Modeling Service Performance and Dynamic Worker Allocation Policies for Order Fulfillment Systems

by

Hyun Ho Kim

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 18, 2009

Keywords: Multi-server queues, approximation models, phase-type distributions, steady state sojourn time distributions, state-dependent sojourn time distributions, order fulfillment systems, probability of success, dynamic worker allocation

Approved by:

Kevin R. Gue, Chair, Associate Professor of Industrial & Systems Engineering
Ming Liao, Professor of Mathematics & Statistics
Emmett J. Lodree, Jr., Assistant Professor of Statistics & Management Science, University of Alabama

Abstract

In this dissertation, we propose several dynamic worker allocation policies based on the sojourn time distribution in due-date order fulfillment systems. To establish a sojourn-based policy, we must be able to compute the sojourn time distribution for an arriving order and each order in the system. We require the distribution because we must make a probabilistic statement about the completion time of an order. We need two types of sojourn time distribution for the system. The first is a steady-state distribution to estimate the probability of success for an arriving order. The second is the state-dependent sojourn time distribution, which estimates the probability of success for an order in system.

First, we develop an approximation model for the sojourn time distribution of customers or orders arriving to an acyclic multi-server queueing network. The model accepts general interarrival times and general service times, and is based on the characteristics of phase-type distributions. Distributions produced by the model agree well with those produced by simulation for a variety of serial and general queueing networks.

Second, we develop an approximation model for the state-dependent sojourn time distribution of customers or orders in a multi-server queueing system, when interarrival and service times can take on general distributions. The model is based on the characteristics of phase-type distributions, and uses a Markov process to represent the all-busy period for a waiting time distribution. We also discuss how the model might be used to execute dynamic delivery promises in an order fulfillment system.

Third, we propose several dynamic worker allocation policies to maximize service performance of an order fulfillment system. Our policies make use of the state-dependent sojourn time distribution for an order, which we compute with a model based on phase-type distributions. Our approach differs from other work on dynamic worker allocation in that we

focus on service performance as perceived by the customer, instead of traditional system performance measures such as cycle time, throughput, and WIP. Our results suggest that order fulfillment systems can significantly improve their service performance by moving the right number of workers to the right place, at the right time.

Acknowledgments

I would like to acknowledge the Republic of Korea Army for giving me the opportunity to study and for the support during the course of my research. When I look back on my past, I have received many benefits from the Army. I have had chances to earn an education from Korea Military Academy to a Ph.D. and it has created the essence of my character as a man and soldier. From now on, I think it is time to pay back my debts more than I received from ROKA.

My first thanks must go to my advisor, Dr. Kevin R. Gue. He was always willing to provide advice on academic work and help me along my way. I have to attribute most of my success to Dr. Gue's guidance. Through most of my years here, I also have learned about life from him. He thoughtfully considers others, tries as best as he can to live a moderate life, and has a serious attitude on life. I am so lucky to have him as my mentor.

I am deeply grateful to Dr. Ming Liao and Dr. Emmett J. Lodree, Jr for their help and support, especially I was able to figure out a lot of problems for my dissertation with Dr. Liao's classes and his help. I shall never forget attending a benevolent professor's classes. I also give thanks to an excellent batch of fellow graduate students, who I will not mention by name for fear of leaving someone out. They made the social, academic and all other aspects of my years at Auburn University the most enjoyable of my life.

Here I am in mid-life without quite realizing it. I am absolutely delighted that I did something to delight my parents as a son. I could not finish my work without their devoted love. This dissertation is dedicated to my wife Hyesook. I owe much of my success to my wife and I thank my wife, Hyesook, for all of her love and support during the last four years. In an unfamiliar foreign land, my family (Hyesook and Siheon) was a great comfort and help for me. Life in Auburn will stay with our family as a happy memory.

I aspire to guide my son as well as my parents guided me!

Table of Contents

List of Illustrations

List of Tables

Chapter 1

Introduction

1.1 The order fulfillment system

An order fulfillment system receives orders from points of sale in a continuous stream, processes them and delivers the finished goods to the customer at a specific time or times during the day. Manufacturing and production systems and distribution centers are the most typical order fulfillment systems.

Consider a distribution center in which orders move through three stages — picking, packing and shipping. Order picking is the task of retrieving items from their storage locations, and typically involves a worker traveling through the warehouse or an automated system retrieving the item and delivering it to the next stage via a conveyor or automated guided vehicle. Packing is the task of preparing items for delivery, and typically involves consolidating items in an order, packaging them for delivery and labeling the packages. Shipping is the task of staging orders and loading them onto trucks, and typically involves sorting orders (often on pallets) and building loads on outbound trucks. Orders arrive in this system throughout the day and are prepared for shipping through the picking and packing processes by workers. Figure 1.1 illustrates a nominal order fulfillment system, where circles represent workers.

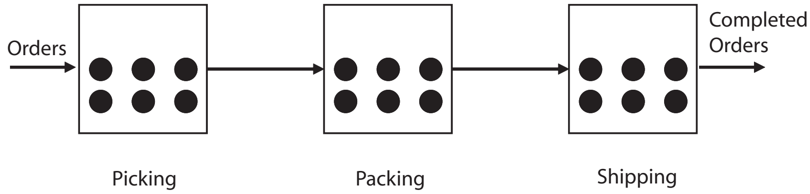


Figure 1.1: A nominal order fulfillment system

In this system, we can view each activity (picking, packing and shipping) as a workstation and workers in each workstation as servers. Thus we can model the order fulfillment system as an open queueing network to analyze system performance.

Customers expect a fair price, fast on-time delivery, and high quality in their purchase of products. Among those customer expectations, we focus on the delivery lead time because it is the most important function of the order fulfillment system. Customers usually want to receive their orders as soon as possible. Thus, how companies respond to customer orders is a matter of primary concern for this system.

In general, when we model a manufacturing or distribution center, we often use cycle time, throughput and WIP as system performance measures. That is, the objective of most research is to minimize cycle time, maximize throughput, or perhaps to minimize WIP. However, these metrics do not directly reflect customer satisfaction and two of them (throughput and WIP) are of no concern to the customer at all. Throughout our dissertation we use a performance metric called *Next Scheduled Departure* (NSD), which was introduced by Doerr and Gue (2006) to measure customer service in a distribution center. NSD records the percentage of orders arriving in the 24 hours between cutoff times on successive days that make it onto the truck in the current day. The cutoff time is established by managers in the distribution center, and any order arriving before this time is due in the current day. Doerr and Gue (2006) claim that an increase in the NSD means a direct increase in customer satisfaction. We might say that NSD is a system performance measure from the customer's perspective, and existing metrics such as WIP, cycle time and throughput are from the manufacturer's perspective.

1.2 Problem Statement

Our research considers ways in which workers can be allocated dynamically in the few hours before the truck departure to "flush" the system of orders that are almost completed, thereby increasing the service performance of the system. Our discussions with distribution center managers suggest that they are increasingly considering cross trained workers to achieve operational flexibility, as is often done in manufacturing. To implement worker allocation correctly, we need to answer the following questions: how many workers should we move and when?

Simulation experiments on simple policies have led us to consider more complex, but intuitively appealing, policies based on *sojourn time.* The idea is to consider the likelihood (probability) that an order in the system will make it on the next departing truck if the allocation of workers does not change. When this probability drops below a threshhold value, then some workers in picking should be moved so as to improve the probability of success for other, more promising orders.

To establish a sojourn-based policy, we must be able to compute the sojourn time *distribution* for an arriving order and for each order in the system. Note that this is different from, and more difficult than, computing the *expected* sojourn time. We require the distribution because we must make a probabilistic statement about the completion time of an order. For example, if an order arrives 3 hours before the truck departs, what is the probability that the order will make it on that truck? After a time lapse of 1 hour, what is the probability that the order will make it on that truck? If the probability is below the threshhold, then we "abandon" some orders and reconfigure the system to help those with a higher chance of making it on time.

To implement the dynamic worker allocation policies we propose in the dissertation, we need to know the probability that an order in the system will make it onto the next departing truck. Because we must be able to compute this probability *at any time*, we require the state-dependent distribution of sojourn time, where the state of the system is defined by the number of busy servers and the number of orders in queue at any workstation. We develop this model in Chapter 3.

As an intermediate step in the development, we develop an approximation model for the *steady-state* distribution of sojourn time for acyclic queueing networks with multiple servers.

In Chapter 2, we show how an order fulfillment system might use this model to establish a cutoff time that maximizes revenue for premium shipping. In a competitive setting, a firm wishes to set the cutoff time as late as possible, in order to maximize the number of orders that could have the option to choose premium shipping; but not so late that the promised delivery cannot be made. We use our models to show how to make this tradeoff.

In the final major section of the dissertation (Chapter 4), we use the state-dependent sojourn time distribution models from Chapter 3 to design dynamic worker allocation policies to improve service performance according to the NSD metric. We test these policies extensively with simulation to show that, despite intuition to the contrary, it is possible to improve NSD by moving workers shortly before truck departure. We use the insights gained from this modeling to suggest a simple, easy-to-implement policy, which is also very effective.

1.3 Contributions

Contribution 1 *We develop the first approximation model for steady state sojourn time distributions in multi-server acyclic queueing networks.*

Our steady state sojourn time distribution model extends the work of Asmussen and Møller (2001) and You et al. (2002). Given the general interarrival times and general service times of the system, Asmussen and Møller (2001) compute waiting time distributions of multi-server single-stage queues, and You et al. (2002) approximate sojourn time distributions of single-server serial lines based on the characteristics of phase-type distributions. Because our system is a multi-server queueing network, we follow the method of Asmussen and Møller (2001) for the waiting time distribution of a single-stage with multiple servers, and convolve all waiting times and service times in the system using the method of You et al. (2002). Distributions produced by the model agree well with those produced by simulation for a variety of serial and general queueing networks.

