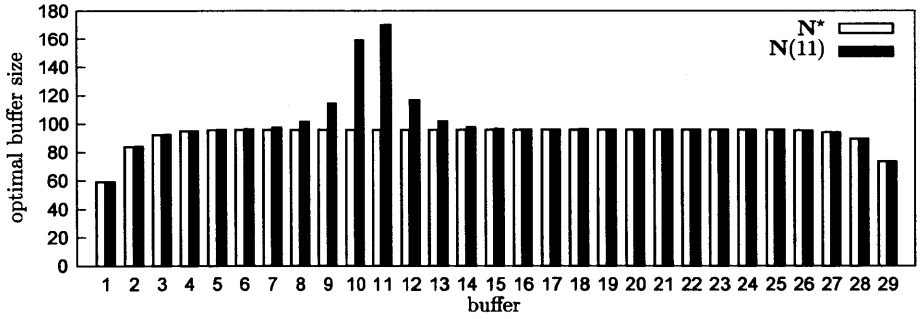


Figure 9-21: Explanation of the additive property, bottleneck machines

Figure 9-21 indicates that the optimal sizes of Buffers B_{10} and B_{11} in $L(11)$ are considerably larger than those in $L^\star$. This is because in $L(11)$, M_{11} is the bottleneck machine whose isolated production rate ($P_{11} = .889$) is smaller than any other machines in the line ($P_i = .909, \forall i \neq 11$). The bottleneck machine exhibits larger variability than any other machines. This large variability requires B_{10} and B_{11} to be large, and therefore they can absorb the variability and guarantee the performance of the line in terms of achieving the target production rate. Note that Buffers B_9 and B_{12} of $L(11)$ are also enlarged due to the large variability of M_{11}. However, they are much smaller than B_{10} and B_{11}, because B_{10} and B_{11} have absorbed most of the variability. In addition, the effect of M_{11} on B_8 and B_{13} is observable but very small. Finally, M_{11} has no observable impact on other buffers in $L(11)$. This is because the small set of buffers affected by M_{11} have absorbed all the variability, and they prevent

M_{11} from affecting buffers further upstream or downstream. Therefore, only a small cluster of buffers adjacent to the cause bottleneck machine M_{11} are affected. If we compute $\mathbf{D}(11)$ by subtracting $\mathbf{N}^\star$ from $\mathbf{N}(11)$, then it looks like a pulse centered at B_{11}.

Next, we modify the base line $L^\star$ by replacing a machine by an anti-bottleneck machine. Consider line $L(11)$ again where M_{11} in the base line is replaced by an anti-bottleneck machine whose repair probability is .13. Then we compare $\mathbf{N}^\star$ and $\mathbf{N}(11)$ in Figure 9-22.

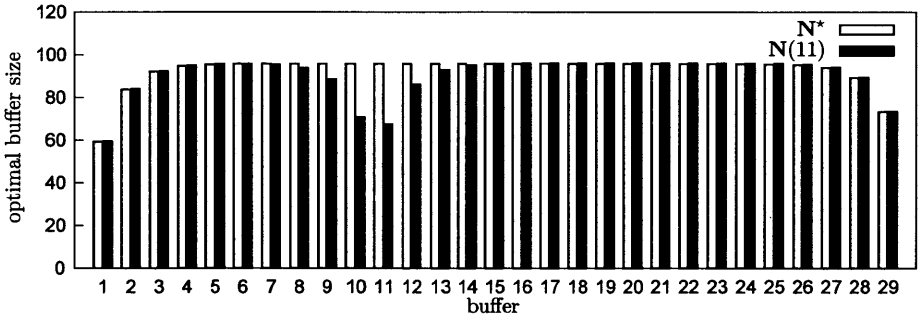


Figure 9-22: Explanation of the additive property, anti-bottleneck machines

Figure 9-22 shows that the optimal sizes of B_{10} and B_{11} in $L(11)$ are considerably smaller than those in $L^\star$. This is because in $L(11)$, M_{11} is the anti-bottleneck machine whose isolated production rate ($P_{11} = .929$) is larger than any other machines ($P_i = .909, \forall i \neq 11$). The anti-bottleneck machine exhibits lower variability (and higher reliability) than any other machines. The high reliability of M_{11} allows B_{10} and B_{11} to be small, since there is less variability for them to absorb in order to achieve the target production rate. Note that B_9 and B_{12} are also pulled down due to the large reliability of M_{11}. However, B_9 and B_{12} are larger than B_{10} and B_{11}, because B_{10} and B_{11} have absorbed most of the reliability. In addition, the effect of M_{11} on B_8 and B_{13} is observable but very small. Finally, M_{11} has no observable impact on other buffers. The few buffers affected by M_{11} have absorbed all the reliability of M_{11}, and prevent it from affecting buffers further upstream or downstream. Therefore, as before, only a small cluster of buffers adjacent to the anti-bottleneck machine are affected. If we compute $\mathbf{D}(11)$ by subtracting $\mathbf{N}^\star$ from $\mathbf{N}(11)$, then it looks like a

inverse pulse centered at B_{11}.

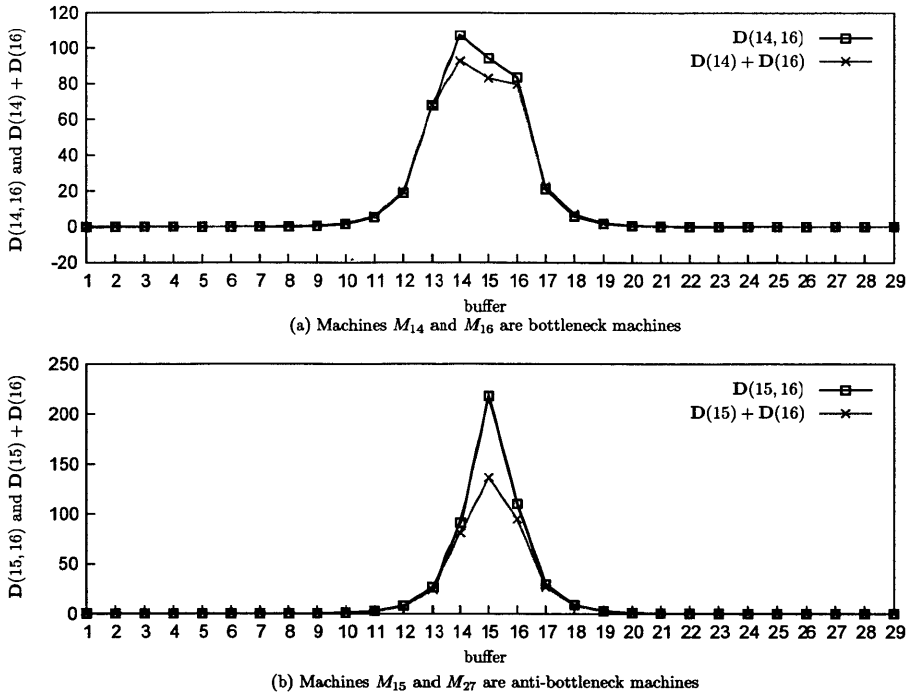
According to this analysis, we can see that a single bottleneck machine or a single anti-bottleneck machine has an observable affect on only a small cluster of buffers adjacent to it. The optimal solution of the line with a bottleneck machine can be found by adding a pulse centered at the buffer downstream of the cause machine to the optimal solution of the base line. On the other hand, the optimal solution of the line with an anti-bottleneck machine can be found by adding an inverse pulse centered at the buffer downstream of the cause machine to the optimal solution of the base line.

If the line has more than one cause machine (and each of them can be either a bottleneck or anti-bottleneck machine), as long as the distance between any two of them is not too small, their effects on the optimal buffer distribution are approximately independent and additive. This explains why the additive property holds in long line optimization. In summary, the additive property in long line optimization says that the effect of multiple cause machines on the optimal buffer distribution is approximately equivalent to the sum of the individual effects of each cause machine, as long as these machines are not too close together.

9.4 Extreme Cases

In the previous sections, we have shown that the additive property is almost insensitive to the distance between any two cause machines and therefore it holds in general. However, we do observe some cases where the additive property is less accurate. This is because when any two cause machines are too close together, there may exist an interaction between their effects on the optimal buffer distribution. However, the additive property fails to account for such an interaction. We explain this in this section.

Consider the first scenario where Machines M_{14} and M_{16} in the base line $L^\star$ are replaced by two bottleneck machines whose repair probabilities are .08. In the second scenario, we replace M_{15} and M_{16} in $L^\star$ by two bottleneck machines. The additive



(a) Machines M_{14} and M_{16} are bottleneck machines

(b) Machines M_{15} and M_{27} are anti-bottleneck machines

Figure 9-23: Extreme case of the additive property, Example 1

property is less accurate in both scenarios (see Figure 9-23). We use the second scenario as an example to discuss this.

Figure 9-23(b) implies that the true optimal size of B_{15} for $L(15,16)$ ($\mathbf{N}^\star + \mathbf{D}(15,16)$) is larger than the value ($\mathbf{N}^\star + \mathbf{D}(15) + \mathbf{D}(16)$) derived from the additive property. In this case, M_{15} and M_{16} are bottleneck machines, and B_{15} is the buffer between them. Both M_{15} and M_{16} require B_{15} to be large so that it absorbs the large variabilities of both machines. The result implies that there is a positive interaction between the effects of the two machines on the optimal size of B_{15}. Such an interaction can be explained as follows. Recall that both M_{15} and M_{16} are bottleneck machines. If M_{15} fails for a long time, B_{15} will be empty and M_{16} will be starved. On the other hand, if M_{16} fails for a long time, B_{15} will be full and M_{15} will be blocked. In other words, since both machines are bottleneck machines, they have greater potential to prevent each other from producing parts by either starvation or blockage if the buffer between them is not big enough. Thus, to achieve the required produc-

tion rate, the size of B_{15} needs to be large enough to absorb the variabilities of both M_{15} and M_{16} as well as to further decouple the two machines from impacting each other. However, the additive property does not account for the interaction between the effects of the two machines. Therefore, we observe a large discrepancy between $\mathbf{D}(15,16)$ and $\mathbf{D}(15)+\mathbf{D}(16)$. The incorrect optimal buffer distribution derived from the additive property overestimates the production rate of the line.

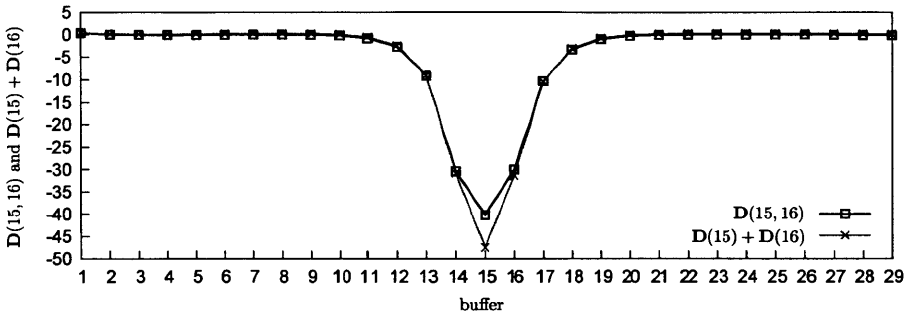


Figure 9-24: Extreme case of the additive property, Example 2

Similar results can be observed when there are two anti-bottleneck machines placed next to each other. Consider the case where Machines M_{15} and M_{16} in the base line $L^{\star}$ are replaced by two anti-bottleneck machines whose repair probabilities are .13 and .12, respectively. In this case, the additive property is less accurate as well, which is illustrated in Figure 9-24. As before, the incorrect optimal buffer distribution derived from the additive property overestimates the throughput of the line.

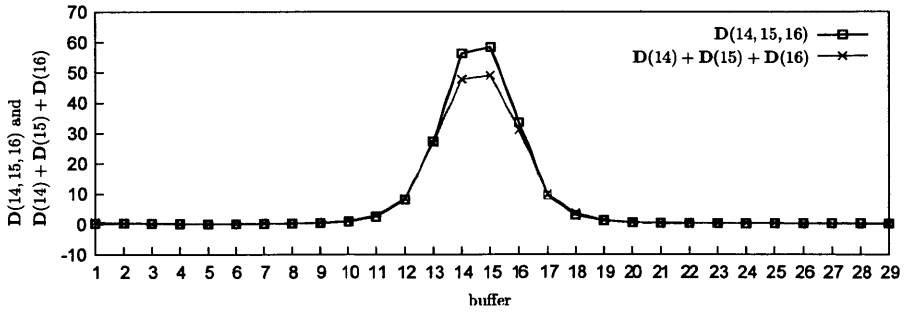


Figure 9-25: Extreme case of the additive property, Example 3

Next, we consider a case where Machines M_{14}, M_{15}, and M_{16} in the base line $L^\star$ are replaced by three bottleneck machines. Figure 9-25 shows that the additive property is less accurate in Buffers B_{14} and B_{15}.

Finally, we consider a case where machines are randomly generated. In particular, the failure probabilities for all 30 machines are still .01, while the repair probabilities of the 30 machines are illustrated in Figure 9-26. We still use the same $L^\star$ as the base line and change the repair probability of one machine at a time. In other words, we have 30 lines derived from $L^\star$ where r_i is changed in $L(i)$ while the parameters of the other machines remain unchanged. Eventually, we change the r_i's for all 30 machines and derive line $L(1, 2, \cdots, 30)$.

We compare $\mathbf{D}(1, 2, \cdots, 30)$ and $\mathbf{D}(1)+\mathbf{D}(2)+\cdots+\mathbf{D}(30)$, where $\mathbf{D}(1, \cdots, 30)$ and $\mathbf{D}(i)$ are computed by $\mathbf{N}(1, \cdots, 30) - \mathbf{N}^\star$ and $\mathbf{N}(i) - \mathbf{N}^\star, \forall i = 1, \cdots, 30$, respectively. The comparison between $\mathbf{D}(1, 2, \cdots, 30)$ and $\mathbf{D}(1) + \mathbf{D}(2) + \cdots + \mathbf{D}(30)$ is shown in Figure 9-27. In addition, The comparison between $\mathbf{N}(1, 2, \cdots, 30)$ and $\mathbf{N}^\star + \mathbf{D}(1) + \mathbf{D}(2) + \cdots + \mathbf{D}(30)$ is shown in Figure 9-28.

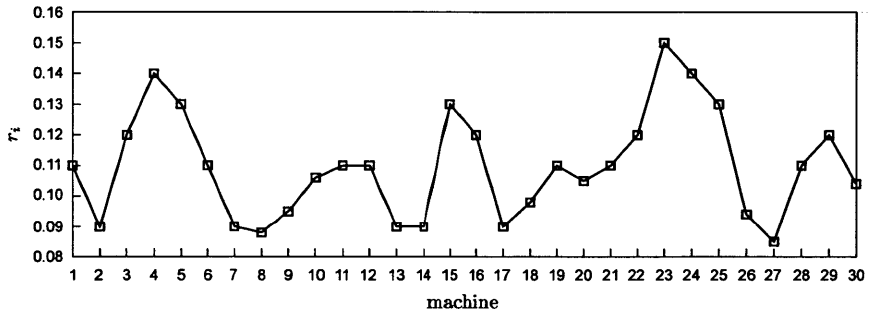


Figure 9-26: Repair probabilities of the 30 machines

Figures 9-27 and 9-28 show that when there are a pair of bottleneck machines or anti-bottleneck machines next to each other, the buffer distribution from the additive property is less accurate. For instance, note that Machines M_{21}, M_{22}, M_{23}, M_{24}, and M_{25} are all anti-bottleneck machines. The optimal buffer distribution in Buffers B_{22}, B_{23}, and B_{24} derived from the additive property diverges away from the correct distribution (see Figure 9-28). However, the middle ranges of the two curve are very close to each other.

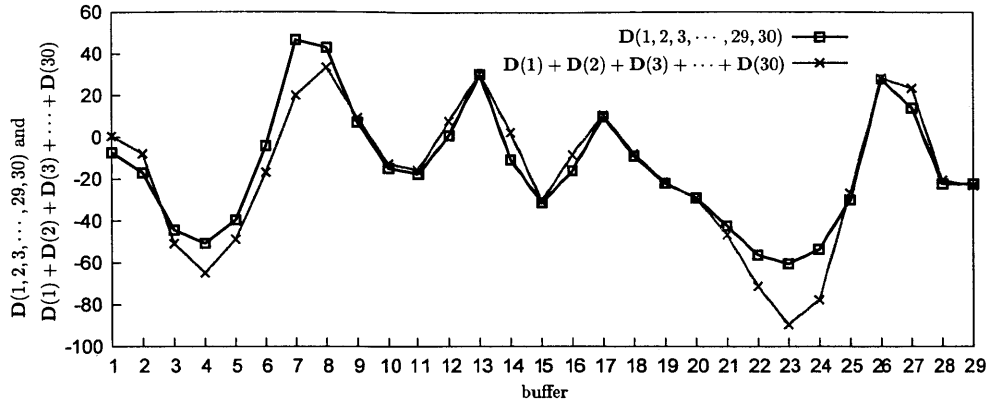


Figure 9-27: Extreme case of the additive property, Example 4

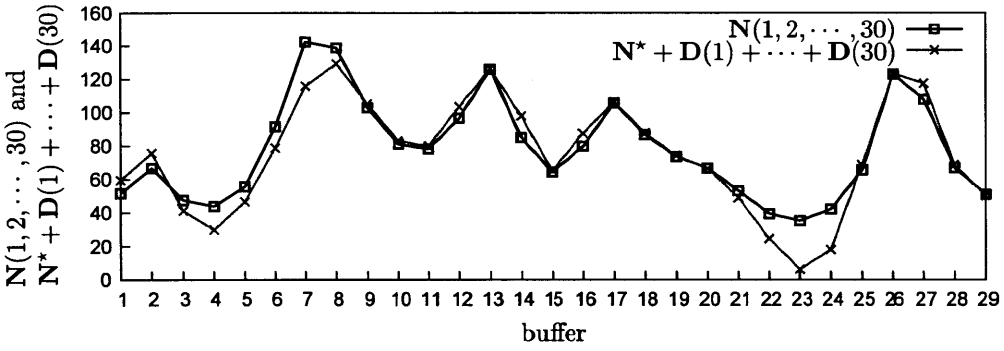


Figure 9-28: Comparison of buffer distributions, Extreme case example 4

The four examples presented in this section show the cases where the additive property is less accurate. If there are a couple of bottleneck or anti-bottleneck machines placed close enough to each other, the additive property may not account for the interaction among those machines and therefore it is less accurate.