Research on steady state sojourn time distributions for queueing networks with general interarrival and service times has been addressed by Shanthikumar and Sumita (1988), You et al. (2002) and Yoon (1994), but their studies were limited to the single-server case. Mandelbaum et al. (1998) develop diffusion approximations for multi-server queueing networks, but their models assume exponential interarrival and service times. Of course, we can find an approximation model for the G/G/c queueing network (Curry and Feldman, 2009), but such models are limited to the expected value of sojourn time.

Contribution 2 *We develop the first approximation model for state-dependent sojourn time distributions in multi-server acyclic queueing networks.*

We develop our model based on the characteristics of phase-type distributions and a Markov process $\{N(t), t \geq 0\}$ to model the all-busy period for a waiting time distribution, where the state is defined as the number of servers in each phase. The state-dependent sojourn time distribution is determined by the number of other orders ahead of an order, the number of servers, and the processing rate of the servers. In contrast to the steady state sojourn time distribution, we do not need to consider the arrival process for the state-dependent sojourn time distribution because it does not affect the sojourn time distribution of an order already in the system. If there are k orders ahead of a particular order in a multi-server queueing system, it can enter service after $k+1$ different waiting times, so the state-dependent sojourn time distribution is the sum of all $k+1$ waiting times and the service time. We approximate each waiting time as a corresponding phase-type distribution based on the Markov process $\{N(t), t \geq 0\}$, and finally all waiting times and the service time are convolved for the single-stage queue.

We extend our work to acyclic G/G/c queueing networks based on the single-stage queue model. First, the state-dependent sojourn time distribution of each workstation is generated after revising the number of orders in front of it at each workstation based on the expected sojourn time of each workstation. And then all state-dependent sojourn time distributions are convolved for the queueing network. We compare distributions generated by the model with simulation.

Little research has been conducted on state-dependent sojourn time distributions for single-stage systems. Whitt (1999) and Nakibly (2002) proposed the method of estimating the waiting time distribution of each customer in a single-stage $G/G/c$ queue. However, they focus on the waiting time distribution using a normal approximation, and their approximation models perform well only for a small queue and a large number of servers. Our single-stage state-dependent sojourn time distribution model focuses on the sojourn time distribution, not waiting time distribution, and can be applied well to any condition. Our approximation model is also different from existing methods because we use the characteristics of the phase-type distribution.

Contribution 3 *We show how to allocate workers dynamically in an order fulfillment system to improve its service performance.*

All of the policies we develop are based on the observation that, without dynamic worker allocation, some orders will just miss a departing truck, and that shifting workers (in a warehousing context) from picking to shipping shortly before the last scheduled truck departure, could cause some of those orders to ship an entire day earlier than they otherwise would. An early question in our research was, "By shifting workers away from picking late in the day, would we hurt performance the *next* day by reducing capacity in picking?" Or, informally, are we "robbing Peter to pay Paul?" In Chapter 4 we show that, indeed, we are robbing Peter to pay Paul; But Peter still has enough cash: it is possible to improve performance—significantly, in some conditions—by moving the right number of workers at the right time everyday.

We test three different policies that use sojourn time distributions, each for a wide range of parameters, and find that the simplest policy performs just as well as more complicated ones. Generally speaking, it seems more important to do anything reasonable, rather than worrying about details of the allocation. We use this observation to construct a very simple policy based on a simple rule of thumb. This simple policy, performs nearly as well as the best policy, and it is much easier to implement in practice.

There is a rich literature on workers moving among tasks in a dynamic fashion in manufacturing and, to a lesser extent, warehousing. It is important to note that in all former papers that we know of the objective is to minimize cycle time, maximize throughput, or perhaps to minimize WIP, but these are not directly related to customer satisfaction. We use NSD as a system performance measure to establish the effect of dynamic worker allocation, and the test results show that worker allocation can improve customer service. We believe an important feature of our research is the application of worker allocation models to systems that are *due-date* or *deadline driven*.

1.4 Organization of the dissertation

The remainder of this dissertation is organized as follows. In the next chapter, we introduce an approximation model of steady state sojourn time distributions for multi-server acyclic queueing networks. In Chapter 3, we present an approximation model of state-dependent sojourn time distributions for multi-server single-stage queues and acyclic queueing networks. In Chapter 4, we describe dynamic worker allocation policies based on sojourn time distribution models which were developed in Chapters 2 and 3. In Chapter 5, we offer conclusions.

Chapter 2

An Approximation Model for Sojourn Time Distributions in Acyclic Multi-server Queueing Networks

We develop an approximation model for the sojourn time distribution of customers or orders arriving to an acyclic multi-server queueing network. The model accepts general interarrival times and general service times, and is based on the characteristics of phase-type distributions. Distributions produced by the model agree well with those produced by simulation for a variety of serial and general queueing networks.

2.1 Introduction

In many order fulfillment systems, the firm makes a delivery promise based on the time of order receipt. Those ordering before a *cutoff time* are "guaranteed" delivery by a particular mode. For example, Amazon's web site famously asks if the customer wants his or her order the next day, and gives the condition that if so, the order must be placed within the next so many hours and minutes. This deadline counts down, in real time, as the customer makes a decision on mode of shipment. Typically, the current time plus this remaining time creates a cutoff time of about 17:30 (at least in the author's time zone), suggesting that after this time Amazon is no longer confident it can make good on its promise. Amazon's service is called "Guaranteed Accelerated Delivery" because if they miss the service promise, they refund the shipping cost.

Promise-to-deliver service is beneficial to the customer because it reduces the uncertainty of delivery time. Most people have experienced waiting for a much anticipated package wondering, "Will it arrive today?" Promise-to-deliver service removes such anxiety. But notice that in such a system the service provider, in fact, manages the expectation of the customer by establishing the cutoff time. Those ordering before the cutoff time expect a certain delivery time, for sure, but those who just miss it have *no* expectation of that delivery. If the adage "service equals perception minus expectation" is true, this puts the retailer or distributor in control of perceived service with respect to delivery time—a very powerful position.

Table 2.1 shows cutoff times (if any) and shipping modes offered by the ten largest internet retailers (Internetretailer, 2009). Six of the ten make an explicit promise on some type of shipping mode. For highly competitive industries such as books or electronics, a later cutoff time gives the firm an advantage over its rivals; however, if the cutoff time is too late, then a significant number of orders might not be processed in time to make the last package carrier pickup that evening (assuming next day service, for now). If the cutoff time is too early, the firm loses both its competitive advantage and the profit on customers who

Table 2.1: Shipping policies of America's top 10 retail business.

Firm	Fastest Shipping Method	Cutoff time	Compensation
1. Amazon.com Inc.	One-day shipping	6:30 p.m.	Refund
2. Staples Inc.	Standard delivery	5 p.m. local	None
3. Dell Inc.	Next business day	None	None
4. Office Depot Inc.	Same day delivery	10 a.m. local	None
5. Apple Inc.	2-Day Shipping	5 p.m. ET	None
6. OfficeMax Inc.	Next business day	5 p.m. local	None
7. Sears Holding Corp.	Overnight air	6 p.m. CT	None
8. CDW Corp.	Next business morning	None	None
9. Newegg.com	Next business day	None	None
10. Best Buy	2-Day Shipping	None	None

(http://www.internetretailer.com, 2009)

otherwise would have ordered premium shipping. How then to establish the cutoff time as late as possible, but not "too late?"

The tradeoff we describe above is similar in spirit to the newsvendor problem, with a potential profit (for premium shipping) and a penalty cost for non-performance. The penalty cost could be in terms of goodwill, or, in Amazon's case, in direct monetary terms. We will assume the penalty cost can be explicitly determined by the firm. Still required for newsvendor-like analysis are the probabilities of performance and non-performance, which in our case require the expected sojourn time distribution for an arriving order. That is, if we know the distribution of sojourn time, then we can determine for any arriving order its probability of meeting or missing the service deadline.

The challenge, then—and the major contribution of our chapter—is to develop a steady state sojourn time distribution for orders arriving to an order fulfillment system, given information about the order stream and processing times for workstations. After covering preliminaries on phase-type distributions in Section 3.2, we describe an approximation model for sojourn time distributions in Section 2.4 and 2.5. With the sojourn time distribution in hand, we can easily calculate the time at which the expected profit for an order equals the expected loss, which then establishes the cutoff time having the highest profit to the firm. We validate our models with simulation in Section 2.5.

To summarize, we make two contributions in this chapter: First, we develop what we believe is the first approximation model of sojourn time distributions for queueing networks with multiple servers per workstation and general interarrival and service times. The approximation we develop in this chapter is quite general, and could be used for any reasonably-sized system for which a queueing network is a reasonable model. Second, we show how an order fulfillment firm might use such an approximation to establish a profit-maximizing cutoff time for promised, premium shipping.

2.2 Related literature

Much of the literature on sojourn time modeling is focused on calculating the mean flow time (or simply the waiting time) in single-stage queues or in queueing networks (Whitt, 1983). Workstations in these systems may have one or many servers. The subject of our work is calculating the *distribution* of sojourn time, a more difficult task.

We divide the literature on sojourn time modeling into four categories, according to whether the models address single-stage systems or networks, and whether they address single or multiple servers per workstation. The simplest case is single-stage, single-server systems. Neuts (1981) describes a matrix-geometric method to calculate the sojourn time distribution for a GI/G/1 system using phase type distributions. Luh and Zheng (2005) implemented Neuts' method in MATHEMATICA. Sengupta (1989) used a bivariate Markov process to model waiting time and queue length distributions in a GI/PH/1 queue. His method is a "continuous analog" of the matrix-geometric method in Neuts (1981). Sengupta also showed that if the interarrival and service time distributions in a single-stage, single-server queue are phase-type, then the waiting time distribution is also phase-type.

Asmussen and O'Cinneide (1998) verify the same result for a single-stage, *multi*-server queue, in a paper extending Sengupta's work to the GI/PH/c case. Their model admits heterogeneous servers. Asmussen and Møller (2001) show how to calculate the waiting time distribution in a GI/PH/c and MAP/PH/c queue, for both homogeneous and heterogeneous servers. Whitt (1999) also addresses single-stage, multi-server systems, but examines state dependent waiting time distributions. For example, he shows how to compute the waiting time distribution for the k-th customer in line in a c-server, single-stage system. Rueda (2003) develop an approximation for the waiting time distribution of single-stage queues with non-stationary interarrival and processing time distributions.

Sojourn time distributions for queueing networks of single-servers have been addressed by Shanthikumar and Sumita (1988), You et al. (2002), and Yoon (1994). Shanthikumar and Sumita (1988) approximate the sojourn time distribution of an M/G/1 queueing system as one of three phase-type distributions (generalized Erlang, exponential, and hyperexponential), based on a "service index," which is defined as the squared coefficient of variation of the total service time of an arbitrary order. Neuts (1981) showed that the convolution of phase type distributions is also phase type. You et al. (2002) used this observation to show how to calculate the sojourn time distribution for queueing networks, with general interarrival and service times. In a paper published in Korean, Yoon (1994) developed a method very similar to that of You et al.

Mandelbaum et al. (1998) develop diffusion approximations for multi-server queueing networks, but their models assume exponential interarrival and service times. Missing from the literature are models of sojourn time distributions for queueing networks with multiple servers, when interarrival and service times can take general distributions. We fill that void here. We believe ours is the first approximation model of sojourn time distributions for queueing networks of multi-server workstations.

Our model is based on the work of Neuts (1981), Asmussen and Møller (2001), and You et al. (2002): We use the bivariate Markov process of Asmussen and Møller to extract the mean and variance of waiting times for each multi-server workstation. We then construct phase type distributions for waiting time and service time based on those means and variances. Then we use the method of You et al. to construct an infinitesimal generator and initial probability vector for the network. With these we can calculate a sojourn time distribution for the network of multi-server queues. For reasons we discuss below, this model produces good results only when the inverse of the squared coefficient of variation ($1/C^2$) is close to an integer. We correct this weakness with a simple but effective interpolation scheme. We demonstrate the validity of our model by comparing it with results from a simulation model.

In the next section, we provide some known results for phase type distributions. We introduce an intuitive model in Section 2.4. In Section 2.5, we develop an interpolation model and test it against a simulation model. We offer conclusions in Section 2.7.

2.3 Phase-type distributions

Consider a Markov process on states $\{1, ..., m+1\}$, having infinitesimal generator

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

where $\mathbf{Q}$ is an $m \times m$ matrix satisfying $Q_{ii} < 0$, for $1 \leq i \leq m$, and $Q_{ij} \geq 0$, for $i \neq j$ (For consistency, we follow the notation of Neuts, 1981). $\mathbf{Q}^0$ is a column vector of size m such that,

$$\mathbf{Qe} + \mathbf{Q^0} = \mathbf{0},$$

where $\mathbf{0}$ is a row vector of zeros and $\mathbf{e}$ is a column vector of ones. The initial probability vector of Q is $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)$ such that $\boldsymbol{\beta}\mathbf{e} = 1$.

We are interested in the time until absorption into state $m+1$, whose distribution Neuts (1981) defines as *phase-type*.

Lemma 2.1 (Neuts, 1981) *The probability distribution $F(\cdot)$ of the time until absorption in the state $m+1$, corresponding to the initial probability vector $(\boldsymbol{\beta}, \beta_{m+1})$ is given by*

$$F(x) = 1 - \boldsymbol{\beta} e^{\mathbf{Q}x}\mathbf{e}.$$

Because $F(\cdot)$ is completely specified by $\boldsymbol{\beta}$ and $\mathbf{Q}$, the pair $(\boldsymbol{\beta}, \mathbf{Q})$ is called a *representation* of $F(\cdot)$.

Neuts (1989) provides the density function on $(0, \infty)$,

$$f(x) = \boldsymbol{\beta} e^{\mathbf{Q}x} \mathbf{Q}^0 = \boldsymbol{\beta} e^{\mathbf{Q}x} (-\mathbf{Q})\mathbf{e}.$$

For each distribution of service or waiting times in a network, we seek a phase-type approximation for which we can generate a matrix-analytic model of the CDF. We use the fact that finite convolutions of phase-type distributions are also phase-type (Neuts, 1981) to generate solutions for a queueing network, as in You et al. (2002).

The phase-type distribution is used to fit a general distribution based on the inflow rate λ and the squared coefficient of variation C^2 of a given positive random variable X. Sauer and Chandy (1975), You et al. (2002), and Tijms (1994) showed different fitting methods that can convert a general distribution to a corresponding phase-type distribution based on the C^2 of the distribution. A general distribution can be approximated with the Erlang distribution, the exponential distribution, or the hyperexponential distribution, based on its first and second moments. We follow the rule of You et al. (2002):

When $C^2 < 1$, we convert a general distribution to an Erlang distribution, $\mathit{Erlang}(m, \mu)$, of order m. The infinitesimal generator is

$$Q = \begin{bmatrix} -\mu & \mu & 0 & \cdots & 0 & 0 \\ 0 & -\mu & \mu & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -\mu \end{bmatrix}, \quad Q^0 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \mu \end{bmatrix},$$

where $m = \lceil \frac{1}{C^2} \rceil$ and $\mu = m\lambda$. The phase-type representation of a general distribution is represented by $\boldsymbol{\beta} = (1, 0, ..., 0)$ and Q.

When $C^2 > 1$, we use the hyperexponential distribution with balanced means, HE_2, of order 2. We use the following normalization for this process,

$$\frac{p}{\mu_1} = \frac{q}{\mu_2}.$$

The phase-type representation of a general distribution is given by $\boldsymbol{\beta} = (p,\ q)$ and

$$Q = \begin{bmatrix} -\mu_1 & 0 \\ 0 & -\mu_2 \end{bmatrix},\ Q^0 = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix},$$

where $p = \frac{1}{2}\left(1 + \sqrt{\frac{C^2-1}{C^2+1}}\right)$, $q = 1 - p$, $\mu_1 = 2p\lambda$ and $\mu_2 = 2q\lambda$.

When $C^2 = 1$, we have an exponential distribution (Tijms, 1994) represented by $\boldsymbol{\beta} = 1$ and Q,

$$Q = -\lambda,\ Q^0 = \lambda.$$

2.4 Intuitive model

We now describe our model, which extends the work of Asmussen and Møller (2001) and You et al. (2002) to produce an approximation of the sojourn time distribution for a network of multi-server queues. For the waiting time distribution of a single-stage, multi-server system, we follow Asmussen and Møller's method, then we convolve waiting times and service times in the system based on the method of You et al. The model requires only the following input, which should be available in most real systems: mean and variance of processing times for each workstation and mean and variance of interarrival times to the system.

The procedure is:

1. Compute the arrival rate and SCV of an arrival process at each workstation using the Queueing Network Analyzer (QNA) method of Whitt (1983).

n	the number of workstations in the general network
λ_i	the total inflow rate into workstation i
γ_i	the external inflow rate into workstation i
p_{ji}	the probability that a product flows from workstation j to i
$C_d^2(i)$	the SCV of the departure stream at workstation i
$C_a^2(i)$	the SCV of the interarrival stream at workstation i
$C_a^2(0,i)$	the SCV of external arrivals at workstation i
$C_s^2(i)$	the SCV of the service time at workstation i
ρ_i	the utilization of workstation i
c_i	the number of servers at workstation i

2. Approximate an interarrival time and service time distribution of each workstation i as a corresponding phase-type distribution $(\boldsymbol{\alpha_i}, T_i)$ and $(\boldsymbol{\beta_i}, S_i)$ based on C_a^2 and C_s^2.

3. Compute the mean and variance of waiting time of each workstation i using the method of Asmussen and Møller. Approximate the waiting time distribution of each workstation i as a corresponding phase-type distribution (γ_i, W_i) based on C_w^2.

4. convolve all waiting times and service times sequentially to generate a phase-type representation of the sojourn time distribution $(\boldsymbol{\zeta}, K)$ using the convolution property of the phase-type distribution.

5. Solve $F(t) = P(T \leq t) = 1 - \boldsymbol{\zeta} e^{Kt}\mathbf{e}, (t \geq 0)$ to obtain the cumulative distribution function (CDF) of the sojourn time distribution.

2.4.1 The flow rate and SCV of the arrival process

We adopt the notation of Curry and Feldman (2009):

To compute the waiting time distribution for each workstation, we must approximate the interarrival time and service time distribution as a corresponding phase-type distribution. Thus, the flow rate λ_i and $C_a^2(i)$ of an interarrival stream at each workstation i must be calculated first, because the fitting distribution is based on the rate and C^2 of a general distribution.

We compute the inflow rate and SCV of the interarrival stream at each workstation i using Whitt's approximations. The total inflow rate into workstation i is

$$\lambda_i = \gamma_i + \sum_{j=1}^{n} p_{ji}\lambda_j.$$

The SCV of the departure stream is approximately,

$$C_d^2(i) = 1 + (1-\rho_i^2)(C_a^2(i)-1) + \rho_i^2 \frac{(C_s^2(i)-1)}{\sqrt{c_i}}.$$

Finally, the SCV of an interarrival stream is approximately,

$$C_a^2(i) = \frac{\gamma_i}{\lambda_i} C_a^2(0,i) + \sum_{j=1}^{n} \frac{\lambda_j p_{ji}}{\lambda_i}[p_{ji}C_d^2(j) + 1 - p_{ji}].$$

From these expressions, we can approximate the interarrival time distribution and service time distribution of each workstation i as a corresponding phase-type distribution $(\boldsymbol{\alpha_i}, T_i)$ and $(\boldsymbol{\beta_i}, S_i)$ based on the flow rate and SCV.