9.5 Summary

In this chapter, we study an additive property in long line optimization. The property is that the effect of a set of local bottlenecks on the optimal buffer distribution is approximately the same as the sum of the effects of each local bottleneck by itself. A similar property is observed with a mixture of local bottlenecks and local anti-

bottlenecks. Both heuristic explanations and numerical examples are provided to demonstrate the additive property. However, when there is an interaction between the effects of two cause machines on the optimal buffer distribution, the additive property may is less accurate as it does not account for such an interaction.

The additive property in long line optimization is an important observation, especially when it is considered together with the segmentation method for long line optimization (Chapter 8). Instead of optimizing the original long line, the segmentation method divides it into several short lines, optimizes these short lines separately, and combines the optimal buffer distributions to find an approximately optimal buffer distribution of the original line. Suppose a line is already set up and its buffer allocation is already optimized accordingly. Now if a certain machine is replaced with a different one, instead of optimizing the entire line with the new machine, we can optimize a set of segmented short lines according to the segmentation method. In addition, because of the additive property, we know in advance that the new machine will not change the optimal sizes of the buffers that are not adjacent to it. Therefore, we can only optimize the specific segmented line that contains the new machine. Consequently, because of the segmentation method and the additive property, instead of re-optimizing the original long line with the new machine, we could simply optimize a much shorter line that contains the new machine. This will dramatically reduce the computer effort and time, while assuring the optimization accuracy.

Given the potential importance of the additive property, future research should be devoted to the formal analysis of this phenomenon and to its application to efficient line design.

Chapter 10

Conclusions and Future Work

10.1 Conclusions

With the goal of developing efficient buffer design algorithms for production system profit maximization, this thesis studies three major topics about production lines and closed-loop systems, as well as investigates methods and properties that facilitate long line optimization.

We define the profit of a production line as the revenue associated with the production rate minus the buffer space cost and the average inventory holding cost. In addition, in our problem, we assume that manufacturing processes and machines have already been chosen and therefore our only decision variables are buffer sizes (for production lines)[1]. In this thesis, we present an accurate, fast, and reliable algorithm for maximizing profits through buffer space optimization for production lines, and extend the algorithm to closed-loop systems and production lines with an addition maximum part waiting time constraint. The major difficulty in this research is the presence of nonlinear components in both the objective function and the constraint of the optimization problem, because the production rate as well as average inventory levels are nonlinear functions of buffer sizes. A nonlinear programming approach is adopted to solve the optimization problem. Numerical experiments are provided to show the accuracy and efficiency of the proposed algorithms for the three topics.

[1]For closed-loop systems, in addition to buffer sizes, the loop invariant is also the decision variable.

Finally, a segmentation method and an additive property of line optimization are analyzed. They enable us to optimize long lines fast and accurately.

The key contributions of this thesis and their importance are highlighted here:

- **A discussion about the qualitative properties of the production rate $P(\mathbf{N})$.**

  The continuity, monotonicity, and concavity of $P(\mathbf{N})$ are studied. The continuity enables us to treat $P(\mathbf{N})$ as a continuous function of $\mathbf{N}$ and facilitates the application of a gradient method in the profit maximization algorithm. In addition, the monotonicity and concavity assumption of $P(\mathbf{N})$ are also used in deriving the optimization algorithm.

- **A discussion about the behavior of the average buffer levels in multiple stage tandem lines as functions of the buffer sizes.**

  We start by focusing on three-machine two-buffer lines, for which we break up the line at each buffer into a single machine and a two-machine, one buffer line and we compare production rates of each. This leads to the observation that there are five possible types of three-machine lines, each with a specific qualitative behavior. For each type, the average inventories of Buffers B_1 and B_2 and the profit of the line are studied as functions of the sizes of the two buffers. The results show that for some types, the average inventories and the profit of the line are neither convex nor concave functions of buffer sizes. However, for each feasible type, we observe that no matter whether the profit of the line is a concave/convex function of buffer sizes or not, there is a unique global optimal solution of buffer allocation that maximizes the profit. This is consistent with Schor's argument on the uniqueness of the maximum of $J(\mathbf{N})$ (Schor 1995 and Gershwin and Schor 2000). Therefore, a gradient method is appropriate to solve the unconstrained profit maximization problem without the production rate constraint.

- **An efficient buffer design algorithm for production line profit maximization subject to a production rate constraint.**

An effective algorithm for maximizing profits through buffer size optimization for production lines is developed. Both buffer space cost and average inventory cost with distinct cost coefficients for different buffers are considered, as well as a nonlinear production rate constraint. To solve the problem, a corresponding unconstrained problem is introduced and a nonlinear programming approach is adopted. The algorithm is proved theoretically by the KKT conditions of nonlinear programming. The proposed algorithm are applied to the three production line models (i.e., the deterministic single failure mode line model of Gershwin 1994, the deterministic multiple failure mode model of Tolio and Matta 1998, and the continuous multiple failure mode model of Levantesi et al. 2003). To study the accuracy and efficiency of the algorithm, we provide numerical experiments on randomly generated lines and compare it with many existing algorithms for solving a special case (i.e., Problem (4.31)) of the constrained problem.

- **Two modifications that improve the accuracy of the existing evaluation algorithm for closed-loop systems and its extension to single open-loop systems.**

  A closed-loop production system is a system in which a constant amount of material flows through a single fixed cycle of work stations and storage buffers. While an evaluation method already exists (Gershwin and Werner 2007) which is accurate for Buzacott systems, it produces results that are discontinuous as functions of certain key design parameters (the Batman effect). These discontinuities are detrimental to the performance of optimization methods. We present two modifications that improve the accuracy of this method, which is based on the decomposition of such systems. Analytical solutions for the evaluation of two new special types of two-machine one-buffer building blocks from the decomposition are developed. Numerical experiments are provided to show the improvement of the evaluation accuracy as compared with the existing algorithm. The Batman effect is eliminated with these modifications.

- **Extension of the profit maximization algorithm for production lines to closed-loop systems.**

  We extend the profit maximization algorithm developed for lines to closed-loop systems. Given the fact that a loop system may have multiple profit maxima (as shown in Chapter 6), the scope of the algorithm is addressed. The performance of the algorithm is shown by studying a set of three-machine and four-machine closed-loop systems. Numerical experiments demonstrate the accuracy of the proposed algorithm.

- **An analytical formulation of the part waiting time distribution in a Buzacott two-machine line and the production line profit maximization algorithm with the additional part waiting time constraint.**

  An analytical formulation for the part waiting time distribution in a Buzacott two-machine one-buffer transfer line is presented. The numerical solution is tested with Little's Law (Little 1961). Numerical experiments are provided to illustrate the accuracy of the solution as it is compared with simulation. This distribution and the decomposition approach allow us to approximately (yet accurately) compute the part waiting time distribution in the given buffer of a long line. The profit maximization algorithm developed in Chapter 4 is extended to cover the additional maximum part waiting time constraint.

- **A segmentation method for long line optimization.**

  A segmentation method for long line optimization is developed. Instead of optimizing the original long line, the segmentation method divides it into several short lines, optimizes these short lines separately, and combines the optimal buffer distributions to find an approximately optimal buffer distribution of the original line. This method reduces the computer time for long line optimization dramatically. Both heuristic explanations and numerical experiments are provided to show the accuracy and speed of the method.

- **A discussion about an additive property in long line optimization.**

The additive property we observe is that the effect of a set of local bottlenecks on the optimal buffer distribution (in which profit is maximized subject to a production rate constraint) is approximately the same as the sum of the effects of each local bottleneck by itself. A similar property is observed with a mixture of bottlenecks and "anti-bottlenecks". Both heuristic explanations and numerical experiments are provided to demonstrate the property. Some limitations on the additive property are also examined.

The additive property in long line optimization is an important observation, especially when it is applied together with the segmentation method. Suppose that a line is already set up and its buffer allocation is already optimized accordingly. Now if a machine is replaced with a different one, instead of optimizing the entire line with the new machine (which can be time-consuming), we can just optimize a small portion of the line that contains the new machine according to the segmentation method. In addition, because of the additive property, we know in advance that the new machine will not change the optimal sizes of the buffers that are far enough from it. Therefore, with the segmentation method and the additive property, we can dramatically reduce the computer effort and time for the optimization of long lines.

10.2 Future Work

There are several research directions to which we can extend our research and algorithms in the future.

1. Tree structured Assembly/Disassembly (A/D) systems.

   Tree structured A/D systems are extensions of lines in which assembly and disassembly take place. The first extension of the profit maximization algorithm could be to acyclic A/D systems.

2. Systems with machine and/or buffer location selection.

   In the design of a production line, each operation has various machine choices. Each machine has its own distinct parameters and cost. Different machine

combinations lead to different production line performance and cost. Nahas et al. (2009) develop a method to select machines and buffers in unreliable series-parallel production lines to maximize the production rate subject to a total cost constraint. We want to decide which machine to choose for each operation as we select buffers to maximize the profit for the line. In addition, we can also study the locations of buffers on the performance of the production lines.

3. Larger loop systems.

   We have shown in Chapter 5 that with the two modifications, we reduce the Batman effect significantly. However, there are still improvement opportunities for loop evaluation since small bumps may still appear in the production line curve and they prevent the algorithm from finding the optimal solution. One possible approach to further smooth the production rate curve is discussed here, with the help of Figure 10-1.

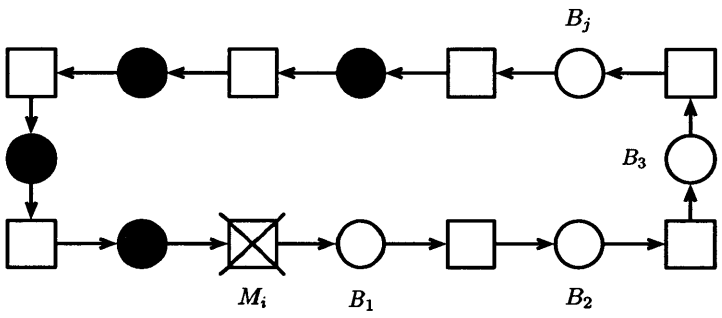


Figure 10-1: Part traveling time in a closed-loop system

Suppose that Machine M_i fails in the closed-loop system, and it causes B_j to be empty. For the building block that contains B_j in the decomposition approach of loop evaluation, the upstream machine $M^u(B_j)$ would fail to a remote failure mode that corresponds to the actual failure mode of M_i that leads to an empty B_j. In the evaluation algorithm of Gershwin and Werner (2007), $M^u(B_j)$ is repaired once the actual failure of M_i is repaired with probability r_i, and it will start operation and add a part into B_j immediately. In other words, the

repair probability of that remote failure mode of $M^u(B_j)$ is r_i. However, the reality is that even after M_i is repaired, the first part produced by M_i needs to travel through Buffers B_1, B_2, and B_3 until it reaches B_j. In other words, B_j needs to wait for at least three time units (provided no failures occur in the machines between B_1 and B_j) until a part is added to it. Therefore, the evaluation algorithm overestimates the production rate of the loop since it does not consider the part traveling time. This is indicated from the numerical experiments of Chapter 5, where the evaluation results of the analytical solution (with the two modifications) are always higher than those from simulation. As a result, if we want to further improve the accuracy of loop evaluation, r_i should not be used as the repair probability of that remote failure mode of $M^u(B_j)$. In fact, the repair probability is not a geometric distribution, and the correct value of the repair probability has to be determined in the decomposition approach.

Once we further reduce the bumps, the loop optimization algorithm of Chapter 6 can be applied to larger closed-loop systems and extended to multiple loop systems.

4. A formulation of the production line profit maximization problem that deals with the maximum part waiting time directly.

   In Chapter 7, we study the production line profit maximization problem subject to both a production rate constraint and a maximum part waiting time constraint (i.e., $\mathbf{p}(T(\mathbf{N}) \leq W_i) \geq 1 - \alpha$). However, due to the lack of a closed form expression of the probability above, we transform the original problem to a transformed problem and solve it iteratively. The accuracy of this alternative approach is verified with a surface search method. However, it is desirable to solve the original problem directly (with or without a closed form of the probability distribution of part waiting time).

5. Systems with part lead time constraints.

   Part lead time is of great interest as it is the time a part spends in the entire production line. Therefore, it will be valuable to study the total time that a part

spends in the whole line and to impose such a constraint in the optimization problem. We expect that the total time distribution is not simply a summation of the part waiting time distributions in all buffers. There may be correlation among different waiting times in different buffers. Even if it were a summation, we would have to deal which convolutions. Some preliminary discussion about the lead time distribution for the deterministic line model is available in Tan (2002).

6. Systems with set-up cost for buffers.

   This means whenever we decide to establish a buffer between two machines, we introduce a fixed buffer set-up cost. After the buffer is established, the buffer space cost will be proportional to its size. So, in this case, buffer space cost will be 0 if $N_i < N_{\min}$ or $a_i + b_i N_i$ if $N_i \geq N_{\min}$ for Buffer B_i.

7. Systems with quality control.

   By taking account of quality control, we assume that machines generate both good parts and bad parts. Unfortunately, buffers delay the inspection of bad parts. Consider a two-machine line. Suppose parts are inspected only after the second machine. So, when the first machine begins to generate bad parts, we will not know it immediately if there are still some good parts in the buffer. We will only know that the first machine is generating bad parts after all good parts in the buffer are processed by the second machine and it begins to process the first bad part. During that delay, the first machine could have generated more bad parts. Thus, in this case, buffers bring potential delay to inspection, which reduces the production rate and the profit of the line. Kim and Gershwin (2005) point out that in the case of our example above, an increase of buffer size could either increase or decrease the production rate of good parts for different lines. The quality issue will change the nature of the profit maximization problem, since the bad quality parts should be considered as cost. Therefore, it is desirable to study how buffers should be allocated to maximize the profit of the line associated with good quality parts.

8. Systems with additional buffer space constraints.

   In some cases, we may encounter additional buffer space constraints of the form $\sum_i g_{ij} N_i \leq h_j$. For instance, consider a production line, in which two buffers, say Buffers B_5 and B_6, are in the clean room. Due to the high cost of buffer space in the clean room, there is a maximum total space constraint for B_5 and B_6 as $N_5 + N_6 \leq C$, where C is a constant. This brings additional constraints to our problem, so we need to make necessary modification to our algorithm so that it can apply to the new problem.

These extensions are key complements to make production line models more practical for modeling actual production lines and more general manufacturing systems in factories. A good understanding of the behavior of the proposed algorithms on those extensions would be valuable in future production line design.

Appendix A

The Continuous Variable Version of the Analytical Solution of the Deterministic Two-Machine Line of Gershwin (1994)

The continuous variable version of the solution of the deterministic two-machine line of Gershwin (1994) is presented in this section. Suppose we have a two-machine line with parameters r_1, p_1, r_2, and p_2, and the buffer size is N. In the two-machine, the state of the system is $s = (n, \alpha_1, \alpha_2)$ where n $(0 \leq n \leq N)$ is the buffer level and α_i $(\alpha_i = 0, 1)$ is the state of Machine M_i, $i = 1, 2$. $\mathbf{p}(n, \alpha_1, \alpha_2)$ stands for the steady-state probability of that state. Gershwin (1994) shows that the steady-state probability distribution is

$$\mathbf{p}(0,0,0) = 0, \tag{A.1}$$

$$\mathbf{p}(0,0,1) = CX\frac{r_1 + r_2 - r_1 r_2 - r_1 p_2}{r_1 p_2}, \tag{A.2}$$

$$\mathbf{p}(0,1,0) = 0, \tag{A.3}$$

$$\mathbf{p}(0,1,1) = 0, \tag{A.4}$$

$$\mathbf{p}(1,0,0) = CX, \tag{A.5}$$

$$\mathbf{p}(1,0,1) = CXY_2, \tag{A.6}$$

$$\mathbf{p}(1,1,0) = 0, \tag{A.7}$$

$$\mathbf{p}(1,1,1) = \frac{CX}{p_2}\frac{r_1+r_2-r_1r_2-r_1p_2}{p_1+p_2-p_1p_2-r_1p_2}, \tag{A.8}$$

$$\mathbf{p}(N-1,0,0) = CX^{N-1}, \tag{A.9}$$

$$\mathbf{p}(N-1,0,1) = 0, \tag{A.10}$$

$$\mathbf{p}(N-1,1,0) = CX^{N-1}Y_1, \tag{A.11}$$

$$\mathbf{p}(N-1,1,1) = \frac{CX^{N-1}}{p_1}\frac{r_1+r_2-r_1r_2-p_1r_2}{p_1+p_2-p_1p_2-p_1r_2}, \tag{A.12}$$

$$\mathbf{p}(N,0,0) = 0, \tag{A.13}$$

$$\mathbf{p}(N,0,1) = 0, \tag{A.14}$$

$$\mathbf{p}(N,1,0) = CX^{N-1}\frac{r_1+r_2-r_1r_2-p_1r_2}{p_1r_2}, \tag{A.15}$$

$$\mathbf{p}(N,1,1) = 0, \tag{A.16}$$

$$\mathbf{p}(n,\alpha_1,\alpha_2) = CX^nY_1^{\alpha_1}Y_2^{\alpha_2}, 2 \le n \le N-2; \alpha_1 = 0,1; \alpha_2 = 0,1, \tag{A.17}$$

where C is a positive normalizing constant and

$$Y_1 = \frac{r_1+r_2-r_1r_2-r_1p_2}{p_1+p_2-p_1p_2-p_1r_2}, \tag{A.18}$$

$$Y_2 = \frac{r_1+r_2-r_1r_2-p_1r_2}{p_1+p_2-p_1p_2-r_1p_2}, \tag{A.19}$$

$$X = \frac{Y_2}{Y_1}. \tag{A.20}$$

For convenience, we define internal states as those states in which $2 \le n \le N-2$; while boundary states as those states in which $n = 0$, 1, $N-1$, or N. We further let p_B be the summation of the steady-state probabilities of all boundary states and p_I be the summation of the steady-state probabilities of all internal states. Then,

$$
\begin{aligned}
p_B &= \sum_{\text{all boundary states}} \mathbf{p}(n, \alpha_1, \alpha_2) \\
&= CX\frac{r_1 + r_2 - r_1 r_2 - r_1 p_2}{r_1 p_2} + CX + CXY_2 + \frac{CX}{p_2}\frac{r_1 + r_2 - r_1 r_2 - r_1 p_2}{p_1 + p_2 - p_1 p_2 - r_1 p_2} \\
&\quad + CX^{N-1} + CX^{N-1}Y_1 + \frac{CX^{N-1}}{p_1}\frac{r_1 + r_2 - r_1 r_2 - p_1 r_2}{p_1 + p_2 - p_1 p_2 - p_1 r_2} \\
&\quad + CX^{N-1}\frac{r_1 + r_2 - r_1 r_2 - p_1 r_2}{p_1 r_2}.
\end{aligned} \tag{A.21}
$$