2.4.2 Waiting time for a single stage queue with multiple servers

Asmussen and Møller (2001) proposed a method to compute the steady-state waiting time distribution in a multi-server queue, with each server having a phase-type service time distribution. They used matrix-analytic methods based on the method of Sengupta (1989) and Asmussen and O'Cinneide (1998). Asmussen and O'Cinneide (1998) showed that the GI/PH/c waiting time is always phase-type, and that the number of phases for the homogeneous case should be

$$\begin{pmatrix} m+c-1 \\ c \end{pmatrix},$$

where m is the number of phases of the service time distribution.

They solve the system using a bivariate Markov process $\{X^t, N^t; t \geq 0\}$ to model the all busy period, where X^t is the time since arrival of the last customer to enter service in the all-busy-period, and N^t is the current phase in which the server is working. They proved that the steady state density π of the Markov process $\{X^t, N^t\}$ is matrix exponential:

$$\pi(x) = \pi(0)\exp(Tx),\ x > 0,$$

where T is the rate matrix. Their procedure is as follows:

The rate matrix T satisfies the non-linear integral equation

$$T = S + \int_0^\infty \exp(Tu)A^{(jump)}H(du),$$

where the matrix S contains transition rates without service completion, $A^{(jump)}$ contains transition rates with service completion in $\{N_t\}$, and $H(du)$ is the distribution function of the interarrival times.

The matrix T is solved by iterative procedure and the phase-type representation of the waiting time of the single-stage is defined by $(\boldsymbol{\rho},\ G)$,

$$\rho_i = \frac{\alpha_i \varphi_i}{\boldsymbol{\alpha}\boldsymbol{\varphi}},\ G_{ij} = \frac{\alpha_j T_{ji}}{\alpha_i},$$

where $\boldsymbol{\alpha}= -\pi(0)T^{-1}$ and $\boldsymbol{\varphi} = (T - S)\mathbf{e}$.

Finally, the mean and variance of the waiting time distribution are computed from the first and second moment, where the *n-th* moment is

$$E(X^n) = (-1)^n n! \boldsymbol{\rho} G^{-n}\mathbf{e}.$$

2.4.3 Approximation of sojourn time distribution of the queueing network

Now we are ready to apply the convolution property of phase-type distributions. Neuts (1981) showed that if $F(\cdot)$ and $G(\cdot)$ are both continuous phase-type distributions with representations $(\boldsymbol{\alpha}, T)$ and $(\boldsymbol{\beta}, S)$ of orders m and n, then the convolution $F * G(\cdot)$ is a phase-type distribution with representation $(\boldsymbol{\gamma},\ L)$, where the infinitesimal generator L and initial probability vector $\boldsymbol{\gamma}$ are

$$L = \begin{bmatrix} T & T^0\boldsymbol{\beta} \\ 0 & S \end{bmatrix},$$

$$\boldsymbol{\gamma} = [\boldsymbol{\alpha}, \alpha_{m+1}\boldsymbol{\beta}].$$

We construct the initial probability vector and the infinitesimal generator of a queueing network by convolving all waiting times $(\boldsymbol{\gamma_i}, W_i)$, and service times $(\boldsymbol{\beta_i}, S_i)$ by order in the network. For example, for a k station serial line, the infinitesimal generator K and initial probability vector $\boldsymbol{\zeta}$ are

$$K = \begin{bmatrix} W_1 & W_1^0\beta_1 & 0 & \cdots & 0 & 0 \\ 0 & S_1 & S_1^0\gamma_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & W_k & W_k^0\beta_k \\ 0 & 0 & 0 & \cdots & 0 & S_k \end{bmatrix},$$

$$\boldsymbol{\zeta} = [\ \boldsymbol{\gamma_1} \quad 0 \quad \cdots \quad 0\ ].$$

The cumulative distribution function (CDF) and the probability density function (PDF) are

$$F(t) = P(T \leq t) = 1 - \boldsymbol{\zeta}e^{Kt}\mathbf{e}, (t \geq 0),$$

$$f(t) = \boldsymbol{\zeta}e^{Kt}K^0 = \boldsymbol{\zeta}e^{Kt}(-K)\mathbf{e}, (t \geq 0).$$

Table 2.2: A comparison of the mean sojourn times between the intuitive model and simulation for a 3-workstation serial line, when all $1/C^2$ are integers.

	$E[T]$	C^2	ρ	Mean sojourn time (hrs)		% difference
				Simulation	Intuitive	
Interarrival	0.5	0.5		5.79	5.77	-0.23
Workstation 1	1.8	0.5	0.6			
Workstation 2	2.2	0.5	0.73			
Workstation 3	1.5	0.5	0.5			

In a more general queueing network, flow between workstations depends on product routings. We estimate the sojourn time distribution for a random order by approximating the CDF of all possible "serial lines" (paths in the network) and mixing those CDFs according to the probabilities of taking those paths. Because our model is based on serial line analysis, we are able to model only acyclic queueing networks. We give an example in a later section.

2.4.4 Numerical results

To validate the intuitive model, we compare it with results of a simulation built in Arena 10.0. We assume processing and interarrival times in the simulation model are Gamma distributed.

Example 2.1 Consider a serial line of 3 workstations, with 6 servers per workstation. Information on the interarrival and processing times are given in Table 2.2.

We can approximate the interarrival time and three service time distributions as $Erlang(m, \mu)$ distributions according to the rule of You et al. (2002) because all C^2 are less than 1. Note that in this case all $\lceil \frac{1}{C^2} \rceil$ and $1/C^2$ have the same value of 2. The mean sojourn times in Table 2.2 show that the intuitive model is close to the simulation result. Figure 2.1 shows the sojourn time distributions from the intuitive model and from the simulation, and we see nice agreement between them.

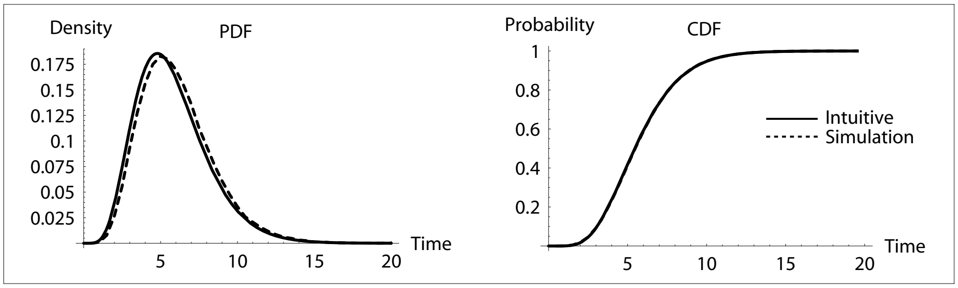


Figure 2.1: Sojourn time distributions for Example 1, when all $1/C^2$ are integers.

Table 2.3: A comparison of the mean sojourn times between the intuitive model and simulation when all $1/C^2$ are not integers.

	$E[T]$	C^2	ρ	Mean sojourn time (hour)		% difference
				Simulation	Intuitive	
Interarrival	0.5	0.8				
Workstation 1	1.8	0.6	0.6	5.97	5.77	-3.35
Workstation 2	2.2	0.5	0.73			
Workstation 3	1.5	0.9	0.5			

Example 2.2 Consider the same 3-workstation serial line with $C_a^2 = 0.8$, $C_s^2(1) = 0.6$, $C_s^2(2) = 0.5$ and $C_s^2(3) = 0.9$.

The mean sojourn time and sojourn time distribution of this system are shown in Table 2.3 and Figure 2.2 respectively. The percent difference of mean sojourn time between the two models is much higher, and the distributions appear not to be so similar. The intuitive model produces the same result when $1/C^2$ falls into the interval $\left(\left\lfloor \frac{1}{C^2} \right\rfloor, \left\lceil \frac{1}{C^2} \right\rceil\right)$, but we get a new simulation result with a new C^2. For example, although C_a^2 changed from 0.5 to 0.8, the interarrival time distribution is still fitted with the same $Erlang(m, \mu)$ distribution, because $\left\lceil \frac{1}{C^2} \right\rceil$ is the same for both cases.

This example illustrates a characteristic of the model that we have confirmed with more thorough testing: the intuitive model produces good results only when $1/C^2$ is close to an integer value (under the $C^2 < 1$ condition). When $1/C^2$ is between integer values (say, for example, when $C^2 = 0.75$), the intuitive model produces less satisfactory results. In fact,

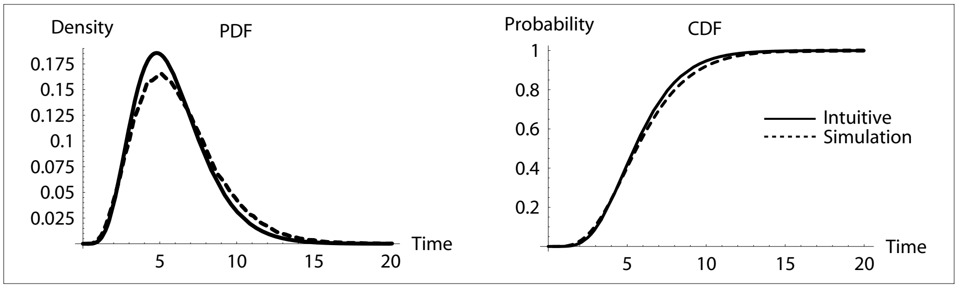


Figure 2.2: A comparison of sojourn time distributions between the intuitive model and simulation when all $1/C^2$ are not integers.

the model produces the same sojourn time distribution for all values of C^2 having a common value of $\lceil \frac{1}{C^2} \rceil$.

2.5 Interpolation model

A naïve attempt to correct for this deficiency by replacing the ceiling operator with a `round( )` function produced even worse results. We propose, then, the following interpolation scheme:

1. Produce CDF $F(t)$ using $\lceil \frac{1}{C^2} \rceil$ phases for each workstation.

2. Produce CDF $G(t)$ using $\lfloor \frac{1}{C^2} \rfloor$ phases for each workstation.

3. Compute the mixed CDF, $H(t) = \alpha G(t) + (1 - \alpha) F(t)$, where α is an interpolation coefficient.

The interpolation coefficient α is based on two observations. First, if $1/C^2$ for interarrival and processing times is close to $\lceil \frac{1}{C^2} \rceil$, then we should expect α to be close to zero, and if $1/C^2 \approx \lfloor \frac{1}{C^2} \rfloor$, α should be close to one. Second, interarrival stream and workstation processing times will have different levels of influence on the sojourn time distribution, depending on whether they appear early or late in the (serial) process.