In addition, we calculate p_I in the continuous variable version by

$$
p_I = \sum_{\text{all internal states}} \mathbf{p}(n, \alpha_1, \alpha_2) = \begin{cases} C\dfrac{X^{N-1} - X^2}{X - 1}(1 + Y_1)(1 + Y_2) & \text{if } X \neq 1, \\ C(N-3)(1 + Y_1)(1 + Y_2) & \text{if } X = 1. \end{cases} \tag{A.22}
$$

The normalizing constant C can be found by condition $p_B + p_I = 1$. A smarter way to find C is also provided in Section 2.2.1. The production rate of the line can be calculated by

$$
\begin{aligned}
P(N) &= \frac{r_1}{r_1 + p_1}(1 - p_b) \\
&= \frac{r_1}{r_1 + p_1}\Big(\mathbf{1} - \mathbf{p}(N, 1, 0)\Big) \\
&= \frac{r_1}{r_1 + p_1}\left(1 - CX^{N-1}\frac{r_1 + r_2 - r_1 r_2 - p_1 r_2}{p_1 r_2}\right)
\end{aligned} \tag{A.23}
$$

where $p_b = \mathbf{p}(N, 1, 0)$ is the probability of blocking, and

$$C = \frac{\dfrac{r_2+p_2}{r_2} - \dfrac{r_1+p_1}{r_1}}{\dfrac{r_2+p_2}{r_2} X^{N-1} \dfrac{r_1+r_2-r_1r_2-p_1r_2}{p_1r_2} - \dfrac{r_1+p_1}{r_1} X \dfrac{r_1+r_2-r_1r_2-r_1p_2}{r_1p_2}}.$$

assuming that $X \neq 1$. (See the expression of C when $X = 1$ in Section 2.2.1.)

Finally, let us consider how to calculate the average inventory for the line. We compute the average inventory for the internal states, denoted by $\bar{n}_I$, as

$$\begin{aligned}
\bar{n}_I &= \sum_{\text{all internal states}} n\mathbf{p}(n, \alpha_1, \alpha_2) \\
&= \begin{cases} C\dfrac{X\left(-2X + (N-1)X^{N-2} - \dfrac{X^{N-1}-X^2}{X-1}\right)}{X-1}(1+Y_1)(1+Y_2) & \text{if } X \neq 1, \\ \dfrac{1}{2}CN(N-3)(1+Y_1)(1+Y_2) & \text{if } X = 1. \end{cases}
\end{aligned} \tag{A.24}$$

Then, to calculate the average inventory for the line, we need to consider both internal states and boundary states whose steady-state probability is non-zero and n is non-zero. Then, the average inventory of the line is calculated as

$$\begin{aligned}
\bar{n} &= \sum_{\text{all states}} n\mathbf{p}(n, \alpha_1, \alpha_2) \\
&= \mathbf{p}(1,0,0) + \mathbf{p}(1,0,1) + \mathbf{p}(1,1,1) + \bar{n}_I \\
&\quad + (N-1)\Big(\mathbf{p}(N-1,0,0) + \mathbf{p}(N-1,1,0) + \mathbf{p}(N-1,1,1)\Big) + N\mathbf{p}(N,1,0).
\end{aligned} \tag{A.25}$$

In the implementation of the computer program, the condition $X = 1$ is replaced by $|X - 1| \leq \delta$, where δ is a very small non-zero positive value; while the condition $X \neq 1$ is replaced by $|X - 1| > \delta$.

Appendix B

Proof of the Assertion in Section 4.2.1 for the Case in Which Some $N_i^\star = N_{\min}$

We provide the proof of the assertion in Section 4.2.1 for the case in which some $N_i^\star = N_{\min}$. Recall that the assertion states that:

Assertion The constrained problem

$$\max_{\mathbf{N}} \quad J(N_1, \cdots, N_{k-1}) = A'P(N_1, \cdots, N_{k-1}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i$$

$$\text{subject to} \quad P(N_1, \cdots, N_{k-1}) \geq \hat{P},$$

$$N_i \geq N_{\min}, \forall i = 1, \cdots, k-1$$

has the same solution for all A' in which the solution of the unconstrained problem (4.5) has $P' < \hat{P}$.

Let B be the set of i such that $\{i | N_i^\star = N_{\min}\}$. Hence, $N_i^\star = N_{\min}, \forall i \in B$, while $N_i^\star > N_{\min}, \forall i \notin B$. In this case, those $N_i^\star$ equal to $N_{\min}$ are on the boundary of the feasible region of the optimal solution. By condition (4.15), we know that

$\mu_i^\star = 0, \forall i \notin B$ and $i \neq 0$. Hence, we simplify the KKT conditions (4.12) to (4.15) to

$$-\begin{pmatrix} \frac{\partial J(\mathbf{N}^\star)}{\partial N_1} \\ \frac{\partial J(\mathbf{N}^\star)}{\partial N_2} \\ \vdots \\ \frac{\partial J(\mathbf{N}^\star)}{\partial N_{k-1}} \end{pmatrix} - \mu_0^\star \begin{pmatrix} \frac{\partial P(\mathbf{N}^\star)}{\partial N_1} \\ \frac{\partial P(\mathbf{N}^\star)}{\partial N_2} \\ \vdots \\ \frac{\partial P(\mathbf{N}^\star)}{\partial N_{k-1}} \end{pmatrix} - \sum_{i \in B} \mu_i^\star e_i = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{B.1}$$

where

$$e_i = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

in which the 1 is in the ith position.

$$\mu_0^\star \left(\hat{P} - P(\mathbf{N}^\star) \right) = 0, \tag{B.2}$$

$$\mu_i^\star (N_{\min} - N_i^\star) = 0, \forall i = 1, \cdots, k-1 \tag{B.3}$$

where $\mu_i^\star \geq 0, \forall i = 0, \cdots, k-1$. We first argue that not all $N_i^\star, i \notin B$ satisfy $\partial J(\mathbf{N}^\star)/\partial N_i = 0$. This is because if $\partial J(\mathbf{N}^\star)/\partial N_i = 0$ for all $N_i^\star, i \notin B$ and all other $N_i^\star = N_{\min}, i \in B$, then the solution of the constrained problem $(N_1^\star, \cdots, N_{k-1}^\star)$ is equivalent to the solution of the unconstrained problem, since it is just constrained by the buffer size constraint. This is a conflict. Therefore, it is not possible for all $N_i^\star, i \notin B$ to satisfy $\partial J(\mathbf{N}^\star)/\partial N_i = 0$. So, there exists some $\hat{i}'$s for which $N_{\hat{i}}^\star > N_{\min}$ and $\partial J(\mathbf{N}^\star)/\partial N_{\hat{i}} \neq 0$. Since $N_{\hat{i}}^\star > N_{\min}$, by condition (B.3), we know $\mu_{\hat{i}}^\star = 0$. Thus, $\mu_0^\star \neq 0$ since otherwise condition (B.1) would be violated. Hence, by condition (B.2), we know that the optimal solution $\mathbf{N}^\star$ satisfies $P(\mathbf{N}^\star) = \hat{P}$. In this case, there are

more than one active inequality constraints: the production rate constraint $g_0(\mathbf{N})$ and buffer size constraints $g_i(\mathbf{N}), \forall i \in B$. It is not hard to show that $\nabla g_0(\mathbf{N}^\star)$ and $\nabla g_i(\mathbf{N}^\star), \forall i \in B$ are linearly independent[1] so the optimal solution is regular.

Replacing $\mu_0^\star$ by $\mu_0 > 0$ and $\mu_i^\star, \forall i \in B$ by $\mu_i > 0, \forall i \in B$ in constraint (B.1) gives

$$-\nabla J(\bar{\mathbf{N}}) + \mu_0 \nabla \left(\hat{P} - P(\bar{\mathbf{N}})\right) + \sum_{i \in B} \mu_i \nabla (N_{\min} - \bar{N}_i) = 0, \tag{B.4}$$

where $\bar{\mathbf{N}}$ is the unique solution. Note that $\bar{\mathbf{N}}$ is exactly the optimal solution of the following optimization problem

$$\begin{aligned} \min_{\mathbf{N}} \quad -\bar{J}(\mathbf{N}) &= -J(\mathbf{N}) + \mu_0 \left(\hat{P} - P(\mathbf{N})\right) + \sum_{i \in B} \mu_i (N_{\min} - N_i) \\ \text{subject to} \quad N_{\min} - N_i &\leq 0, \forall i = 1, \cdots, k-1, \\ N_i &= N_{\min}, \forall i \in B, \end{aligned} \tag{B.5}$$

which is equivalent to

$$\begin{aligned} \max_{\mathbf{N}} \quad \bar{J}(\mathbf{N}) &= J(\mathbf{N}) - \mu_0 \left(\hat{P} - P(\mathbf{N})\right) \\ \text{subject to} \quad N_{\min} - N_i &\leq 0, \forall i = 1, \cdots, k-1, \\ N_i &= N_{\min}, \forall i \in B, \end{aligned} \tag{B.6}$$

[1]This is because $\nabla g_0(\mathbf{N}^\star) = \nabla(\hat{P} - P(\mathbf{N}^\star))$ has all negative component due to the monotonicity of $P(\mathbf{N})$, but a linear combination of $\nabla g_i(\mathbf{N}^\star) = -e_i, \forall i \in B$ cannot generate all corresponding non-zero components of $\nabla g_0(\mathbf{N}^\star)$ since not all buffers belong to set B.

or

$$\begin{aligned} \max_{\mathbf{N}} \quad \bar{J}(\mathbf{N}) &= (A+\mu_0)P(\mathbf{N}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i \\ \text{subject to} \quad N_i &\geq N_{\min}, \forall i = 1, \cdots, k-1, \\ N_i &= N_{\min}, \forall i \in B, \end{aligned} \tag{B.7}$$

or, finally,

$$\begin{aligned} \max \quad \bar{J}(\mathbf{N}) &= A'P(\mathbf{N}) - \sum_{i=1}^{k-1} b_i N_i - \sum_{i=1}^{k-1} c_i \bar{n}_i \\ \text{subject to} \quad N_i &\geq N_{\min}, \forall i = 1, \cdots, k-1, \\ N_i &= N_{\min}, \forall i \in B, \end{aligned} \tag{B.8}$$

where $A' = A + \mu_0$. Again, this is exactly the unconstrained problem in which A is replaced by A', and $\bar{\mathbf{N}}$ is its optimal solution with the fact that $\bar{N}_i = N_{\min}, \forall i \in B$.

Appendix C

The Continuous Variable Version of the Analytical Solution of the Deterministic Two-Machine Line of Tolio et al. (2002)

The continuous variable version of the solution of the deterministic multiple-failure mode two-machine line of Tolio et al. (2002) is presented in this section. Suppose we have a two-machine line where the upstream machine has s failure modes while the downstream machines has t failure modes. The parameters of the upstream machine are r^{u_i} and $p^{u_i}, i = 1, \cdots, s$ and the parameters of the downstream machine are r^{d_j} and $p^{d_j}, j = 1, \cdots, t$. The buffer size is N. The state of the system is $s = (n, \alpha_1, \alpha_2)$ where n $(0 \leq n \leq N)$ is the buffer level and α_1 and α_2 are the states of Machines M_1 and M_2, respectively. In particular, $\alpha_1 = 1$ if M_1 is up, while $\alpha_1 = u_i$ if M_1 is down in failure mode i. Similarly, $\alpha_2 = 1$ if M_2 is up, while $\alpha_1 = d_j$ if M_2 is down in failure mode j. $\mathbf{p}(n, \alpha_1, \alpha_2)$ stands for the steady-state probability of that state. In addition, internal states are defined as those states in which $2 \leq n \leq N - 2$; while boundary states are defined as those states in which $n = 0, 1, N - 1$ or N. Tolio et al. (2002) show that the steady-state probability distribution for all internal states is

$$\mathbf{p}(n,1,1) = \sum_{m=1}^{R} C_m X_m^n,$$

$$\mathbf{p}(n,1,d_j) = \sum_{m=1}^{R} C_m X_m^n D_{j,m}, \quad j = 1,\cdots,t,$$

$$\mathbf{p}(n,u_i,1) = \sum_{m=1}^{R} C_m X_m^n U_{i,m}, \quad i = 1,\cdots,s,$$

$$\mathbf{p}(n,u_i,d_j) = \sum_{m=1}^{R} C_m X_m^n U_{i,m} D_{j,m}, \quad i = 1,\cdots,s, j = 1\cdots,t$$

where $R = s + t$, $C_m, m = 1,\cdots,R$ are normalizing constants, and $X_m, U_{i,m}$, and $D_{j,m}, m = 1,\cdots,R, i = 1,\cdots,s, j = 1,\cdots,t$ are system parameters computed by machine parameters (see Tolio et al. 2002 for their definitions). In addition, for $i = 1,\cdots,s$ and $j = 1,\cdots,t$, the steady-state probability distribution for boundary states is

$$\begin{aligned}
\mathbf{p}(0,1,d_j) &= 0, \\
\mathbf{p}(0,u_i,d_j) &= 0, \\
\mathbf{p}(0,1,1) &= 0, \\
\mathbf{p}(0,u_i,1) &= \frac{p^{u_i}(1-P^D)}{r^{u_i}p^{d_j}} \sum_{m=1}^{R} C_m X_m \frac{D_{j,m}}{K_m} + \frac{1-r^{u_i}}{r^{u_i}} \sum_{m=1}^{R} C_m \frac{U_{i,m}}{K_m}, \\
\mathbf{p}(1,1,1) &= \frac{1}{p^{d_j}} \sum_{m=1}^{R} C_m X_m \frac{D_{j,m}}{K_m}, \\
\mathbf{p}(1,1,d_j) &= 0,
\end{aligned}$$

$$\mathbf{p}(1, u_i, 1) = \sum_{m=1}^{R} C_m X_m U_{i,m},$$

$$\mathbf{p}(1, u_i, d_j) = \sum_{m=1}^{R} C_m X_m U_{i,m} D_{j,m},$$

$$\mathbf{p}(N-1, 1, 1) = \frac{1}{p^{u_i}} \sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m} K_m,$$

$$\mathbf{p}(N-1, 1, d_j) = \sum_{m=1}^{R} C_m X_m^{N-1} D_{j,m},$$

$$\mathbf{p}(N-1, u_i, 1) = 0,$$

$$\mathbf{p}(N-1, u_i, d_j) = \sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m} D_{j,m},$$

$$\mathbf{p}(N, 1, 1) = 0,$$

$$\mathbf{p}(N, 1, d_j) = \frac{p^{d_j}(1-P^U)}{r^{d_j} p^{u_i}} \sum_{m=1}^{R} C_m X_m^{N-1} U_{i,m} K_m + \frac{1-r^{d_j}}{r^{d_j}} \sum_{m=1}^{R} C_m X_m^{N} D_{j,m} K_m,$$

$$\mathbf{p}(N, u_i, 1) = 0,$$

$$\mathbf{p}(N, u_i, d_j) = 0.$$

According to Tolio et al. (2002), the production rate of the line can be computed by

$$\begin{aligned} P(N) &= \mathbf{p}(\alpha_1 = 1 \text{ and } n < N) \\ &= \sum_{n=0}^{N-1} \left[\mathbf{p}(n, 1, 1) + \sum_{j=1}^{t} \mathbf{p}(n, 1, d_j) \right]. \end{aligned} \tag{C.1}$$

With the steady-state probabilities of all states, (C.1) can be further expressed by

$$P(N) = \mathbf{p}(1,1,1) + \sum_{n=2}^{N-2} \mathbf{p}(n,1,1) + \sum_{n=2}^{N-2} \sum_{j=1}^{t} \mathbf{p}(n,1,d_j) + \mathbf{p}(N-1,1,1) + \sum_{j=1}^{t} \mathbf{p}(N-1,1,d_j). \tag{C.2}$$

To evaluate $P(N)$ with a non-integer N, we need to find analytical expressions for the second and the third terms on the right hand side of (C.2). We consider them separately.

$$\begin{aligned} \sum_{n=2}^{N-2} \mathbf{p}(n,1,1) &= \sum_{n=2}^{N-2} \sum_{m=1}^{R} C_m X_m^n \\ &= \sum_{m=1}^{R} C_m \left(\sum_{n=2}^{N-2} X_m^n \right). \end{aligned} \tag{C.3}$$