These observations lead us to a regression model of the squared coefficients of variation for each distribution in the process. To address the first observation, we define a *fitting* *coefficient* for each interarrival and processing time distribution. The fitting coefficient is an interpolation coefficient reflecting the nearness of the squared coefficient of variation to successive values of C^2 when $1/C^2$ takes on an integer value. For example, if the SCV of the arrival process $C_a^2 = 0.8$, we compute a fitting coefficient $\delta_0 = 0.6$, because $0.5 + 0.6(1 - 0.5) = 0.8$, or said another way, 0.8 is 60% of the way between 0.5 and 1. We can compute fitting coefficients δ_k for each workstation k in a similar fashion.

Next, we determine the different levels of influence of the fitting coefficients with a regression model. We use the intuitive model to establish the dependent variable (mean sojourn time) under possible combinations of SCVs, which function as the independent variables. We test the intuitive model for all possible combinations of three SCVs (1, 1/2, 1/3), each corresponding to an integer value of $1/C^2$. For example, the regression for two workstation serial lines must consider three distributions—the interarrival times and service times for two workstations, and there are $3^3 = 27$ combinations. We excluded SCVs less than 1/3 because they have very little effect on mean sojourn time.

There is one last detail: we must consider the possible effects of utilization on the mean sojourn times and therefore on the coefficients ω_i produced by the regression. We ran two cases ($\rho = 0.5,\ 0.85$ for each workstation in the system), and found that the resulting coefficients ω_i are different. Which values to use, given that we seek a single set of coefficients for any serial line of a specified length? We answered this question by using for each distribution the average of the two values, for two reasons: (1) For most systems in practice, the utilization of workstations in a system are different anyway, and (2) the results of the model are not very sensitive to the precise values of the ω_i's.

Table 2.4 contains results of the regression analysis, which are the weight coefficients ω_i for 2, 3, and 4 station serial lines. In general, the interarrival stream and the first workstation account for the largest influence on the sojourn time distribution and the influence decreases as it goes downstream.

Table 2.4: The weight coefficient of serial lines with 2, 3 and 4 workstations.

Workstation	ω_0	ω_1	ω_2	ω_3	ω_4
2	0.493	0.313	0.194		
3	0.412	0.292	0.174	0.123	
4	0.353	0.262	0.180	0.123	0.081

Now that we have the fitting coefficients δ_i and weight coefficients ω_i. We calculate the interpolation coefficient

$$\alpha = \sum_{i=0}^{k} \omega_i \delta_i,$$

where index zero represents interarrival times. We should emphasize the conditions for which the ω_i's in Table 2.4 are relevant: These coefficients should be appropriate for any serial lines of these lengths for which the Gamma distribution is a reasonable approximation of processing times.

Before fully testing the interpolation model, we need to show that it is better than the simpler, intuitive model. Figure 2.3 shows the differences in mean sojourn times among the simulation and the two approximation models when each workstation has the same utilization ρ=0.5 and SCVs. The interpolation and intuitive models are the same when all $1/C^2$ are integers, but when they are not, the interpolation model is closer to the simulation. The intuitive model also corrects the deficiency noted in Table 2.3: for this problem, the difference in means is reduced to -1.26% (mean sojourn time = 5.90 hours).

2.5.1 Validation

To test the performance of the model, we consider four queueing networks: The first is a small serial line with 3 workstations (Figure 2.4), the second a simple acyclic queueing network with 4 workstations (Figure 2.5), the third is a large-scale system with almost 100 servers in 3 workstations, and the last is a long serial line with 8 workstations. We wish to confirm the effectiveness of the interpolation model on a basic serial line and general queueing network through the first two experiments, and wish to know computation time and accuracy

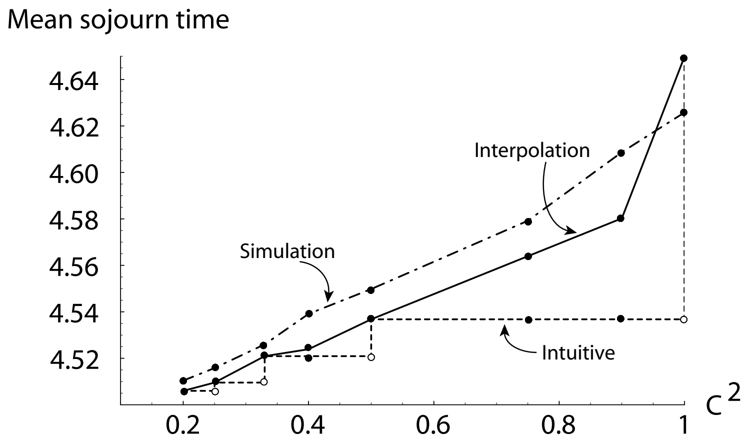


Figure 2.3: A comparison of mean sojourn times among three models(ρ=0.5). Dots indicate data; lines are inserted for visual clarity only.

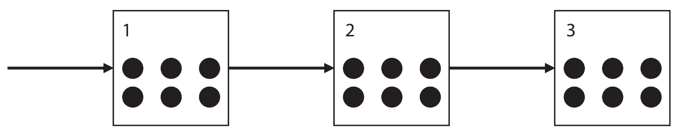


Figure 2.4: A serial line with 3 workstations.

of the interpolation model as we increase the number of servers and workstations. In Figures 2.4 and 2.5, the black circle in each workstation represents a server, and in the case of the general network, a customer or order departing from the first workstation selects its follow on workstation with probability p.

Example 2.3 Consider a 3-workstation serial line with 6 servers per workstation (Figure 2.4). All servers have the same capacity ($E[T] = 1.5$), and interarrival times are adjusted to produce the appropriate utilization ($\rho = 0.5,\ 0.85$).

We compare simulation and model results over different values of SCV for interarrival and processing times. Table 2.5 shows the mean sojourn times for the interpolation model and simulation.

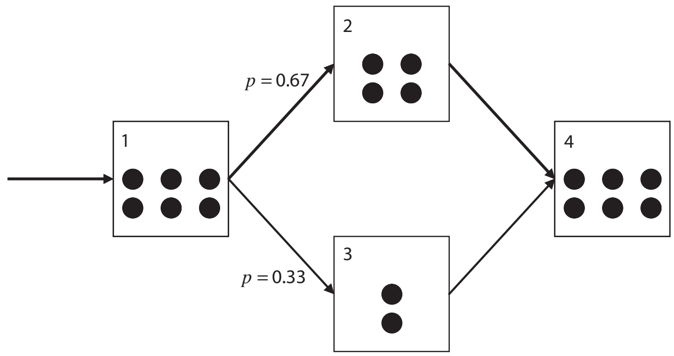


Figure 2.5: A general network with 4 workstations.

Table 2.5: Mean sojourn times for the interpolation and simulation models of the serial line.

ρ	C_a^2	$C_s^2(1)$	$C_s^2(2)$	$C_s^2(3)$	Interpolation	Simulation	% difference
0.85	1	1	1	1	7.63	7.57	0.83
	0.75	0.75	0.75	0.75	6.77	6.77	-0.04
	0.57	0.7	0.6	0.9	6.44	6.43	0.05
	0.45	0.75	0.4	0.9	6.13	6.19	-0.35
	0.4	0.6	0.26	0.8	5.88	5.89	-0.13
	0.33	0.33	0.33	0.33	5.57	5.48	1.71
0.5	1	1	1	1	4.65	4.62	0.51
	0.75	0.75	0.75	0.75	4.53	4.58	-0.30
	0.57	0.7	0.6	0.9	4.57	4.55	0.29
	0.45	0.75	0.4	0.9	4.54	4.54	0.04
	0.4	0.6	0.26	0.8	4.53	4.53	-0.03
	0.33	0.33	0.33	0.33	4.52	4.53	-0.23

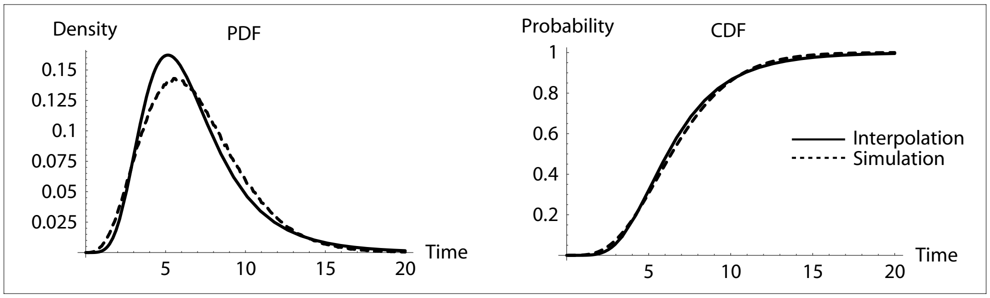


Figure 2.6: CDF and PDF of the serial line (ρ=0.85, $C^2 = 0.75,\ 0.75,\ 0.75,\ 0.75$).

The percent difference of the mean sojourn time between the two models under high utilization is a little bit higher than in the low utilization case. We conjecture that this is due to the amount of waiting time at each workstation. In the low utilization case, there is very little waiting in front of each workstation, so inaccuracies in the waiting time model do not have much effect on the estimate of the mean sojourn time. In the high utilization case, model inaccuracies have a greater effect. However, we see that the percent differences between the interpolation model and the simulations are still less than 2 percent. Notice that this statement only pertains to the *mean*, when what we seek is the distribution; that is, agreement in the means is only an *indication* that the model is effective, not a proof.

Figures 2.6–2.11 show the distributions for representative cases. We use the Anderson-Darling test (Law and Kelton, 2000) to check the graphical agreement of the distributions between simulation and approximation results. We also include the percent difference of the 90^{th} and 95^{th} percentiles. As shown in Table 3.8, we accept that there is no difference between the distributions based on the A-D test, and the percent differences of the 90^{th} and 95^{th} percentiles are less than 3 percent.

Figures 2.6–2.11 indicate that the model consistently has a higher peak in the PDF than does the simulation model. We believe this is due to our approximation of waiting time distributions, which for all cases we tested had $C^2 > 1$, and therefore led us to the hyperexponential distribution as an approximation. Because the hyperexponential distribution is

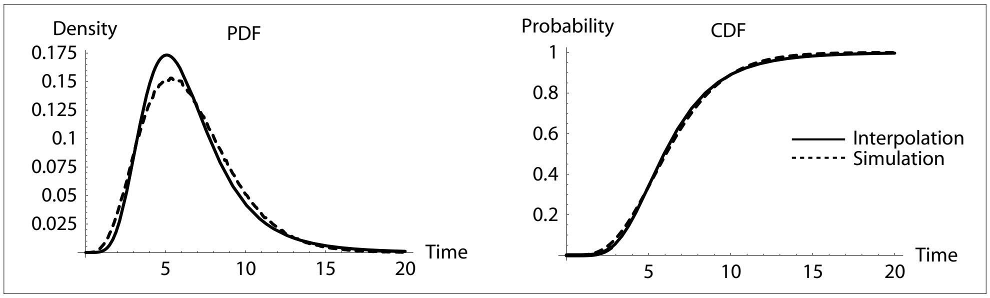


Figure 2.7: CDF and PDF of the serial line (ρ=0.85, $C^2 = 0.57,\ 0.7,\ 0.6,\ 0.9$).

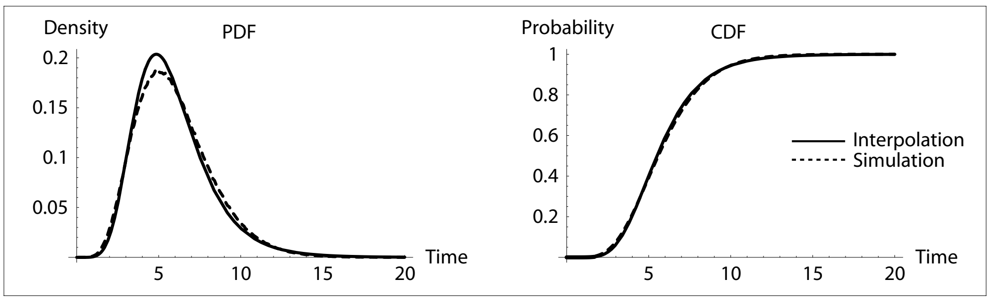


Figure 2.8: CDF and PDF of the serial line (ρ=0.85, $C^2 = 0.4,\ 0.6,\ 0.26,\ 0.8$).

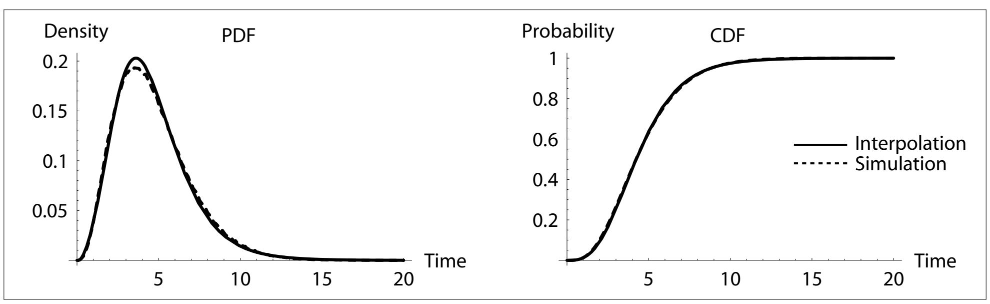


Figure 2.9: CDF and PDF of the serial line (ρ=0.5, $C^2 = 0.75,\ 0.75,\ 0.75,\ 0.75$).

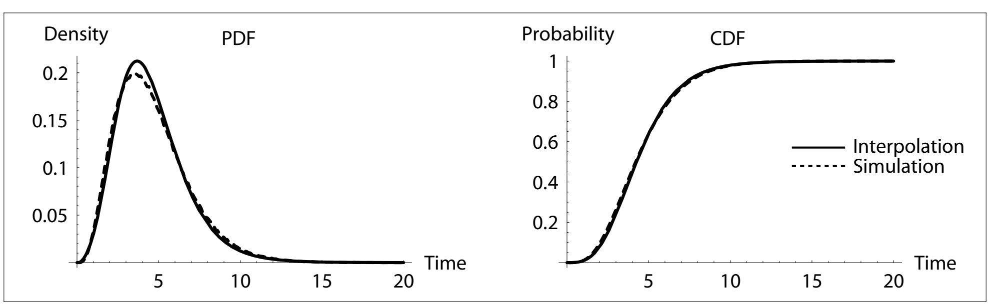


Figure 2.10: CDF and PDF of the serial line (ρ=0.5, $C^2 = 0.57,\ 0.7,\ 0.6,\ 0.9$).

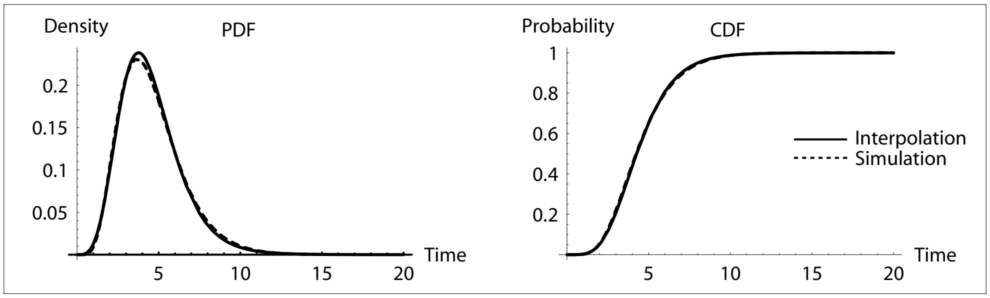


Figure 2.11: CDF and PDF of the serial line (ρ=0.5, $C^2 = 0.4,\ 0.6,\ 0.26,\ 0.8$).

Table 2.6: Anderson-Darling tests for the serial line test case.

ρ	C_a^2	$C_s^2(1)$	$C_s^2(2)$	$C_s^2(3)$	A-D	$\alpha = 5\%$	Decision	% difference 90^{th}	% difference 95^{th}
	0.75	0.75	0.75	0.75	1.691		accept	0.44	-0.32
0.85	0.57	0.7	0.6	0.9	1.726	2.492	accept	0.25	-0.30
	0.4	0.6	0.26	0.8	1.529		accept	-1.15	-0.38
	0.75	0.75	0.75	0.75	0.639		accept	1.92	-0.26
0.5	0.57	0.7	0.6	0.9	1.366	2.492	accept	-2.35	-1.99
	0.4	0.6	0.26	0.8	1.542		accept	-1.65	-1.88

skewed to the left, it may cause our distributions of sojourn time also to be overly skewed to the left.

Example 2.4 Consider the general acyclic queueing network with 4-workstations in Figure 2.5. Workstations 1 and 4 have 6 servers, and workstations 2 and 3 have 4 and 2 servers respectively. All servers have the same capacity ($E[T] = 1.5$), and interarrival times are adjusted to produce the appropriate utilization ($\rho = 0.5,\ 0.85$).

Table 2.7 shows the mean sojourn time of the interpolation model and the simulation model in a 4-workstation general queueing network. Similar to the serial line case, the interpolation model gives better results under low utilization than under high utilization. The percent differences are all less than 5 percent.

Figures 2.12–2.17 show distributions from the model and simulation. The Anderson-Darling tests are all acceptable and values for the 90^{th} and 95^{th} percentiles show fairly nice agreement.

Table 2.7: Mean sojourn times for a 4-workstation general network.

ρ	C_a^2	$C_s^2(1)$	$C_s^2(2)$	$C_s^2(3)$	$C_s^2(4)$	Interpolation	Simulation	% difference
0.85	1	1	1	1	1	8.99	8.91	0.89
	0.75	0.75	0.75	0.75	0.75	7.78	7.98	-2.61
	0.57	0.7	0.6	0.6	0.9	7.31	7.49	-2.42
	0.45	0.75	0.4	0.4	0.9	6.73	7.05	-4.48
	0.4	0.6	0.26	0.26	0.8	6.38	6.68	-4.46
	0.33	0.33	0.33	0.33	0.33	6.11	6.27	-2.62
0.5	1	1	1	1	1	4.76	4.81	-1.11
	0.75	0.75	0.75	0.75	0.75	4.73	4.74	-0.17
	0.57	0.7	0.6	0.6	0.9	4.68	4.69	-0.36
	0.45	0.75	0.4	0.4	0.9	4.60	4.66	-1.12
	0.4	0.6	0.26	0.26	0.8	4.59	4.64	-0.94
	0.33	0.33	0.33	0.33	0.33	4.58	4.62	-0.84

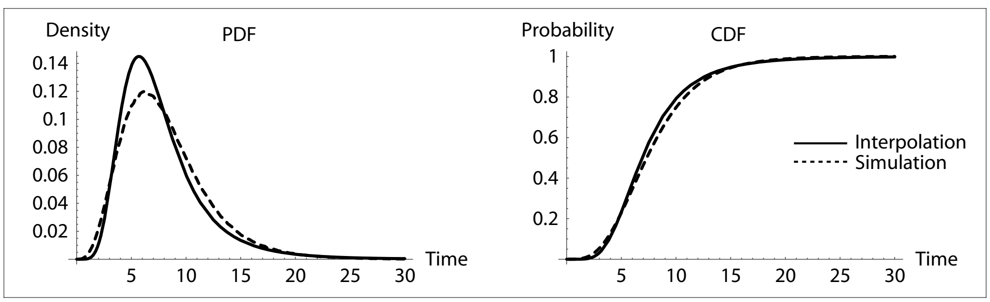


Figure 2.12: CDF and PDF of the general network(ρ=0.85,$C^2 = 0.75, 0.75, 0.75, 0.75, 0.75$).