For $m = 1, \cdots, R$, we need to treat $X_m \neq 1$ and $X_m = 1$ differently. Let $\mathcal{M}$ be the set of m such that $\{m | X_m \neq 1\}$. Therefore, the complementary of $\mathcal{M}$ is $\mathcal{M}^c = \{m | X_m = 1\}$. It can be seen that

$$\sum_{n=2}^{N-2} X_m^n = \begin{cases} \dfrac{X_m^{N-1} - X_m^2}{X_m - 1} & \text{if } X_m \neq 1, \\ \\ N-3 & \text{if } X_m = 1. \end{cases} \tag{C.4}$$

Therefore,

$$\begin{aligned} \sum_{n=2}^{N-2} \mathbf{p}(n,1,1) &= \sum_{m=1}^{R} C_m \left(\sum_{n=2}^{N-2} X_m^n \right) \\ &= \sum_{m \in \mathcal{M}} C_m \left(\frac{X_m^{N-1} - X_m^2}{X_m - 1} \right) + \sum_{m \in \mathcal{M}^c} C_m (N-3). \end{aligned} \tag{C.5}$$

Similarly, for the third term on the right hand side of (C.2) can be computed by

$$\sum_{n=2}^{N-2}\sum_{j=1}^{t}\mathbf{p}(n,1,d_j) = \sum_{m=1}^{R}C_m\sum_{j=1}^{t}D_{j,m}\left(\sum_{n=2}^{N-2}X_m^n\right)$$

$$= \sum_{m\in\mathcal{M}}C_m\sum_{j=1}^{t}D_{j,m}\left(\frac{X_m^{N-1}-X_m^2}{X_m-1}\right)+\sum_{m\in\mathcal{M}^c}C_m\sum_{j=1}^{t}D_{j,m}(N-3). \tag{C.6}$$

Therefore, the production rate $P(N)$ in the continuous variable version is

$$P(N) = \frac{1}{p^{d_j}}\sum_{m=1}^{R}C_mX_m\frac{D_{j,m}}{K_m}+\sum_{m\in\mathcal{M}}C_m\left(\frac{X_m^{N-1}-X_m^2}{X_m-1}\right)+\sum_{m\in\mathcal{M}^c}C_m(N-3)$$

$$+\sum_{m\in\mathcal{M}}C_m\sum_{j=1}^{t}D_{j,m}\left(\frac{X_m^{N-1}-X_m^2}{X_m-1}\right)+\sum_{m\in\mathcal{M}^c}C_m\sum_{j=1}^{t}D_{j,m}(N-3)$$

$$+\frac{1}{p^{u_i}}\sum_{m=1}^{R}C_mX_m^{N-1}U_{i,m}K_m+\sum_{j=1}^{t}\sum_{m=1}^{R}C_mX_m^{N-1}D_{j,m}. \tag{C.7}$$

Note that (C.7) does not require N to be integer and therefore it can be evaluated with non-integer N.

Next, we consider the average inventory, which can be calculated by

$$\bar{n}(N) = \mathbf{p}(1,1,1)+\sum_{j=1}^{t}\mathbf{p}(1,1,d_j)+\sum_{i=1}^{s}\mathbf{p}(1,u_i,1)+\sum_{i=1}^{s}\sum_{j=1}^{t}\mathbf{p}(1,u_i,d_j)+\bar{n}_I(N)$$

$$+(N-1)\left(\mathbf{p}(N-1,1,1)+\sum_{j=1}^{t}\mathbf{p}(N-1,1,d_j)+\sum_{i=1}^{s}\sum_{j=1}^{t}\mathbf{p}(N-1,u_i,d_j)\right)$$

$$+N\sum_{j=1}^{t}\mathbf{p}(N,1,d_j), \tag{C.8}$$

where $\bar{n}_I(N)$ represents the portion of the average inventory contributed by all internal

states and therefore it is

$$\begin{aligned}\bar{n}_I(N) &= \sum_{n=2}^{N-2} n\left[\mathbf{p}(n,1,1)+\sum_{j=1}^{t}\mathbf{p}(n,1,d_j)+\sum_{i=1}^{s}\mathbf{p}(n,u_i,1)+\sum_{i=1}^{s}\sum_{j=1}^{t}\mathbf{p}(n,u_i,d_j)\right] \\ &= \sum_{n=2}^{N-2} n\mathbf{p}(n,1,1)+\sum_{n=2}^{N-2}\sum_{j=1}^{t} n\mathbf{p}(n,1,d_j)+\sum_{n=2}^{N-2}\sum_{i=1}^{s} n\mathbf{p}(n,u_i,1) \\ &\quad +\sum_{n=2}^{N-2}\sum_{i=1}^{s}\sum_{j=1}^{t} n\mathbf{p}(n,u_i,d_j).\end{aligned} \tag{C.9}$$

To evaluate $\bar{n}(N)$ with a non-integer N, we have to find the analytical expression for $\bar{n}_I(N)$. Let us consider the four terms of $\bar{n}_I(N)$ separately. We use $\sum_{n=2}^{N-2} n\mathbf{p}(n,1,1)$ as an example.

$$\begin{aligned}\sum_{n=2}^{N-2} n\mathbf{p}(n,1,1) &= \sum_{n=2}^{N-2} n\sum_{m=1}^{R} C_m X_m^n \\ &= \sum_{m=1}^{R} C_m\left(\sum_{n=2}^{N-2} nX_m^n\right) \\ &= \sum_{m=1}^{R} C_m\left(2X_m^2+3X_m^3+\cdots+(N-2)X_m^{N-2}\right).\end{aligned} \tag{C.10}$$

Let us first consider those $m \in \mathcal{M}$. Define A as,

$$A = 2X_m^2+3X_m^3+\cdots+(N-2)X_m^{N-2}, \tag{C.11}$$

and therefore,

$$AX_m = 2X_m^3+3X_m^4+\cdots+(N-2)X_m^{N-1}. \tag{C.12}$$

Subtracting (C.12) from (C.11) gives

$$
\begin{aligned}
(1-X_m)A &= 2X_m^2 + X_m^3 + X_m^4 + \cdots + X_m^{N-2} - (N-2)X_m^{N-1} \\
&= 2X_m^2 + X_m^3 + X_m^4 + \cdots + X_m^{N-2} + X_m^{N-1} - (N-1)X_m^{N-1} \qquad \text{(C.13)} \\
&= 2X_m^2 - (N-1)X_m^{N-1} + \frac{X_m^3 - X_m^N}{1-X_m}.
\end{aligned}
$$

Thus,

$$
A = \frac{2X_m^2 - (N-1)X_m^{N-1} + \dfrac{X_m^3 - X_m^N}{1-X_m}}{1-X_m}. \qquad \text{(C.14)}
$$

On the other hand, for those $m \in \mathcal{M}^c$, we have

$$
2X_m^2 + 3X_m^3 + \cdots + (N-2)X_m^{N-2} = \frac{N(N-3)}{2}. \qquad \text{(C.15)}
$$

Therefore,

$$
\sum_{n=2}^{N-2} n\mathbf{p}(n,1,1) = \sum_{m\in\mathcal{M}} C_m \left[\frac{2X_m^2 - (N-1)X_m^{N-1} + \dfrac{X_m^3 - X_m^N}{1-X_m}}{1-X_m} \right] + \sum_{m\in\mathcal{M}^c} C_m \frac{N(N-3)}{2}. \qquad \text{(C.16)}
$$

We can apply similar analysis to the other three terms. They are

$$
\begin{aligned}
\sum_{n=2}^{N-2}\sum_{j=1}^{t} n\mathbf{p}(n,1,d_j) &= \sum_{m\in\mathcal{M}}\sum_{j=1}^{t} C_m D_{j,m} \left[\frac{2X_m^2 - (N-1)X_m^{N-1} + \dfrac{X_m^3 - X_m^N}{1-X_m}}{1-X_m} \right] \\
&\quad + \sum_{m\in\mathcal{M}^c}\sum_{j=1}^{t} C_m D_{j,m} \frac{N(N-3)}{2}, \qquad \text{(C.17)}
\end{aligned}
$$

$$\sum_{n=2}^{N-2}\sum_{i=1}^{s} n\mathbf{p}(n,u_i,1) = \sum_{m\in\mathcal{M}}\sum_{i=1}^{s} C_m U_{i,m}\left[\frac{2X_m^2-(N-1)X_m^{N-1}+\frac{X_m^3-X_m^N}{1-X_m}}{1-X_m}\right]$$

$$+\sum_{m\in\mathcal{M}^c}\sum_{i=1}^{s} C_m U_{i,m}\frac{N(N-3)}{2}, \tag{C.18}$$

and

$$\sum_{n=2}^{N-2}\sum_{i=1}^{s}\sum_{j=1}^{t} n\mathbf{p}(n,u_i,d_j) =$$

$$\sum_{m\in\mathcal{M}}\sum_{i=1}^{s}\sum_{j=1}^{t} C_m U_{i,m} D_{j,m}\left[\frac{2X_m^2-(N-1)X_m^{N-1}+\frac{X_m^3-X_m^N}{1-X_m}}{1-X_m}\right] \tag{C.19}$$

$$+\sum_{m\in\mathcal{M}^c}\sum_{i=1}^{s}\sum_{j=1}^{t} C_m U_{i,m} D_{j,m}\frac{N(N-3)}{2}.$$

Consequently, $\bar{n}(N)$ in the continuous variable version can be computed by (C.8), (C.16), (C.17), (C.18), and (C.19). Therefore, we have shown that both the production rate and the average buffer level can be evaluated with non-integer N. These analytical expressions for $P(N)$ and $\bar{n}(N)$ are used in the gradient method to solve the unconstrained problem for the deterministic multiple failure mode model of production lines. In the implementation of the computer program, the condition $X_m = 1$ is replaced by $|X_m - 1| \leq \delta$, where δ is a very small non-zero positive value; while the condition $X_m \neq 1$ is replaced by $|X_m - 1| > \delta$.

Appendix D

Supplementary Explicit Analytical Solutions to Levantesi et al. (1999a) for Continuous Multiple Failure Mode Two-machine Lines

The material covered in this section is an extension to the evaluation of continuous two-machine lines with multiple failure modes and finite buffer capacity developed by Levantesi et al. (1999a). In this appendix, we provide explicit, complete analytical solutions which are suitable for writing in computer algorithms. These are implied, but not stated explicitly by Levantesi et al. (1999a).

In Levantesi et al. (1999a), the system is modeled as a continuous time, mixed state Markov process. The state (x, α_u, α_d) represents the amount of material in the buffer (x) and the condition of the upstream machine M^u and the downstream machine M^d (α_u and α_d, respectively). Note that x is a real number in the continuous model. When M^u is operational $\alpha_u = 1$, while $\alpha_u = u_i, i = 1, \cdots, s$ means that M^u is down due to failure mode i. Similarly, α_d can assume the values $1, d_1, d_2, \cdots, d_t$. It is possible to distinguish the states the system can reach in internal states $(0 < x < N)$ and boundary states ($x = 0$ or N). The internal states are described by probability

density function $f(x, \alpha_u, \alpha_d)$ while the probability of finding the system in a boundary state is given by a probability mass function $\mathbf{p}(0, \alpha_u, \alpha_d)$ or $\mathbf{p}(N, \alpha_u, \alpha_d)$.

Levantesi et al. (1999a) provide in detail the steps to analyze, establish, and solve the model. In particular, they provide a general form of the probability density functions for all internal states. In addition, they also solved the steady-state probabilities of the boundary states in the case that $\mu_u > \mu_d$. However, they do not discuss the solutions for the cases where $\mu_u < \mu_d$ or $\mu_u = \mu_d$, although the case that $\mu_u < \mu_d$ can be solved easily by reversing the line in which $\mu_u > \mu_d$. In addition, in Levantesi et al. (1999a), both the production rate and the average inventory are given in integral forms, which cannot be used directly for programming. Therefore, we provide the analytical solution for the case $\mu_u = \mu_d$. Some discussion from the perspective of algorithm realization, including the analytical forms of the production rate and the average inventory, is also provided.

D.1 Note on Algorithm Realization when $\mu_u \neq \mu_d$

We discuss some issues that facilitate the realization of the algorithm developed by Levantesi et al. (1999a) when $\mu_u \neq \mu_d$. In particular, we assume that $\mu_u > \mu_d$ and the case $\mu_u < \mu_d$ can be realized by reversing the line. These issues include:

- The distribution of the roots of the polynomial in K;
- The method to determine the set of normalizing constants C_r;
- Modification of the steady-state probabilities of some non-transient boundary states;
- and the analytical expressions for the production rate and average inventory.

D.1.1 The Distribution of Roots of the Polynomial in K

The general form of the probability density functions for internal states (x, u_i, d_j) in Levantesi et al. (1999a) is

$$f(x, u_i, d_j) = \sum_{r=1}^{R} C_r e^{\lambda_r x} U_{i,r} D_{j,r}, \quad r = 1, \cdots, R, i = 1, \cdots, s, j = 1, \cdots, t \qquad \text{(D.1)}$$

where

$$\begin{aligned} U_{i,r} &= \frac{p^{u_i}}{r^{u_i} + K_r}, \quad i = 1, \cdots, s, \\ D_{j,r} &= \frac{p^{d_j}}{r^{d_i} - K_r}, \quad j = 1, \cdots, t, \\ \lambda_r &= \frac{-K_r}{\mu_d} \left[1 + \sum_{j=1}^{t} \frac{p^{d_j}}{r^{d_j} - K_r} \right], \end{aligned} \qquad \text{(D.2)}$$

and C_r are the normalizing constants to be determined. In the solution above, K_r is the rth root of the following polynomial in K of degree $R = s + t + 1$[1]:

$$\mu_u K \left[1 + \sum_{j=1}^{t} \frac{p^{d_j}}{r^{d_j} - K} \right] = \mu_d K \left[1 + \sum_{i=1}^{s} \frac{p^{u_i}}{r^{u_i} + K} \right]. \qquad \text{(D.3)}$$

Although not indicated explicitly in Levantesi et al. (1999a), it can be shown (according to an argument similar to the one provided in Tolio et al. 2002) that all the roots of the polynomial are real. Therefore, (D.3) can be solved by a binary search method. A clear understanding about how the roots of K distribute will enable us to solve it efficiently. According to a similar analysis presented in Tolio et al. (2002), we conclude the distribution of the roots of K as follows[2]:

- One root of K is $K = 0$;
- Re-arranging $r^{u_i}, i = 1, \cdots, s$ in an ascending order, for $i = 1, \cdots, s - 1$, there

[1] This is under the assumption that $\mu_u \neq \mu_d$. When $\mu_u = \mu_d$, we find a similar polynomial in K of degree $R = s + t$. We discuss this in Section D.2.1.

[2] We assume that all r^{u_i} are not the same and all r^{d_j} are not the same. Therefore, there are $s + t + 1$ different roots of K.

is a root of K between every two adjacent $-r^{u_i}$ and $-r^{u_{i+1}}$. So there are totally $s-1$ roots of K that are smaller than 0;

- Re-arranging $r^{d_j}, j = 1, \cdots, t$ in an ascending order, for $j = 1, \cdots, t-1$, there is a root of K between every two adjacent r^{d_j} and $r^{d_{j+1}}$. In addition, there is another root of K that is greater than r^{d_t}. Therefore, there are totally t roots of K that are greater than 0.

- Finally, there is a root of K between $-r^{u_1}$ and r^{d_1}.

Hence, there are $s+t+1$ roots of K in total. We provide an example to illustrate this graphically. Suppose we have a two machine line with parameters $\mu_u = 0.9$, $\mu_d = 0.8$, $r^{u_1} = 0.1$, $p^{u_1} = 0.01$, $r^{u_2} = 0.11$, $p^{u_2} = 0.01$, $r^{d_1} = 0.1$, $p^{d_1} = 0.009$, $r^{d_2} = 0.12$, and $p^{d_2} = 0.008$. The distribution of the roots of K in this example is illustrated in Figure D-1. In particular, the roots are $K_1 = -0.1053$, $K_2 = -0.0289$, $K_3 = 0$, $K_4 = 0.1104$, and $K_5 = 0.3468$. As a verification, K_1 is between $-r^{u_2}$ and $-r^{u_1}$, K_2 is between $-r^{u_1}$ and 0, K_3 is 0, K_4 is between r^{d_1} and r^{d_2}, and K_5 is greater than r^{d_2}.

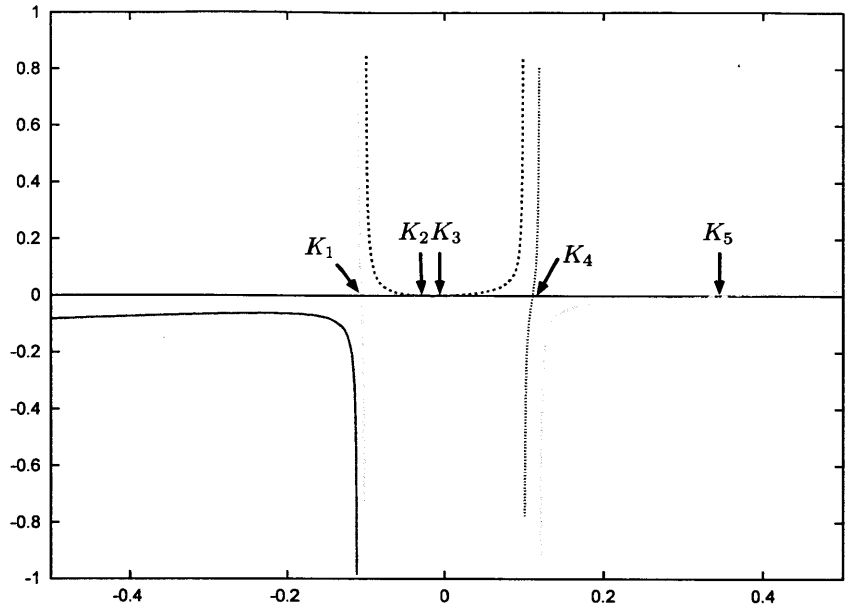
D.1.2 The Method to Determine the Normalizing Constants C_r

Levantesi et al. (1999a) provide $s+t+1$ equations for solving $C_r, r = 1, \cdots, s+t+1$ for the case that $\mu_u > \mu_d$. They are

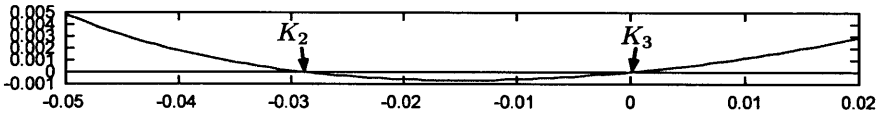
- $$\sum_{r=1}^{R} C_r D_{j,r} = 0, \quad j = 1, \cdots, t. \tag{D.4}$$

- $$\sum_{r=1}^{R} C_r e^{\lambda_r N} \left[\mu_u \left(1 + \sum_{j=1}^{t} D_{j,r} \right) - \mu_d \left(1 + \frac{P^U}{p^{u_i}} U_{i,r} \right) \right] = 0, \quad i = 1, \cdots, s \tag{D.5}$$

  where $P^U = \sum_{i=1}^{s} p^{u_i}$.