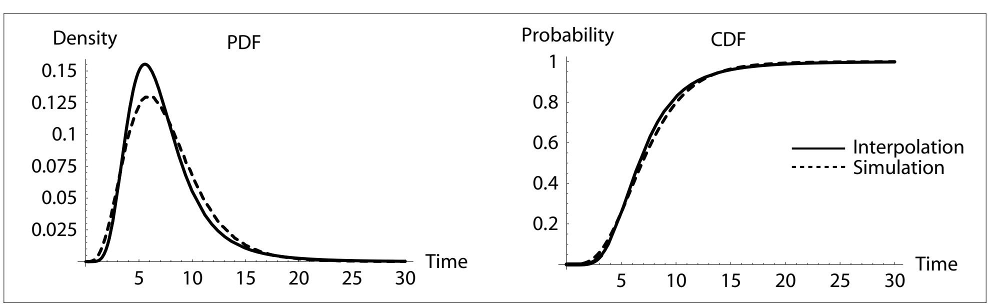


Figure 2.13: CDF and PDF of the general network (ρ=0.85, $C^2 = 0.57,\ 0.7,\ 0.6,\ 0.6,\ 0.9$).

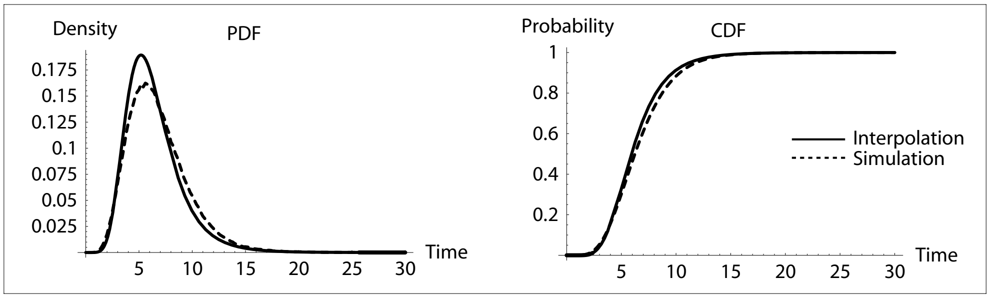


Figure 2.14: CDF and PDF of the general network (ρ=0.85, $C^2 = 0.4,\ 0.6,\ 0.26,\ 0.26,\ 0.8$).

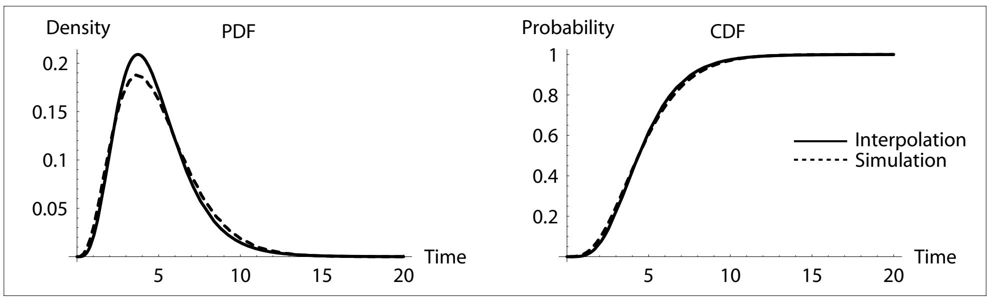


Figure 2.15: CDF and PDF of the general network (ρ=0.5,$C^2 = 0.75, 0.75, 0.75, 0.75, 0.75$).

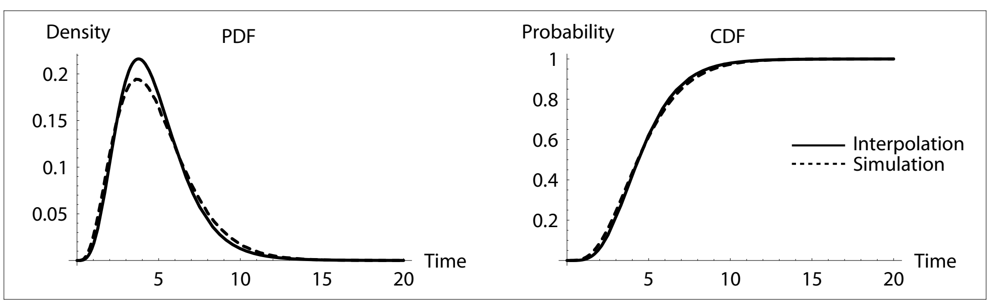


Figure 2.16: CDF and PDF of the general network (ρ=0.5, $C^2 = 0.57,\ 0.7,\ 0.6,\ 0.6,\ 0.9$).

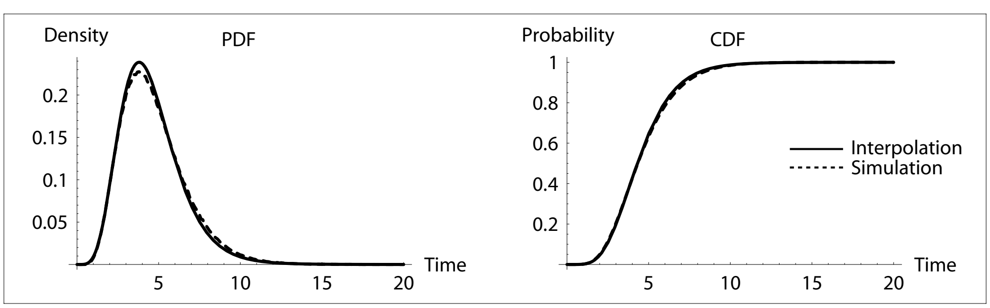


Figure 2.17: CDF and PDF of the general network (ρ=0.5, $C^2 = 0.4,\ 0.6,\ 0.26,\ 0.26,\ 0.8$).

Table 2.8: Anderson-Darling tests for the general queueing network.

ρ	C_a^2	$C_s^2(1)$	$C_s^2(2)$	$C_s^2(3)$	$C_s^2(4)$	A-D	$\alpha = 5\%$	Decision	% difference	
									90^{th}	95^{th}
	0.75	0.75	0.75	0.75	0.75	2.343		accept	-4.02	-1.33
0.85	0.57	0.7	0.6	0.6	0.9	2.401	2.492	accept	-2.94	0.22
	0.4	0.6	0.26	0.26	0.8	2.046		accept	-5.70	-5.27
	0.75	0.75	0.75	0.75	0.75	1.475		accept	-3.57	-3.28
0.5	0.57	0.7	0.6	0.6	0.9	2.086	2.492	accept	-3.98	-4.33
	0.4	0.6	0.26	0.26	0.8	1.454		accept	-2.93	-3.05

Table 2.9: The system information of the Example 2.5

	$E[T]$	C_s^2	C_a^2	ρ	Number of workers
Interarrival	0.23		0.75		
Workstation 1	7.80	0.70		0.85	40
Workstation 2	4.50	0.70		0.78	25
Workstation 3	5.80	0.80		0.84	30

Example 2.5 Consider a serial line with 3-workstations, and 95 workers in the system. We assume that both the interarrival and service times follow general distributions. Table 2.9 shows the system information in detail.

We compare the mean sojourn time and distribution between the interpolation and simulation model. We also record computation time using a 2.4 GHz Intel Core 2 Duo processer, and it takes 1.55 hours. Table 2.10 and Figure 2.18 show mean sojourn time and sojourn time distribution.

We see that the interpolation model also works well for the large scale system, but the computation time has increased significantly. As mentioned before, the size of the infinitesimal generator K and the initial probability vector $\boldsymbol{\zeta}$, which decide the sojourn time distribution, are determined by the size of the matrix of the waiting time distribution W

Table 2.10: Mean sojourn time for the Example 2.5.

	Interpolation	% difference	Simulation	% difference	
				90^{th}	95^{th}
Mean sojourn time	18.60	0.30	18.54	0.01	0.01

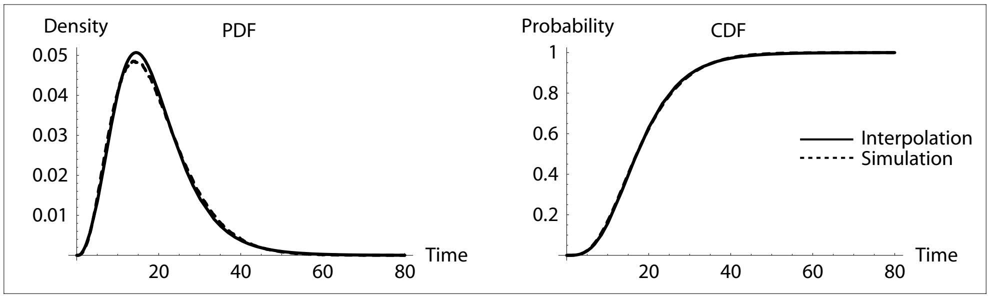


Figure 2.18: PDF and CDF of the Example 2.5.

Table 2.11: The system information for Example 2.6.

	$E[T]$	C_s^2	C_a^2	ρ	Number of workers
Interarrival	0.23		0.75		
Workstation 1	2.90	0.70		0.84	15
Workstation 2	2.50	0.60		0.91	12
Workstation 3	1.40	0.80		0.87	7
Workstation 4	1.50	0.65		0.82	8
Workstation 5	2.75	0.50		0.91	13
Workstation 6	2.20	0.85		0.87	11
Workstation 7	3.10	0.70		0.79	17
Workstation 8	1.50	0.60		0.82	8

and service time distribution S. Here, W is determined by $\begin{pmatrix} m+c-1 \\ c \end{pmatrix}$, where m is the number of phases of the service time distribution and c is the number of servers, and S is determined by only m. So if we increase c, it affects mainly to the size of matrix W, and this creates a significant computational burden.