(a) Overview



(b) Partially enlarged view around $K = 0$

Figure D-1: Distribution of the solutions of K

- The normalization equation, i.e., the fact that the steady state probabilities of all states must add to 1.

However, it is formidable to write out the normalization equation. To avoid doing so, we adopt a trick in which we let C_{s+t+1} be 1 and solve the other $s+t$ equations for the $C_r, r = 1, \cdots, s+t$. Then we adjust $C_r, r = 1, \cdots, s+t+1$ such that the normalization equation is satisfied.

D.1.3 Modification of the Steady-State Probabilities of some Non-Transient Boundary States

Equations (37) and (38) of Levantesi et al. (1999a) are formulas to determine the steady-state probabilities of the two sets of boundary states $(N, 1, 1)$ and $(N, 1, d_j)$, respectively. They are

$$\mathbf{p}(N,1,1) = \frac{\mu_u}{p^{u_i}} f(N, u_i, 1) = \frac{\mu_u}{p^{u_i}} \sum_{r=1}^{R} C_r e^{\lambda_r N} U_{i,r}, \tag{D.6}$$

$$\mathbf{p}(N,1,d_j) = \frac{\mu_u}{r^{d_j}} \sum_{r=1}^{R} C_r e^{\lambda_r x} \left[\frac{p^{d_j}}{p^{u_i}} U_{i,r} + D_{j,r} \right], \quad j = 1, \cdots, t. \tag{D.7}$$

Although they are correct, the appearance of i in Equations (D.6) and (D.7) brings unnecessary confusion to the understanding of them. They indicate that $\mathbf{p}(N,1,1)$ and $\mathbf{p}(N,1,d_j)$ can be calculated by the parameters of any failure mode i of the upstream machine M^u. The results of $\mathbf{p}(N,1,1)$ or $\mathbf{p}(N,1,d_j)$ with different is are automatically identical[3]. However, we can avoid the confusion by a slight modification. Equation (D.6) comes from Equation (21) of Levantesi et al. (1999a), which is

$$\mu_d f(N, u_i, 1) = \frac{\mu_d}{\mu_u} p^{u_i} \mathbf{p}(N,1,1). \tag{D.8}$$

Summing over i yields

[3]We have verified this through numerical experiments.

$$
\begin{aligned}
\mu_d \sum_{i=1}^{s} f(N,u_i,1) &= \frac{\mu_d}{\mu_u} \sum_{i=1}^{s} p^{u_i} \mathbf{p}(N,1,1), \\
\text{or} \quad \sum_{i=1}^{s} f(N,u_i,1) &= \frac{P^U}{\mu_u} \mathbf{p}(N,1,1), \\
\text{or} \quad \mathbf{p}(N,1,1) &= \frac{\mu_u}{P^U} \sum_{i=1}^{s} f(N,u_i,1), \\
\text{or} \quad \mathbf{p}(N,1,1) &= \frac{\mu_u}{P^U} \sum_{i=1}^{s} \sum_{r=1}^{R} C_r e^{\lambda_r N} U_{i,r}.
\end{aligned}
\tag{D.9}
$$

By (17) of Levantesi et al. (1999a), we modify the formula of $\mathbf{p}(N,1,d_j)$ to

$$
\begin{aligned}
\mathbf{p}(N,1,d_j) &= \frac{1}{r^{d_j}} \left[\mathbf{p}^{d_j} p(N,1,1) + \mu_u f(N,1,d_j) \right] \\
&= \frac{1}{r^{d_j}} \left[p^{d_j} \frac{\mu_u}{P^U} \sum_{i=1}^{s} \sum_{r=1}^{R} C_r e^{\lambda_r N} U_{i,r} + \mu_u f(N,1,d_j) \right] \\
&= \frac{\mu_u}{r^{d_j}} \left[\frac{p^{d_j}}{P^U} \sum_{i=1}^{s} \sum_{r=1}^{R} C_r e^{\lambda_r N} U_{i,r} + \sum_{r=1}^{R} C_r e^{\lambda_r N} D_{j,r} \right] \\
&= \frac{\mu_u}{r^{d_j}} \sum_{r=1}^{R} C_r e^{\lambda_r x} \left[\frac{p^{d_j}}{P^U} \sum_{i=1}^{s} U_{i,r} + D_{j,r} \right], \quad j = 1, \cdots, t.
\end{aligned}
\tag{D.10}
$$

D.1.4 Integral Calculation

Levantesi et al. (1999a) indicate that the production rate of the continuous two-machine line is computed as

$$
P(N) = \mu_u E_u = \mu_u \left[\mathbf{p}(0,1,1) + \int_0^N \left(\sum_{j=1}^{t} f(x,1,d_j) + f(x,1,1) \right) dx \right] + \mu_d \mathbf{p}(N,1,1).
\tag{D.11}
$$

In the case where $\mu_u > \mu_d$, $\mathbf{p}(0,1,1) = 0$, and $\mathbf{p}(N,1,1)$ is given by Equation

(D.9). Hence, we discuss how to further compute the integral part above to find the analytical expression of $P(N)$.

$$\mu_u \int_0^N \left(\sum_{j=1}^{t} f(x,1,d_j) + f(x,1,1) \right) dx$$

$$= \mu_u \int_0^N \sum_{j=1}^{t} \sum_{r=1}^{R} C_r e^{\lambda_r x} D_{j,r} dx + \mu_u \int_0^N \sum_{r=1}^{R} C_r e^{\lambda_r x} dx$$

$$= \mu_u \sum_{j=1}^{t} \sum_{r=1}^{R} \int_0^N C_r e^{\lambda_r x} D_{j,r} dx + \mu_u \sum_{r=1}^{R} \int_0^N C_r e^{\lambda_r x} dx$$

$$= \mu_u \sum_{j=1}^{t} \sum_{r\in\mathcal{H}} \frac{C_r D_{j,r}}{\lambda_r} \left(e^{\lambda_r N} - 1\right) + \mu_u \sum_{r\in\mathcal{H}} \frac{C_r}{\lambda_r} \left(e^{\lambda_r N} - 1\right) + \mu_u \sum_{j=1}^{t} \sum_{r\notin\mathcal{H}} C_r D_{j,r} N + \mu_u \sum_{r\notin\mathcal{H}} C_r N \tag{D.12}$$

where $\mathcal{H}$ denotes the set of all nonzero λ_r. Therefore, the analytical expression of the production rate is

$$\begin{aligned} P(N) &= \mu_u \sum_{j=1}^{t} \sum_{r\in A} \frac{C_r D_{j,r}}{\lambda_r} \left(e^{\lambda_r N} - 1\right) + \mu_u \sum_{r\in A} \frac{C_r}{\lambda_r} \left(e^{\lambda_r N} - 1\right) \\ &+ \mu_u \sum_{j=1}^{t} \sum_{r\notin\mathcal{H}} C_r D_{j,r} N + \mu_u \sum_{r\notin\mathcal{H}} C_r N + \frac{\mu_u \mu_d}{P^U} \sum_{i=1}^{s} \sum_{r=1}^{R} C_r e^{\lambda_r N} U_{i,r}. \end{aligned} \tag{D.13}$$

Next, let us consider the average inventory for the case $\mu_u > \mu_d$. According to Levantesi et al. (1999a), the average inventory, $\bar{x}$, can be calculated by

$$\begin{aligned} \bar{x} &= \int_0^N x \left[f(x,1,1) + \sum_{i=1}^{s} f(x,u_i,1) + \sum_{j=1}^{t} f(x,1,d_j) + \sum_{i=1}^{s} \sum_{j=1}^{t} f(x,u_i,d_j) \right] dx \\ &+ N \left[\mathbf{p}(N,1,1) + \sum_{i=1}^{s} \mathbf{p}(N,u_i,1) + \sum_{j=1}^{t} \mathbf{p}(N,1,d_j) + \sum_{i=1}^{s} \sum_{j=1}^{t} \mathbf{p}(N,u_i,d_j) \right]. \end{aligned} \tag{D.14}$$

In steady state, $\mathbf{p}(N,u_i,1) = \mathbf{p}(N,u_i,d_j) = 0$, while $\mathbf{p}(N,1,1)$ and $\mathbf{p}(N,1,d_j)$

are given by Equations (D.9) and (D.10). Again, we focus on the integral part in Equation (D.14). We consider each component in the integral part separately.

- $$\begin{aligned} & \int_0^N x f(x,1,1)dx \\ = & \int_0^N x \sum_{r=1}^{R} C_r e^{\lambda_r x} dx \\ = & \sum_{r=1}^{R} C_r \int_0^N x e^{\lambda_r x} dx \\ = & \sum_{r\in\mathcal{H}} \frac{C_r}{\lambda_r}\left(Ne^{\lambda_r N} - \frac{1}{\lambda_r}\left(e^{\lambda_r N}-1\right)\right) + \sum_{r\notin\mathcal{H}} \frac{1}{2} C_r N^2. \end{aligned} \tag{D.15}$$

- $$\begin{aligned} & \int_0^N x \sum_{i=1}^{s} f(x,u_i,1)dx \\ = & \int_0^N x \sum_{i=1}^{s}\sum_{r=1}^{R} C_r e^{\lambda_r x} U_{i,r} dx \\ = & \sum_{i=1}^{s}\sum_{r=1}^{R} C_r U_{i,r} \int_0^N x e^{\lambda_r x} dx \\ = & \sum_{i=1}^{s}\sum_{r\in\mathcal{H}} \frac{C_r U_{i,r}}{\lambda_r}\left(Ne^{\lambda_r N} - \frac{1}{\lambda_r}\left(e^{\lambda_r N}-1\right)\right) + \sum_{i=1}^{s}\sum_{r\notin\mathcal{H}} \frac{1}{2} C_r U_{i,r} N^2. \end{aligned} \tag{D.16}$$

•

$$\int_0^N x \sum_{j=1}^{t} f(x,1,d_j)dx$$

$$= \int_0^N x \sum_{j=1}^{t} \sum_{r=1}^{R} C_r e^{\lambda_r x} D_{j,r} dx$$

$$= \sum_{j=1}^{t} \sum_{r=1}^{R} C_r D_{j,r} \int_0^N x e^{\lambda_r x} dx \tag{D.17}$$

$$= \sum_{j=1}^{t} \sum_{r\in\mathcal{H}} \frac{C_r D_{j,r}}{\lambda_r} \left(N e^{\lambda_r N} - \frac{1}{\lambda_r} \left(e^{\lambda_r N} - 1 \right) \right) + \sum_{j=1}^{t} \sum_{r\notin\mathcal{H}} \frac{1}{2} C_r D_{j,r} N^2.$$

•

$$\int_0^N x \sum_{i=1}^{s} \sum_{j=1}^{t} f(x,u_i,d_j)dx$$

$$= \int_0^N x \sum_{i=1}^{s} \sum_{j=1}^{t} \sum_{r=1}^{R} C_r e^{\lambda_r x} U_{i,r} D_{j,r} dx$$

$$= \sum_{i=1}^{s} \sum_{j=1}^{t} \sum_{r=1}^{R} C_r U_{i,r} D_{j,r} \int_0^N x e^{\lambda_r x} dx$$

$$= \sum_{i=1}^{s} \sum_{j=1}^{t} \sum_{r\in\mathcal{H}} \frac{C_r U_{i,r} D_{j,r}}{\lambda_r} \left(N e^{\lambda_r N} - \frac{1}{\lambda_r} \left(e^{\lambda_r N} - 1 \right) \right) + \sum_{i=1}^{s} \sum_{j=1}^{t} \sum_{r\notin\mathcal{H}} \frac{1}{2} C_r U_{i,r} D_{j,r} N^2. \tag{D.18}$$

Substituting Equations (D.9), (D.10), and (D.15) to (D.18) into Equation (D.14) gives a analytical expression of the average inventory.

D.2 Algorithm Realization when $\mu_u = \mu_d$

Note that $\mu_u = \mu_d$ requires necessary modification of Equation (D.3) and the distribution of the roots of K. It also requires the modification to the steady-state probabilities of the boundary states since, for example, $\mathbf{p}(0,1,1)$ is no longer zero as it is in the case $\mu_u \neq \mu_d$. In the reminder of this section, let $\mu = \mu_u = \mu_d$.

D.2.1 Distribution of the Roots of the Polynomial in K when $\mu_u = \mu_d$

Since $\mu_u = \mu_d$, Equation (D.3) is simplified to

$$K\sum_{j=1}^{t}\frac{p^{d_j}}{r^{d_j}-K} = K\sum_{i=1}^{s}\frac{p^{u_i}}{r^{u_i}+K}. \tag{D.19}$$

Note that this is a polynomial in K of degree $R = s+t$, rather than $R = s+t+1$ as in the case $\mu_u \neq \mu_d$. This is because if we multiple both sides of (D.19) by

$$\prod_{i=1}^{s}(r^{u_i}+K)\prod_{j=1}^{t}(r^{d_j}-K),$$

each term on the left hand side will become

$$Kp^{d_j}(r^{d_1}-K)\cdots(r^{d_{j-1}}-K)(r^{d_{j+1}}-K)\cdots(r^{d_t}-K)\prod_{i=1}^{s}(r^{u_i}+K),\quad j=1,\cdots,t,$$

while each term on the right hand side will become

$$Kp^{u_i}(r^{u_1}+K)\cdots(r^{u_{i-1}}+K)(r^{u_{i+1}}+K)\cdots(r^{u_s}+K)\prod_{j=1}^{t}(r^{d_j}-K),\quad i=1,\cdots,s.$$

It is easy to see that the highest order of K in all terms are K^{s+t} and therefore (D.19) is a polynomial in K of degree $s+t$. On the other hand, if we do the same thing to both sides of (D.3), the terms $\mu_u K$ and $\mu_d K$ on the right and left hand sides

of (D.3) will become

$$\mu_u K \prod_{i=1}^{s}(r^{u_i}+K)\prod_{j=1}^{t}(r^{d_j}-K),$$

and

$$\mu_d K \prod_{i=1}^{s}(r^{u_i}+K)\prod_{j=1}^{t}(r^{d_j}-K).$$

Since both of them contain K^{s+t+1}, which is the highest order of K that (D.3) has after the modification, (D.3) is a polynomial in K of degree $s+t+1$.

Assuming that all r^{u_i} are distinct and all r^{d_j} are distinct, the distribution of the roots of K from (D.19) is concluded as follows:

- One root of K is $K=0$;
- Re-arranging $r^{u_i}, i=1,\cdots,s$ in an ascending order, for $i=1,\cdots,s-1$, there is a root of K between every two adjacent $-r^{u_i}$ and $-r^{u_{i+1}}$. So there are $s-1$ roots of K due to the distinct values of $r^{u_i}, i=1,\cdots,s$;
- Re-arranging $r^{d_j}, j=1,\cdots,t$ in an ascending order, for $j=1,\cdots,t-1$, there is a root of K between every two adjacent r^{d_j} and $r^{d_{j+1}}$. Therefore, there are $t-1$ roots of K due to the different values of $r^{d_j}, j=1,\cdots,t$;
- There is another root of K between $-r^{u_1}$ and r^{d_1}. In particular,
  - If $\sum_{j=1}^{t}\frac{p^{d_j}}{r^{d_j}} > \sum_{i=1}^{s}\frac{p^{u_i}}{r^{u_i}}$, then the root is between $-r^{u_1}$ and 0;
  - If $\sum_{j=1}^{t}\frac{p^{d_j}}{r^{d_j}} < \sum_{i=1}^{s}\frac{p^{u_i}}{r^{u_i}}$, then the root is between 0 and r^{d_1};
  - If $\sum_{j=1}^{t}\frac{p^{d_j}}{r^{d_j}} = \sum_{i=1}^{s}\frac{p^{u_i}}{r^{u_i}}$, then the root is 0. Therefore, 0 is a root with a multiplicity of 2.

Hence, there are totally $s+t$ roots of K. Again, we provide an example to illustrate this graphically. Suppose we have a two machine line with parameters $\mu_u=\mu_d=1$, $r^{u_1}=0.1$, $p^{u_1}=0.01$, $r^{u_2}=0.11$, $p^{u_2}=0.01$, $r^{d_1}=0.1$, $p^{d_1}=0.009$, $r^{d_2}=0.12$, and $p^{d_2}=0.008$. Note that in this example, we have $\sum_{j=1}^{2}\frac{p^{d_j}}{r^{d_j}} < \sum_{i=1}^{2}\frac{p^{u_i}}{r^{u_i}}$. Therefore, we

expect to have a root of K between 0 and r^{d_1}, which is 0.1. The distribution of the roots of K in this example is illustrated in Figure D-2. In particular, the roots are $K_1 = -0.1051$, $K_2 = 0$, $K_3 = 0.0105$, $K_4 = 0.1111$. As a verification, K_1 is between $-r^{u_2}$ and $-r^{u_1}$, K_2 is 0, K_3 is between 0 and r^{d_1}, and K_4 is between r^{d_1} and r^{d_2}.

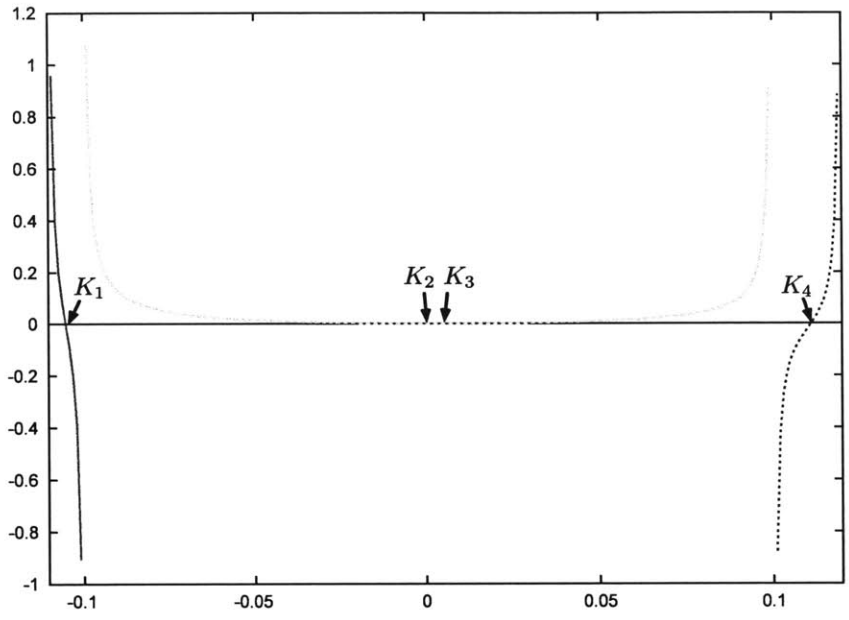


Figure D-2: Distribution of the solutions of K, $\mu_u = \mu_d$

D.2.2 Steady-state Probabilities of Boundary States

In the case $\mu_u > \mu_d$, the steady-state probabilities of those non-transient boundary states are deducted based on the fact that $\mathbf{p}(0,1,1) = 0$. However, this is no longer true when $\mu_u = \mu_d$. Thus, we need to re-construct the formulas of the steady-state probabilities of those non-transient boundary states.

When $\mu_u = \mu_d$, Equation (21) of Levantesi et al. (1999a) becomes

$$\mu f(N, u_i, 1) = p^{u_i}\mathbf{p}(N, 1, 1), \quad i = 1, \cdots, s, \tag{D.20}$$

and therefore,

$$\mathbf{p}(N,1,1) = \frac{\mu}{p^{u_i}} f(N,u_i,1) = \frac{\mu}{p^{u_i}} \sum_{r=1}^{R} C_r e^{\lambda_r N} U_{i,r}, \tag{D.21}$$

or, summing over i in Equation (D.20),

$$\begin{aligned} \mu \sum_{i=1}^{s} f(N,u_i,1) &= \sum_{i=1}^{s} p^{u_i} \mathbf{p}(N,1,1), \\ \text{or} \quad \sum_{i=1}^{s} f(N,u_i,1) &= \frac{P^U}{\mu} \mathbf{p}(N,1,1), \\ \text{or} \quad \mathbf{p}(N,1,1) &= \frac{\mu}{P^U} \sum_{i=1}^{s} f(N,u_i,1), \\ \text{or} \quad \mathbf{p}(N,1,1) &= \frac{\mu}{P^U} \sum_{i=1}^{s} \sum_{r=1}^{R} C_r e^{\lambda_r N} U_{i,r}. \end{aligned} \tag{D.22}$$

Substituting Equation (D.21) into Equation (17) of Levantesi et al. (1999a) gives

$$\mathbf{p}(N,1,d_j) = \frac{\mu}{r^{d_j}} \sum_{r=1}^{R} C_r e^{\lambda_r N} \left[\frac{p^{d_j}}{p^{u_i}} U_{i,r} + D_{j,r} \right], \quad j = 1, \cdots, t. \tag{D.23}$$

Note that (D.23) is the same as Equation (D.7) except that μ_u in (D.7) is replaced by μ in (D.23). Substituting Equation (D.22) into Equation (17) of Levantesi et al. (1999a) gives

$$\mathbf{p}(N,1,d_j) = \frac{\mu}{r^{d_j}} \sum_{r=1}^{R} C_r e^{\lambda_r N} \left[\frac{p^{d_j}}{P^U} \sum_{i=1}^{s} U_{i,r} + D_{j,r} \right], j = 1, \cdots, t \tag{D.24}$$

Next, we show how to find $\mathbf{p}(0,1,1)$. By Equation (11) of Levantesi et al. (1999a) we know that

$$\left(P^U + P^D\right) \mathbf{p}(0,1,1) = \sum_{i=1}^{s} r^{u_i} \mathbf{p}(0,u_i,1) \tag{D.25}$$

where $P^U = \sum_{i=1}^{s} p^{u_i}$ and $P^D = \sum_{j=1}^{t} p^{d_j}$. In addition, from Equation (10) of Levantesi et al. (1999a) we know that

$$r^{u_i}\mathbf{p}(0,u_i,1) = p^{u_i}\mathbf{p}(0,1,1) + \mu f(0,u_i,1), i = 1,\cdots,s. \tag{D.26}$$

Summing over i gives

$$\begin{aligned}
\sum_{i=1}^{s} r^{u_i}\mathbf{p}(0,u_i,1) &= \sum_{i=1}^{s} p^{u_i}\mathbf{p}(0,1,1) + \sum_{i=1}^{s} \mu f(0,u_i,1), \\
\text{or } \left(P^U + P^D\right)\mathbf{p}(0,1,1) &= P^U\mathbf{p}(0,1,1) + \sum_{i=1}^{s} \mu f(0,u_i,1), \\
\text{or } \left(P^U + P^D\right)\mathbf{p}(0,1,1) &= P^U\mathbf{p}(0,1,1) + \mu \sum_{i=1}^{s}\sum_{r=1}^{R} C_r U_{i,r}, \qquad \text{(D.27)} \\
\text{or } P^D\mathbf{p}(0,1,1) &= \mu \sum_{i=1}^{s}\sum_{r=1}^{R} C_r U_{i,r}, \\
\text{or } \mathbf{p}(0,1,1) &= \frac{\mu}{P^D}\sum_{i=1}^{s}\sum_{r=1}^{R} C_r U_{i,r}.
\end{aligned}$$

Therefore, substituting Equation (D.27) into Equation (D.26), we have

$$\mathbf{p}(0,u_i,1) = \frac{1}{r^{u_i}}\left(\frac{\mu p^{u_i}}{P^D}\sum_{i=1}^{s}\sum_{r=1}^{R} C_r U_{i,r} + \mu\sum_{r=1}^{R} C_r U_{i,r}\right), \quad i = 1,\cdots,s. \tag{D.28}$$

Equations (D.22), (D.24), (D.27) and (D.28) are the steady-state probabilities of all non-transient boundary states. Now, we are ready to find out the normalizing constants C_r according to the method introduced in Section D.1.2. Note that Section D.1.2 offers $s + t + 1$ equations while we only have $s + t$ unknowns of C_r when $\mu_u = \mu_d$. Therefore, we adopt the t equations in the equation set (D.4) and the first $s - 1$ equations in the equation set (D.5), together with the normalization equation.

Appendix E

Proof of the Assertion in Section 6.3 for the Case where Some $N_i^\star = N_{\min}$

We provide the proof of the assertion in Section 6.3 for the case where some $N_i^\star = N_{\min}$. Recall that the assertion states that the constrained problem

$$\begin{aligned}
\max_{\mathbf{N},I} \quad & J(N_1,\cdots,N_k,I) = A'P(N_1,\cdots,N_k,I) - \sum_{i=1}^{k} b_i N_i - cI \\
\text{subject to} \quad & P(N_1,\cdots,N_k,I) \geq \hat{P}, \\
& N_i \geq N_{\min}, \forall i = 1,\cdots,k, \\
& \sum_{i=1}^{k} N_i \geq I, \\
& I \geq 0
\end{aligned}$$

has the same solution for all A' in which the solution of the unconstrained problem (6.5) has $P(N_1^u,\cdots,N_k^u,I^u) < \hat{P}$.

Let B be $\{i|N_i^\star = N_{\min}\}$. Hence, $N_i^\star = N_{\min}, \forall i \in B$, while $N_i^\star > N_{\min}, \forall i \in B^c$, where B^c is $\{i|N_i^\star > N_{\min}\}$. In this case, those $N_i^\star$ equal to $N_{\min}$ are on the boundary of the feasible region of the optimal solution. By condition (6.12), we know that $\mu_i^\star = 0, \forall i \in B^c$. Hence, we simplify the KKT conditions (6.9) to (6.14) to

$$-\begin{pmatrix} \frac{\partial J(\mathbf{N}^\star, I^\star)}{\partial N_1} \\ \vdots \\ \frac{\partial J(\mathbf{N}^\star, I^\star)}{\partial N_k} \\ \frac{\partial J(\mathbf{N}^\star, I^\star)}{\partial I} \end{pmatrix} - \mu_0^\star \begin{pmatrix} \frac{\partial P(\mathbf{N}^\star, I^\star)}{\partial N_1} \\ \vdots \\ \frac{\partial P(\mathbf{N}^\star, I^\star)}{\partial N_k} \\ \frac{\partial P(\mathbf{N}^\star, I^\star)}{\partial I} \end{pmatrix} - \sum_{i \in B} \mu_i^\star e_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \tag{E.1}$$

where

$$e_i = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

in which the 1 is in the ith position, and

$$\mu_0^\star \left(\hat{P} - P(\mathbf{N}^\star, I^\star) \right) = 0, \tag{E.2}$$

$$\mu_i^\star (N_{\min} - N_i^\star) = 0, \forall i = 1, \cdots, k, \tag{E.3}$$

$$\mu_{k+1}^\star \left(I^\star - \sum_{i=1}^{k} N_i^\star \right) = 0, \tag{E.4}$$

$$\mu_{k+2}^\star I^\star = 0, \tag{E.5}$$

where $\mu_i^\star \geq 0, \forall i = 0, \cdots, k+2$. In particular, $\mu_i = 0, \forall i \in B^c$ and $\mu_{k+1} = \mu_{k+2} = 0$ from (E.4) and (E.5) since $0 < I^\star < \sum_{i=1}^{k} N_i^\star$.

Next, we argue that not all $N_i^\star, i \in B^c$ satisfy $\partial J(\mathbf{N}^\star)/\partial N_i = 0$. This is because if $\partial J(\mathbf{N}^\star)/\partial N_i = 0$ for all $N_i^\star, i \in B^c$ and all other $N_i^\star = N_{\min}, i \in B$, then the solution of the constrained problem $(N_1^\star, \cdots, N_k^\star, I^\star)$ is equivalent to the solution of the unconstrained problem, since it is just constrained by the buffer size constraint. This is a conflict. Therefore, it is not possible for all $N_i^\star, i \in B^c$ to satisfy $\partial J(\mathbf{N}^\star)/\partial N_i = 0$. So, there exists some $\hat{i}'s$ for which $N_{\hat{i}}^\star > N_{\min}$ and $\partial J(\mathbf{N}^\star)/\partial N_{\hat{i}} \neq 0$. Since $N_{\hat{i}}^\star > N_{\min}$, by condition (E.3), we know $\mu_{\hat{i}}^\star = 0$. Thus, $\mu_0^\star \neq 0$ since otherwise condition (E.1) would be violated. Hence, by condition (E.2), we know that the optimal solution $(\mathbf{N}^\star, I^\star)$ satisfies $P(\mathbf{N}^\star, I^\star) = \hat{P}$. In this case, there are more than one active inequality constraints: the production rate constraint $g_0(\mathbf{N}, I)$ and buffer size constraints $g_i(\mathbf{N}, I), \forall i \in B$. It is not hard to see that $\nabla g_0(\mathbf{N}^\star, I^\star)$ and $\nabla g_i(\mathbf{N}^\star, I^\star), \forall i \in B$ are linearly independent[1] so the optimal solution is regular.

As before, replacing $\mu_0^\star$ by $\mu_0 > 0$ and $\mu_i^\star, \forall i \in B$ by $\mu_i > 0, \forall i \in B$ in constraint (E.1) gives

$$-\nabla J(\bar{\mathbf{N}}, I) + \mu_0 \nabla \left(\hat{P} - P(\bar{\mathbf{N}}, I)\right) + \sum_{i \in B} \mu_i \nabla \left(N_{\min} - \bar{N}_i\right) = 0, \qquad \text{(E.6)}$$

where $\bar{\mathbf{N}}$ is the unique solution. Note that $\bar{\mathbf{N}}$ is exactly the optimal solution of the following optimization problem

[1]This is because in $\nabla g_0(\mathbf{N}^\star, I^\star) = \nabla(\hat{P} - P(\mathbf{N}^\star, I^\star))$, the first k components are negative due to the monotonicity of $P(\mathbf{N}, I)$ with respect to $\mathbf{N}$ while the last component can be either positive or negative depending on whether I is greater than half of the total buffer size of the loop or not, but a linear combination of $\nabla g_i(\mathbf{N}^\star, I^\star) = -e_i, \forall i \in B$ cannot generate all corresponding non-zero components of $\nabla g_0(\mathbf{N}^\star)$ since not all buffers belong to set B.

$$
\begin{aligned}
\min_{\mathbf{N},I} \quad -\bar{J}(\mathbf{N},I) &= -J(\mathbf{N},I) + \mu_0\left(\hat{P} - P(\mathbf{N},I)\right) + \sum_{i\in B}\mu_i(N_{\min} - N_i) \\
\text{subject to} \quad N_{\min} - N_i &\leq 0, \forall i = 1,\cdots,k, \\
N_i &= N_{\min}, \forall i \in B, \\
I - \sum_{i=1}^{k} N_i &\leq 0, \\
-I &\leq 0,
\end{aligned}
\tag{E.7}
$$

which is equivalent to

$$
\begin{aligned}
\max_{\mathbf{N},I} \quad \bar{J}(\mathbf{N},I) &= J(\mathbf{N},I) - \mu_0\left(\hat{P} - P(\mathbf{N},I)\right) \\
\text{subject to} \quad N_{\min} - N_i &\leq 0, \forall i = 1,\cdots,k, \\
N_i &= N_{\min}, \forall i \in B, \\
I - \sum_{i=1}^{k} N_i &\leq 0, \\
-I &\leq 0,
\end{aligned}
\tag{E.8}
$$

or

$$\begin{aligned} \max_{\mathbf{N},I} \quad & \bar{J}(\mathbf{N},I) = (A+\mu_0)P(\mathbf{N},I) - \sum_{i=1}^{k} b_i N_i - cI \\ \text{subject to} \quad & N_i \geq N_{\min}, \forall i = 1, \cdots, k, \\ & N_i = N_{\min}, \forall i \in B, \\ & \sum_{i=1}^{k} N_i \geq I, \\ & I \geq 0, \end{aligned} \tag{E.9}$$

or, finally,

$$\begin{aligned} \max_{\mathbf{N},I} \quad & \bar{J}(\mathbf{N},I) = A'P(\mathbf{N},I) - \sum_{i=1}^{k} b_i N_i - cI \\ \text{subject to} \quad & N_i \geq N_{\min}, \forall i = 1, \cdots, k, \\ & N_i = N_{\min}, \forall i \in B, \\ & \sum_{i=1}^{k} N_i \geq I, \\ & I \geq 0, \end{aligned} \tag{E.10}$$

where $A' = A + \mu_0$. Again, this is exactly the unconstrained problem in which A is replaced by A', and $(\bar{\mathbf{N}}, \bar{I})$ is its optimal solution with the fact that $\bar{N}_i = N_{\min}, \forall i \in B$.

Appendix F

$\hat{P}$ Surface Search

We provide a brief description about the $\hat{P}$ surface search. All buffer size allocations, $\mathbf{N}$, such that $P(\mathbf{N}) = \hat{P}$ compose the $\hat{P}$ surface. For instance, consider a 4-machine 3-buffer line with parameters $r_1 = .11$, $p_1 = .008$, $r_2 = .12$, $p_2 = .01$, $r_3 = .1$, $p_3 = .01$, $r_4 = .09$, and $p_4 = .01$. The target production rate is $\hat{P} = .88$. A portion of the $P^\star$ surface of this example is shown in Figure F-1. It means that all points on this surface satisfy $P(\mathbf{N}) = \hat{P}$. We further let the profit be $J(\mathbf{N}) = 2000P(\mathbf{N}) - \sum_{i=1}^{3} N_i - \sum_{i=1}^{3} \bar{n}_i$. The optimal solution achieved by the algorithm developed in Chapter 4 is also shown in Figure F-1.

To verify the optimal solution, we conduct a search on this $\hat{P}$ surface. It is obvious that the $\hat{P}$ surface is boundless and it is impractical to search the whole surface, even for the simplest three-machine two-buffer line. Therefore, in our implementation, we only search around the optimal solution achieved by our algorithm. Since there is only one maximum, searching around the optimal solution is accurate enough.

In the $\hat{P}$ surface search, we first find all feasible points on the surface around the optimal solution gained by the algorithm. To do this, we set ranges for all buffers and then at each time we vary one buffer size by a step size 1 while keeping other buffers unchanged. Therefore, we can calculate the production rates and the profits for all combinations of all buffers. All those points, whose production rate P satisfy $P(\mathbf{N}) - \hat{P} \leq \delta$ where δ is a very small non-zero positive value, compose the $\hat{P}$ surface. Then we compare the profits of those points. The point having the maximal profit

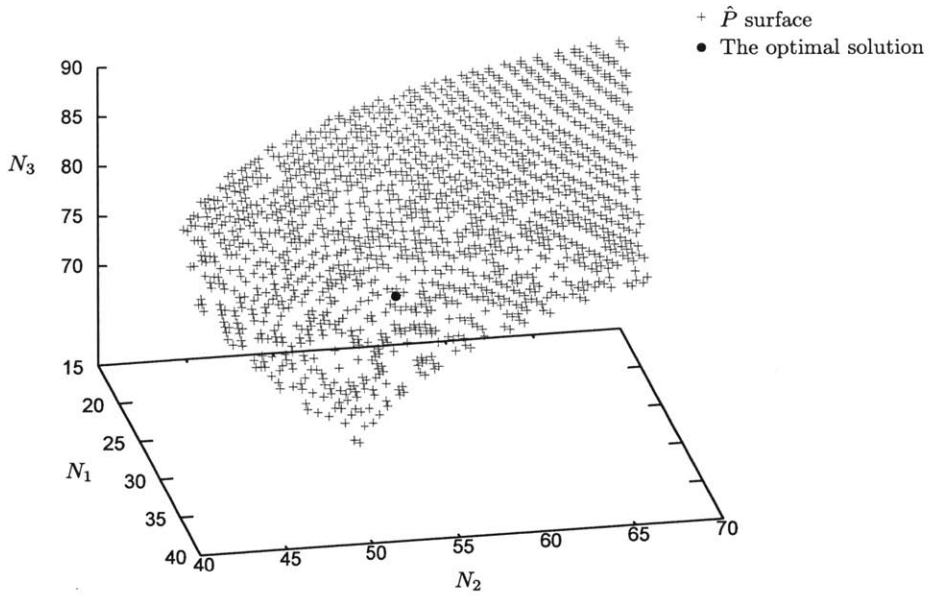


Figure F-1: $\hat{P}$ surface

in the $\hat{P}$ search becomes the optimal solution by this searching method. Then, we compare our optimal solution with this one and calculate the error.

Appendix G

Details of the 5000 Three-Machine Lines Studied in Chapter 3

We mention in Section 3.6.6 that, in order to investigate the single profit maximum issue of production lines, we have generated 5000 three-machine two-buffer lines randomly and studied their profits. For each type of three-machine line, we consider 1000 lines.

These 5000 three-machine two-buffer lines are generated according to the method described in Gershwin (2011). In particular, the isolated production rate of each machine M_i (i.e., $r_i/(r_i + p_i)$) is between .667 and .952, with r_i and p_i randomly generated. In addition, in order to compute the profit, we randomly generated the revenue coefficient A between 1500 and 7500, the buffer space coefficients b_1 and b_2 between 0 and 5, and the inventory cost coefficients c_1 and c_2 between 0 and 5.

For each line, we search for local maxima in (N_1, N_2) space by varying both N_1 and N_2 from 4 to 500. (Note that 4 is the minimum value of N_i given the convention of the deterministic model we use.) To check if a point $J(N_1, N_2)$ is a local maximum, we compute the profits associated with the buffer sizes adjacent to (N_1, N_2). Specifically, if (N_1, N_2) is not on the boundary or the corner of the search space, then we calculate the profits $J(N_1-1, N_2)$, $J(N_1+1, N_2)$, $J(N_1, N_2-1)$, and $J(N_1, N_2+1)$. If and only if $J(N_1, N_2)$ is greater than all these four values mentioned above, it is considered as a local maximum. If (N_1, N_2) is on the boundary or the corner of the search space, we

compute the profits of the points that are both adjacent to (N_1, N_2) and within the search space. For instance, for $J(80, 500)$, we compare it with $J(80, 499)$, $J(79, 500)$, and $J(81, 500)$, but not $J(80, 501)$ because $(8, 501)$ is outside the search space. As explained at the end of Section 3.6.6, the profit J is decreasing in N_1 and N_2 for N_1 and N_2 sufficiently large and it finally goes to $-\infty$ if N_1 and N_2 keep increasing. Therefore, there are no local maxima for large (N_1, N_2).

In all these 5000 three-machine lines studied, there is only a single global profit maximum for each line. This indicates that the single profit maximum is a reasonable assumption for production lines and therefore we can adopt a gradient method to solve the profit maximization problem without a production rate constraint.

Bibliography

Akyildiz, I. (1988). On the exact and approximate throughput analysis of closed queuing networks with blocking. *IEEE Transactions on Software Engineering 14*(1), 62 –70.

Albright, S. C., W. L. Winston, and C. J. Zappe (2009). *Data Analysis & Decision Making with Microsoft®Excel, Revised Third Edition.* Mason, OH: South-Western Cengage Learning.

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

Altiok, T. and S. Stidham (1983). The allocation of interstage buffer capacities in production lines. *IIE Transactions 15*(4), 292–299.

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

Anantharam, V. and P. Tsoucas (1990). Stochastic concavity of throughput in series of queues with finite buffers. *Advances in Applied Probability 22*(3), 761–763.

Anthony, B. W. (2011). Private communication.

Balsamo, S., V. de Nitto Personé, and R. Onvural (2001). *Analysis of Queuing Networks with Blocking.* Kluwer Academic Publishers.

Bard, J. F. and T. A. Feo (1989). Operations sequencing in discrete parts manufacturing. *Management Science 35*(2), 249 – 255.

Bautista, J. and J. Pereira (2007). Ant algorithms for a time and space constrained assembly line balancing problem. *European Journal of Operational Research 177*, 2016–2032.

Berresford, G. C. and A. M. Rockett (2008). *Applied Calculus, fifth edition.* Belmont, CA: Brooks/Cole.

Bertsekas, D. P. (1999). *Nonlinear Programming: 2nd Edition.* Athena Scientific.

Bertsekas, D. P. and J. N. Tsitsiklis (2008). *Introduction to Probability, 2nd Edition.* Athena Scientific.

Bierbooms, R., I. J. Adan, and M. van Vuuren (2011). Performance analysis of fluid flow production lines with finite buffers and generally distributed up- and downtimes. In *VIII Conference on Stochastic Models of Manufacturing and Service Operations*, Kuşadası, Turkey, pp. 269 - 276.

Biller, S., S. Marin, S. Meerkov, and L. Zhang (2009). Closed bernoulli production lines: Analysis, continuous improvement, and leanness. *Automation Science and Engineering, IEEE Transactions on 6*(1), 168–180.

Bonvik, A. M., C. E. Couch, and S. B. Gershwin (1997). A comparison of production-line control mechanisms. *International Journal of Production Research 35*(3), 789–804.

Bonvik, A. M., Y. Dallery, and S. B. Gershwin (2000). Approximate analysis of production systems operated by a conwip/finite buffer hybrid control policy. *International Journal of Production Research 38*(13), 2845 – 2869.

Borgh, D. (2009a). Optimal buffer capacity allocation in continuous lines. Working paper, Politecnico di Milano, Dipartimento di Meccanica.

Borgh, D. (2009b). Revised experiments. Unpublished manuscript.

Borgh, D. (2010). *An Analytical Method for the Optimal Design of Asynchronous Transfer Lines.* Ph. D. thesis, Politecnico di Milano PhD course on Manufacturing Technologies and Systems XXI cycle.

Bozer, Y. A. and Y.-J. Hsieh (2005). Throughput performance analysis and machine layout for discrete-space closed-loop conveyors. *IIE Transactions 37*, 77–89.

Burman, M. H. (1995, June). *New Results in Flow Line Analysis.* Ph. D. thesis, Massachusetts Institute of Technology, Operations Research Center.

Burman, M. H., S. B. Gershwin, and C. Suyematsu (1998). Hewlett-packard uses operations research to improve the design of a printer production line. *Interfaces 28*(1), 24–36.

Buxey, G. M., N. D. Slack, and R. Wild (1973). Production flow line system design - a review. *AIIE Transactions 5*(1), 37–48.

Buzacott, J. A. (1967a). Automatic transfer lines with buffer stocks. *International Journal of Production Research 5*(3), 183–200.

Buzacott, J. A. (1967b). *Markov chain analysis of automatic transfer lines with buffer stock.* Ph. D. thesis, University of Birmingham.

Buzacott, J. A. (1968). Prediction of the efficiency of production systems without internal storage. *International Journal of Production Research 6*(3), 173–188.

Buzacott, J. A. (1971). The role of inventory banks in flow-line production systems. *International Journal of Production Research 9*(4), 425–436.

Buzacott, J. A. (1972). The effect of station breakdowns and random processing times on the capacity of flow lines with in-process storage. *AIIE Transactions 4*(4), 308–312.

Buzacott, J. A. and L. E. Hanifin (1978). Models of automatic transfer lines with inventory banks a review and comparison. *AIIE Transactions 10*(2), 197–207.

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

Carano, M. and J. Fjelstad (2003). Chapter 7: Printed circuit board fabrication. In C. Harper (Ed.), *Electronic Materials and Processes Handbook* (3 ed.). New York: McGraw-Hill Professional.

Chan, F. T. S. and E. Y. H. Ng (2002). Comparative evaluations of buffer allocation strategies in a serial production line. *The International Journal of Advanced Manufacturing Technology 19*(11), 789–800.

Choong, Y. F. and S. B. Gershwin (1987). A decomposition method for the approximate evaluation of capacitated transfer lines with unreliable machines and random processing times. *IIE Transactions 19*(2), 150–159.

Colledani, M., A. Matta, and T. Tolio (2005). Performance evaluation of production lines with finite buffer capacity producing two different products. *OR Spectrum 27*, 243–263.

Colledani, M., M. Matta, T. Grasso, and T. Tolio (2003). A new analytical method for optimal buffer space allocation in production lines. In *37 CIRP International Seminar on Manufacturing Systems*, Budapest, pp. 231–237.

Colledani, M. and T. Tolio (2005). An analytical method for optimal buffer capacity allocation in production systems. In *AITEM Conference.*

Cox, D. R. (1955). A use of complex probabilities in the theory of stochastic processes. *Mathematical Proceedings of the Cambridge Philosophical Society 51*, 313–319.

Dallery, Y. (1999). Extending the scope of analytical methods for performance evaluation of manufacturing flow systems. In *Second Aegean International Conference on "Analysis and Modeling of Manufacturing Systems*, Tinos Island, Greece.

Dallery, Y., R. David, and X.-L. Xie (1988). An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers. *IIE Transactions 20*(3), 280–283.

Dallery, Y., R. David, and X.-L. Xie (1989). Approximate analysis of transfer lines with unreliable machines and finite buffers. *IEEE Transactions on automatic control 34*(9), 943–953.

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

Dallery, Y. and H. Le Bihan (1995). An improved decomposition method for the analysis of production lines with unreliable machines and finite buffers. In *Proceedings of the INRIA/IEEE Symposium on Emerging Technologies and Factory Automation (ETFA)*, Volume 3, Paris, pp. 567 – 578.

Dallery, Y. and H. Le Bihan (1999). An improved decomposition method for the analysis of production lines with unreliable machines and finite buffers. *International Journal of Production Research 37*(5), 1093–1117.

Dallery, Y., Z. Liu, and D. Towsley (1994). Equivalence, reversibility, symmetry and concavity properties in fork-join queuing networks with blocking. *Journal of the ACM (JACM) 41*(5), 903–942.

Dallery, Y. and D. Towsley (1991). Symmetry property of the throughput in closed tandem queueing networks with finite capacity. *Operations Research Letters 10*(9), 541–547.

De Koster, M. B. M. (1987). Estimation of line efficiency by aggregation. *International Journal of Production Research 25*(4), 615–626.

De Koster, M. B. M. (1988). An improved algorithm to approximate the behaviour of flow lines. *International Journal of Production Research 26*(4), 691–700.

Di Mascolo, M., R. David, and Y. Dallery (1991). Modeling and analysis of assembly systems with unreliable machines and finite buffers. *IIE Transactions 23*(4), 315–331.

Diamantidis, A. C. and C. T. Papadopoulos (2004). A dynamic programming algorithm for the buffer allocation problem in homogeneous asymptotically reliable serial production lines. *Mathematical Problems in Engineering 2004*(3), 209–223.

Dolgui, A., A. Eremeev, A. Kolokolov, and V. Sigaev (2002). A genetic algorithm for the allocation of buffer storage capacities in a production line with unreliable machines. *Journal of Mathematical Modelling and Algorithms 1*(2), 89–104.

Frein, Y., C. Commault, and Y. Dallery (1996). Modeling and analysis of closed-loop production lines with unreliable machines and finite buffers. *IIE Transactions 28*, 545–554.

Gebennini, E., A. Grass, C. Fantuzzi, S. B. Gershwin, and I. C. Schick (2009). On the introduction of a restart policy in the two-machine, one-buffer transfer line model. In *VII Conference on Stochastic Models of Manufacturing and Service Operations*, Ostuni, Italy, pp. 81–88.

Gebennini, E., A. Grassi, and C. Fantuzzi (2011). A building block for the decomposition analysis of production lines with restart policy. In *VIII Conference on Stochastic Models of Manufacturing and Service Operations*, Kuşadası, Turkey, pp. 101 – 108.

Gershwin, S. B. (1984). An efficient decomposition method for the approximate evaluation of production lines with finite storage space. In A. Bensoussan and J. Lions (Eds.), *Analysis and Optimization of Systems*, Volume 63 of *Lecture Notes in Control and Information Sciences*, pp. 645–658. Springer Berlin / Heidelberg.

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

Gershwin, S. B. (1987b). Representation and analysis of transfer lines with machines that have different processing rates. *Annals of Operations Research 9*, 511–530.

Gershwin, S. B. (1989). An efficient decomposition algorithm for unreliable tandem queuing systems with finite buffers. In H. G. Perros and T. Altiok (Eds.), *Queueing networks with blocking*, pp. 127–146. Amsterdam: North-Holland.

Gershwin, S. B. (1991). Assembly/disassembly systems: An efficient decomposition algorithm for tree-structured networks. *IIE Transactions 23*(4), 302–314.

Gershwin, S. B. (1994). *Manufacturing Systems Engineering.* Prentice-Hall. Currently available at `http://home.comcast.net/~hierarchy/MSE/mse.html`. For corrections, see `http://web.mit.edu/manuf-sys/www/gershwin.errata.html`.

Gershwin, S. B. (2003). Factory models for manufacturing systems engineering. Technical Report IMST005, Massachusetts Institute of Technology.

Gershwin, S. B. (2005). Design and operation of manufacturing systems. unpublished manuscript.

Gershwin, S. B. (2011). Case generation for algorithm testing. In *VIII Conference on Stochastic Models of Manufacturing and Service Operations*, Kuşadası, Turkey, pp. 193 – 200.

Gershwin, S. B. and O. Berman (1981). Analysis of transfer lines consisting of two unreliable machines with random processing times and finite storage buffers. *AIIE Transactions 13*(1), 2–11.

Gershwin, S. B. and M. H. Burman (2000). A decomposition method for analyzing inhomogeneous assembly/disassembly systems. *Annals of Operations Research 93*(1–4), 91–115.

Gershwin, S. B. and S. Fallah-Fini (2007). A general model and analysis of a discrete two-machine production line. In *Sixth Conference on the Analysis of Manufacturing Systems*, Lunteren, Netherlands, pp. 39–44.

Gershwin, S. B. and Y. Goldis (1995). Efficient algorithms for transfer line design. Technical Report LMP-95-005, Laboratory for Manufacturing and Productivity, Massachusetts Institute of Technology.

Gershwin, S. B. and I. C. Schick (1980). Continuous model of an unreliable two-machine material flow system with a finite interstage buffer. Report LIDS-R-1039, MIT Laboratory for Information and Decision Systems.

Gershwin, S. B. and I. C. Schick (1983). Modeling and analysis of three stage transfer lines with unreliable machines and finite buffers. *Operations Research 31*(2), 354–377.

Gershwin, S. B. and J. E. Schor (2000). Efficient algorithms for buffer space allocation. *Annals of Operations Research 93*, 117–144.

Gershwin, S. B. and L. M. Werner (2007). An approximate analytical method for evaluating the performance of closed-loop flow systems with unreliable machines and finite buffers. *International Journal of Production Research 45*(14), 3085–3111.

Glasserman, P. and D. D. Yao (1996). Structured buffer-allocation problems. *Discrete Event Dynamic Systems 6*(1), 9–41.

Glassey, C. R. and Y. Hong (1986). The analysis of behavior of an unreliable two-stage automatic transfer line with inter-stage buffer storage. Technical report, Department of Industrial Engineering and Operations Research, University of California, Berkeley.

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

Glover, F. and M. Laguna (1997). *Tabu Search.* Norwell, MA: Kluwer.

Graves, S. C. (2011). Private communication.

Gün, L. and A. M. Makowski (1989). An approximation method for general tandem queueing systems subject to blocking. In H. G. Perros and T. Altiok (Eds.), *Queueing Networks with Blocking*, pp. 147–174. New York: North-Holland.

Helber, S., K. Schimmelpfeng, and R. Stolletz (2009). Setting inventory levels of conwip flow lines via linear programming. Diskussionspapiere der Wirtschaftswissenschaftlichen Fakultt der Universitt Hannover dp-436, Universitt Hannover, Wirtschaftswissenschaftliche Fakultt.

Hillier, F. S. and K. C. So (1995). On the optimal design of tandem queueing systems with finite buffers. *Queueing Systems 21*, 245–266.

Hopp, W. J. and M. L. Spearman (2000). *Factory Physics: foundations of manufacturing management.* Irwin/McGraw-Hill.

Huang, M.-G., P.-L. Chang, and Y.-C. Chou (2002, February). Buffer allocation in flow-shop-type production systems with general arrival and service patterns. *Computer & Operations Research 29*(2), 103–121.

Ip, W., M. Huang, K. Yung, D. Wang, and X. Wang (2007). CONWIP based control of a lamp assembly production line. *Journal of Intelligent Manufacturing 18*(2), 261–271.

Isaacson, E. and H. B. Keller (1994). *Analysis of Numerical Methods.* New York: Dover Publication.

Jackson, J. R. (1963). Jobshop-like queueing systems. *Management Science 10*, 131–142.

Jeong, K.-C. and Y.-D. Kim (2000). Heuristics for selecting machines and determining buffer capacities in assembly systems. *Computers & Industrial Engineering 38*(3), 341–360.

Jin, Z., Z. Yang, and T. Ito (2006). Metaheuristic algorithms for the multistage hybrid flowshop scheduling problem. *International Journal of Production Economics 100*, 322–334.

Kim, D., D. Kulkarni, and F. Lin (2002). An upper bound for carriers in a three-workstation closed serial production system operating under production blocking. *IEEE Transactions on Automatic Control 47*(7), 1134 –1138.

Kim, J. and S. B. Gershwin (2005). Integrated quality and quantity modeling of a production line. *OR Spectrum 27*(2–3), 287–314. Reprinted in *Stochastic Modeling of Manufacturing Systems — Advances in Design, Performance Evaluation, and Control Issues*, edited by George Liberopoulos, Chrissoleon T. Papadopoulos, Barış Tan, J. MacGregor Smith, and Stanley B. Gershwin, Springer, 2006.

Kim, J.-H. and T.-E. Lee (2008). Schedulability analysis of time-constrained cluster tools with bounded time variation by an extended petri net. *IEEE Transactions on Automation Science and Engineering 5*(3), 490–503.

Kim, J.-H., T.-E. Lee, H.-Y. Lee, and D.-B. Park (2003). Scheduling analysis of time-constrained dual-armed cluster tools. *IEEE Transactions on Semiconductor Manufacturing 16*(3), 521–534.

Kitamura, S., K. Mori, and A. Ono (2006, October 18-21). Capacity planning method for semiconductor fab with time constraints between operations. In *Proceedings of the 2006 SICE-ICASE International Joint Conference*, Busan, Korea, pp. 1100–1103.

Koenigsberg, E. (1959). Production lines and internal storage - a review. *Management Science 5*(4), 410–433.

Kwon, S.-T. (2006). On the optimal buffer allocation of an fms with finite in-process buffers. In M. Gavrilova, O. Gervasi, V. Kumar, C. Tan, D. Taniar, A. Lagan, Y. Mun, and H. Choo (Eds.), *Computational Science and Its Applications - ICCSA 2006*, Volume 3982 of *Lecture Notes in Computer Science*, pp. 767–776. Springer Berlin / Heidelberg.

Larson, R., R. P. Hostetler, and B. H. Edwards (2005). *Calculus (With Analytic Geometry), eight edition.* Boston, MA: Houghton Mifflin Company.

Lau, H.-S. (1986a). A directly-coupled two-stage unpaced line. *IIE Transactions 18*(3), 304–312.

Lau, H.-S. (1986b). The production rate of a two-stage system with stochastic processing times. *International Journal of Production Research 24*(2), 401–412.

Le Bihan, H. and Y. Dallery (2000). A robust decomposition method for the analysis of production lines with unreliable machines and finite buffers. *Annals of Operations Research 93*, 265–297.

Lee, T.-E. and S.-H. Park (2005). An extended event graph with negative places and tokens for time window constraints. *IEEE Transactions on Automation Science and Engineering 2*(4), 319–332.

Levantesi, R., A. Matta, and T. Tolio (1999a). Continuous two-machine line with multiple failure modes and finite buffer capacity. In *IV CONVEGNO AITEM*, Brescia, Italia.

Levantesi, R., A. Matta, and T. Tolio (1999b). A decomposition method for performance evaluation of production lines with random processing times, multiple failure modes and finite buffer capacity. In *Proceedings of the Second Aegean International Conference on Analysis and Modeling of Manufacturing Systems*, Tinos Island, Greece.

Levantesi, R., A. Matta, and T. Tolio (1999c). Exponential two-machine lines with multiple failure modes and finite buffer capacity. In *Second International Aegean Workshop on the Analysis and Modeling of manufacturing system*, Tinos Island, Greece.

Levantesi, R., A. Matta, and T. Tolio (2001). A new algorithm for buffer allocation in production lines. In *The Third Aegean International Conference on Design and Analysis of Manufacturing Systems*, Tinos Island, Greece.

Levantesi, R., A. Matta, and T. Tolio (2003). Performance evaluation of continuous production lines with machines having different processing times and multiple failure modes. *Performance Evaluation 51*, 247–268.

Li, J., D. E. Blumenfeld, and J. M. Alden (2006). Comparisons of two-machine line models in throughput analysis. *International Journal of Production Research 44*, 1375–1398.

Li, J., D. E. Blumenfeld, N. Huang, and J. M. Alden (2009). Throughput analysis of production systems: recent advances and future topics. *International Journal of Production Research 47*(14), 3823–3851.

Li, N., S. Yao, G. Liu, and C. Zhuang (2010). Optimization of a multi-constant work-in-process semiconductor assembly and test factory based on performance evaluation. *Computers & Industrial Engineering 59*(2), 314–322.

Lim, J. T. and S. M. Meerkov (1993). On asymptotically reliable closed serial production lines. *Control Engineering Practice 1*(1), 147 – 152.

Lim, J.-T., S. M. Meerkov, and F. Top (1990). Homogeneous, asymptotically reliable serial production line: theory and a case study. *IEEE Transaction on Automatic Control 35*(5), 524–534.

Little, J. D. C. (1961). A proof of the queueing formula $L = \lambda W$. *Operations Research 9*, 383–387.

Lu, S. C.-H., D. Ramaswamy, and P. R. Kumar (1994). Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants. *IEEE Transactions Semiconductor Manufacturing 7*(3), 374–388.

Maggio, N. (2000). An analytical method for evaluating the performance of closed loop production lines with unreliable machines and finite buffer. Master's thesis, Politecnico di Milano.

Maggio, N., A. Matta, S. B. Gershwin, and T. Tolio (2009). A decomposition approximation for three-machine closed-loop production systems with unreliable machines, finite buffers and a fixed population. *IIE Transactions 41*(6), 562–574.

Marshall, A. W. and M. Shaked (1986). Multivariate new better than used distributions. *Mathematics of Operations Research 11*(1), 110–116.

Meester, L. E. and J. G. Shanthikumar (1990). Concavity of the throughput of tandem queueing systems with finite buffer storage space. *Advances in Applied Probability 22*(3), 764–767.

Mhada, F. and R. Malhamé (2011). Approximate performance analysis of conwip disciplines in unreliable non homogeneous transfer lines. *Annals of Operations Research 182*, 213–233.

Miller, G., J. Pawloski, and C. R. Standridge (2010). A case study of lean, sustainable manufacturing. *Journal of Industrial Engineering and Management 3*(1), 11 – 32.

Monden, Y. (1998). *Toyota production system: an integrated approach to just-in-time.* Engineering & Management Press.

Muth, E. J. (1979). The reversibility property of production lines. *Management Science 25*(2), 152–158.

Nahas, N., M. Nourelfath, and D. Ait-Kadi (2009). Selecting machines and buffers in unreliable series-parallel production lines. *International Journal of Production Research 47*(14), 3741 – 3774.

Neacy, E., S. Brown, and R. McKiddie (1994). Measurement and improvement of manufacturing capacity (mimac) survey and interview results. Technical report, SEMATECH Technology Transfer 94052374A-XFR.

Neuts, M. F. (1981). *Matrix-geometric Solutions in Stochastic Models - An Algorithmic Approach.* Baltimore, MD: The Johns Hopkins University Press.

Okamura, K. and H. Yamashina (1977). Analysis of the effect of buffer storage capacity in transfer line systems. *AIIE Transactions 9*(2), 127 – 135.

Onvural, R. O. (1990). Closed queueing networks with blocking. In H. Takagi (Ed.), *Stochastic Analysis of Computer and Communication Systems*, pp. 499–528. New York: North-Holland.

Onvural, R. O. and H. G. Perros (1986). On equivalencies of blocking mechanisms in queueing networks with blocking. *Operations Research Letters 5*(6), 293 – 297.

Onvural, R. O. and H. G. Perros (1987). Throughput analysis in closed queueing networks with finite buffers. Technical Report CCSP-TR-87/2, Center for Communications and Signal Processing, North Carolina State University.

Onvural, R. O. and H. G. Perros (1989). Approximate throughput analysis of cyclic queueing networks with finite buffers. *IEEE Transactions on Software Engineering 15*(6), 800 – 808.

Ovalle, O. R. and A. C. Marquez (2003). Exploring the utilization of a conwip system for supply chain management. a comparison with fully integrated supply chains. *International Journal of Production Economics 83*(2), 195–215.

Papadopoulos, H. T. and C. Heavey (1996). Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines. *European Journal of Operational Research 92*(1), 1 – 27.

Papadopoulos, H. T., C. Heavey, and J. Browne (1993). *Queueing Theory in Manufacturing Systems Analysis and Design*. London: Chapman & Hall.

Park, T. (1993). A two-phase heuristic algorithm for determining buffer sizes of production lines. *International Journal of Production Research 31*(3), 613–631.

Perros, H. G. (1990). Approximation algorithm for open queueing networks with blocking. In H. Takagi (Ed.), *Stochastic Analysis of Computer and Communication Systems*, pp. 451–498. New York: North-Holland.

Rajan, R. and R. Agrawal (1998). Concavity of queueing systems with nbu service times. *Advances in Applied Probability 30*(2), 551–567.

Rao, N. P. (1975). On the mean production rate of a two–stage production system of the tandem type. *International Journal of Production Research 13*(2), 207–217.

Resano Lázaro, A. and C. J. Luis Pérez (2008). Analysis of an automobile assembly line as a network of closed loops working in both, stationary and transitory regimes. *International Journal of Production Research 46*(17), 4803–4825.

Resano Lázaro, A. and C. J. Luis Pérez (2009). Dynamic analysis of an automobile assembly line considering starving and blocking. *Robotics and Computer-Integrated Manufacturing 25*(2), 271 – 279.

Robinson, J. K. and R. Giglio (1999). Capacity planning for semiconductor wafer fabrication with time constraints between operations. In *Proceedings of the 1999 Winter Simulation Conference*, Phoenix, Arizona, USA, pp. 880–887.

Rodzewicz, T. C., J. H. Potterton, K. M. Lappe, B. C. Yezefski, and D. J. Singer (2010). A new approach to ship repair using conwip and parts kits. *Journal of Ship Production 26*, 155–162.

Rostami, S., B. Hamidzadeh, and D. Camporese (2001). An optimal periodic scheduler for dual-arm robots in cluster tools with residency constraints. *IEEE Transactions on Robotics and Automation 17*(5), 609–618.

Schick, I. C. and S. B. Gershwin (1978). Modelling and analysis of unreliable transfer lines with finite interstage buffers (Volume VI of Complex materials handling and assembly systems). Report ESL-FR-834-6, MIT Electronic Systems Laboratory.

Schor, J. E. (1995). Efficient algorithms for buffer allocation. Master's thesis, Massachusetts Institute of Technology. Also available as MIT Laboratory for Manufacturing and Productivity report LMP-95-006.

Senanayake, C. D., V. Subramaniam, and Y. Cao (2011). Performance evaluation of hybrid manufacturing systems for continuous improvement: An analytical method. In *VIII Conference on Stochastic Models of Manufacturing and Service Operations*, Kuşadası, Turkey, pp. 185 – 192.

Seong, D., S. Y. Chang, and Y. Hong (1994). An algorithm for buffer allocation with linear resource constraints in a continuous flow production line. Technical report, Department of Industrial Engineering, POSTECH, Korea. Technical Report IE-TR-94-05.

Seong, D., S. Y. Chang, and Y. Hong (1995). Heuristic algorithms for buffer allocation in a production line with unreliable machines. *International Journal of Production Research 33*(7), 1989–2005.

Shanthikumar, J. G. and D. D. Yao (1988). Second-order properties of the throughput of a closed queueing network. *Mathematics of Operations Research 13*(3), 524–534.

Shanthikumar, J. G. and D. D. Yao (1989a). Monotonicity and concavity properties in cyclic queueing networks with finite buffers. In H. G. Perrios and T. Altiok (Eds.), *Queuing Networks with Blocking*, pp. 325–344. New York: North-Holland.

Shanthikumar, J. G. and D. D. Yao (1989b). Optimal buffer allocation in a multicell system. *International Journal of Flexible Manufacturing Systems 1*(4), 347–356.

Shi, C. and S. B. Gershwin (2009a). An efficient buffer design algorithm for production line profit maximization. *International Journal of Production Economics 122*(2), 725–740.

Shi, C. and S. B. Gershwin (2009b). An efficient buffer design algorithm for production line profit maximization. In *VII Conference on Stochastic Models of Manufacturing and Service Operations*, Ostuni, Italy, pp. 217–224.

Shi, C. and S. B. Gershwin (2011a). Part waiting time distribution in a two-machine line. In *VIII Conference on Stochastic Models of Manufacturing and Service Operations*, Kuşadası, Turkey, pp. 261 – 268.

Shi, C. and S. B. Gershwin (2011b). The segmentation method for long line optimization. In *VIII Conference on Stochastic Models of Manufacturing and Service Operations*, Kuşadası, Turkey, pp. 128 – 135.

Shi, L. and S. Men (2003). Optimal buffer allocation in production lines. *IIE Transactions 35*(1), 1–10.

Smith, J. M. and F. R. B. Cruz (2005). The buffer allocation problem for general finite buffer queueing networks. *IIE Transactions 37*(4), 343–365.

Smunt, T. L. and W. C. Perkins (1985). Stochastic unpaced line design: Review and further experimental results. *Journal of Operations Management 5*(3), 351 – 373.

So, K. C. (1997). Optimal buffer allocation strategy for minimizing work-in-process inventory in unpaced production lines. *IIE Transactions 29*(1), 81–88.

Spearman, M. L., D. L. Woodruff, and W. J. Hopp (1990). Conwip: a pull alternative to kanban. *International Journal of Production Research 28*(5), 879–894.

Sung, C. S. and S. T. Kwon (1994). Performance modeling of an fms with finite input and output buffers. *International Journal of Production Economics 37*(2-3), 161 – 175.

Syrowicz, D. (1999, June). Decomposition analysis of a deterministic, multiple-part-type, multiple-failure-mode production line. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science.

Takahashi, K., Myreshka, and D. Hirotani (2005). Comparing conwip, synchronized conwip, and kanban in complex supply chains. *International Journal of Production Economics 93-94*, 25–40.

Tan, B. (2002). State-space modeling and analysis of pull controlled production systems. In S. B. Gershwin, Y. Dallery, C. Papadopoulos, and J. Smith (Eds.), *Analysis and Modeling of Manufacturing Systems*, Chapter 15. Kluwer International Series in Operations Research and Management Science.

Tan, B. and S. B. Gershwin (2009). Analysis of a general markovian two-stage continuous-flow production system with a finite buffer. *International Journal of Production Economics 120*(2), 327 – 339.

Tan, B. and S. B. Gershwin (2011). Modelling and analysis of markovian continuous flow systems with a finite buffer. *Annals of Operations Research 182*, 5–30.

Tempelmeier, H. (2003). Practical considerations in the optimization of flow production systems. *International Journal of Production Research 41*(1), 149–170.

Terracol, C. and R. David (1987). An aggregation method for performance evaluation of transfer lines with unreliable machines and finite buffers. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1333–1338.

Tolio, T. (2011). Performance evaluation of two-machines line with multiple up and down states and finite buffer capacity. In *VIII Conference on Stochastic Models of Manufacturing and Service Operations*, Kuşadası, Turkey, pp. 117 – 127.

Tolio, T., M. Colledani, and D. Borgh (2009). An analytical method for the optimal design of buffers in asynchronous transfer lines. In *VII Conference on Stochastic Models of Manufacturing and Service Operations*, Ostuni, Italy, pp. 225–232.

Tolio, T. and S. B. Gershwin (1996). Deterministic two-machine lines with multiple failure modes. Technical report, Politecnico di Milano — Dipartimento di Meccanica.

Tolio, T. and S. B. Gershwin (1998). Throughput estimation in cyclic queueing networks with blocking. *Annals of Operations Research 79*, 207–229.

Tolio, T. and A. Matta (1998). A method for performance evaluation of automated flow lines. *CIRP Annals - Manufacturing Technology 47*(1), 373–376.

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

Uzsoy, R., C.-Y. Lee, and L. A. Martin-Vega (1992). A review of production planning and scheduling models in the semiconductor industry part i: System characteristics, performance evaluation and production planning. *IIE Transactions 24*(4), 47–60.

van Vuuren, M. and I. J. Adan (2009). Performance analysis of tandem queues with small buffers. *IIE Transactions 41*(10), 882–892.

Wang, C.-H. and D. A. Bourne (1997). Design and manufacturing of sheet-metal parts: Using features to aid process planning and resolve manufacturability problems. *Robotics and Computer-Integrated Manufacturing 13*(3), 281 – 294.

Werner, L. (2001). Analysis and design of closed loop manufacturing systems. Master's thesis, Massachusetts Institute of Technology Operations Research Center.

Wijngaard, J. (1979). The effect of interstage buffer storage on the output of two unreliable production units in series, with different production rates. *AIIE Transactions 11*(1), 42–47.

Xie, X. (2002). Evaluation and optimization of two-stage continuous transfer lines subject to time-dependent failures. *Discrete Event Dynamic Systems 12*, 109–122.

Yang, D.-L. and M.-S. Chern (1995). A two-machine flowshop sequencing problem with limited waiting time constraints. *Computers Industry Engineering 28*(1), 63–70.

Zhang, J. Z. (2006). *Analysis and Design of Manufacturing Systems with Multiple-Loop Structures.* Ph. D. thesis, Massachusetts Institute of Technology Department of Mechanical Engineering.