Example 2.6 Consider a serial line with 8-workstations, and 91 workers in the system. We assume that both the interarrival and service times follow general distributions. Table 2.11 shows the system information in detail.

We compare the mean sojourn time and distribution between the interpolation and simulation model. Table 2.12 and Figure 2.19 show mean sojourn time and sojourn time distribution, but the results are not good. The percent difference in the means is greater

Table 2.12: Mean sojourn time for Example 2.6.

	Interpolation	% difference	Simulation	% difference 90^{th}	% difference 95^{th}
Mean sojourn time	20.91	-7.71	22.65	7.51	-6.31

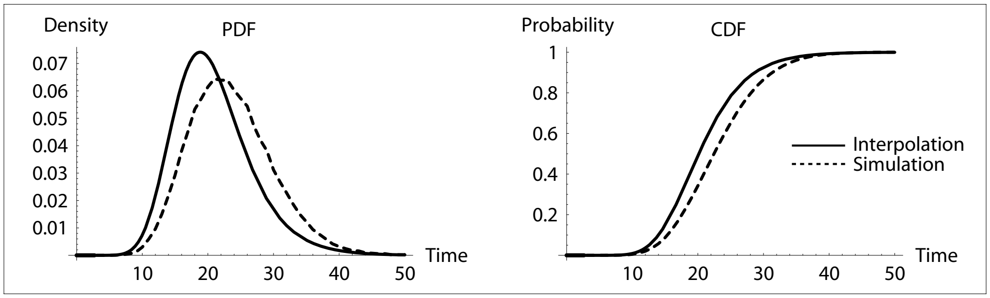


Figure 2.19: PDF and CDF for the long serial line in Example 2.6.

than 5%, and the percentiles of 90^{th} and 95^{th} also are greater than 5%. However, it takes just about 13 minutes because the number of servers in each workstation is not too large (around 15), and matrix size of each waiting time distribution is of reasonable size.

2.6 Determining an optimal cutoff time

Consider an order fulfillment center with three major functional areas—picking, packing, and shipping, all of which function as multi-server queues. Assume that each functional area processes orders individually. For most warehouses, this is a reasonable assumption for the packing and shipping areas, but for manual warehouses, picking is often done in batches. We assume for purposes of illustration a suitable part-to-picker automated system for which a multi-server queue is an appropriate model. Nearly every order moves from picking to packing to shipping, sequentially, so we can model the process as a serial line, with multiple workers in each stage.

Assume the firm realizes marginal shipping profit r for an order successfully delivered for premium shipping. If the firm promises premium service but fails to deliver, it realizes marginal cost c. For most applications, we would expect $r \ll c$ (in Amazon's case, c is

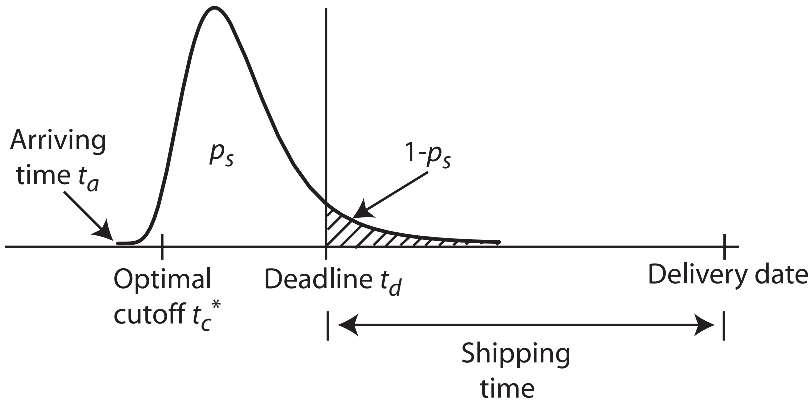


Figure 2.20: Relationship between cutoff time and probability of success of an arriving order.

the entire cost of shipping). We seek a cutoff time t_c^*, which gives the maximum expected shipping profit.

Let probability of success p_s be the probability that an arriving order will make it on the next departing truck. We can compute p_s of an arriving order from the steady state sojourn time distribution. Let a continuous random variable T be the sojourn time of an order. Then,

$$p_s = P[T < t_r],$$

where t_r is remaining time, equal to deadline t_d minus arrival time t_a.

Figure 2.20 shows the relationship between arrival time and probability of success for an arriving order. As the arrival time gets closer to the deadline (shift the distribution to the right), the probability of success p_s gets smaller. Consequently, the expected shipping profit decreases, and the expected shipping loss increases. When the expected marginal profit of an arriving order equals the expected marginal loss, we have the optimal cutoff time t_c^*.

Formally,

$$\begin{aligned} p_s^* r &= (1 - p_s^*) c \\ p_s^* &= \frac{c}{r + c}, \end{aligned}$$

and the optimal cutoff time t_c^* is determined by

$$p_s^* = P[T < t_r],$$

where $t_r = t_d - t_c^*$.

For example, consider an order fulfillment system defined by the third example in Table 2.5. Suppose that the delivery truck makes its final pickup at t_d=17:00 each day. Marginal shipping profit and shipping refund per item are \$5 and \$20, respectively. Management wishes to establish the optimal cutoff time.

First, we compute p_s^* corresponding to the break-even point between marginal shipping profit and loss,

$$p_s^* = \frac{c}{r+c} = \frac{20}{20+5} = 0.8$$

Second, we find $t_c^* = t_d - t_r$ using the CDF of the steady state sojourn time distribution:

$$\begin{aligned} p_s^* &= 0.8 = P(T \leq t_r) = 1 - \boldsymbol{\zeta} e^{Kt_r}\mathbf{e} \\ t_r &= 8.365 \ hours \\ t_c^* &= 09:38 \end{aligned}$$

To test the correctness of our approach, we ran a simulation experiment to check the potential profit for a range of cutoff times. The results are in Figure 2.21. For the simulations, we assumed customers ordering after midnight and before the cutoff time were guaranteed premium shipping, and those ordering between the cutoff time and midnight (thereby missing the deadline) were not offered the option. As expected, the profit is low when the cutoff time is early, because too few customers have the option of ordering premium shipping. When the cutoff time is too late, many orders arriving just before the cutoff do not make it onto the

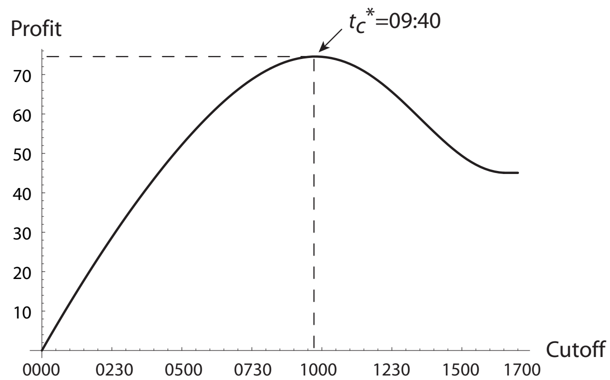


Figure 2.21: Profit curve for a range of cutoff times.

truck that night, and the firm must incur a loss and refund the shipping. The peak profit corresponds to a cutoff time of 09:40, which agrees nicely with our predicted 09:38.

2.7 Chapter conclusions

The ability to approximate the sojourn time distribution for orders in a manufacturing or order fulfillment system gives a company the ability to make statements about service performance. Such statements could be used purely as internal metrics, or, as in the case of published cutoff times, they could be used to make guarantees to customers.

We have developed an approximation model for the steady state sojourn time distribution of entities arriving to an acyclic network of multi-server queues. In general, the distributions produced by our model agree well with simulations of identical systems for the cases we have tested. Means of the approximated distributions were generally within 3 percent of the simulated means, and the distributions produced met the requirements of the Anderson-Darling test. Our test of a long serial line (8 workstations) produced less satisfactory results, suggesting that the quality of the results would degrade as the size and complexity of the network increases. Nevertheless, the model could be used by any organization interested in making probabilistic statements about the time an order or customer will spend in a network of multi-server queues, including the order fulfillment systems which motivated our research.

The method we have developed requires only basic statistics about interarrival and processing times, which can take on any general distribution. For processing times this seems straightforward, but an acknowledged weakness of our model—at least for order fulfillment systems—is the assumption of stationary interarrival times. We suspect that for many systems, this assumption is appropriate and for some it is plainly inappropriate, but we have no basis to make conclusive statements. Users facing a practical problem with highly non-stationary interarrival times would have to verify the approximation model with simulation or with empirical data, in order to proceed with confidence.

The mention of empirical data raises an interesting question, "Why use these models at all? Why not use empirical sojourn time distributions to establish the cutoff time?" For establishing a cutoff time under known and constant conditions, empirical data is likely the best approach, but suppose the optimal cutoff time is deemed unacceptably early by management. In this case, it would be helpful to recompute optimal cutoff times under different conditions (numbers of workers, processing times, etc.), which would be impossible from empirical data.

Chapter 3

Predicting Departure Times in Multi-Stage Queueing Systems

We develop an approximation model for the state-dependent sojourn time distribution of customers or orders in a multi-stage, multi-server queueing system, when interarrival and service times can take on general distributions. The model can be used to make probabilistic statements about the departure time of a customer or order, given the number and location of customers currently in process or waiting, and these probabilities can be recomputed while waiting at any point during the sojourn time. The model uses phase-type distributions and a new method to estimate the remaining processing times of customers in service when the sojourn time distribution is computed.

3.1 Motivation

Service systems often make explicit statements—perceived by customers, effectively, as promises—about how long a wait will be. For example, call centers may claim, “expected wait time is 3 minutes,” or an internet bookseller may promise, “order in the next 2 hours and receive it tomorrow.” To make such promises, a firm must estimate both processing times and waiting times, whose sum is the sojourn time. In stochastic environments, the result must be a distribution, from which service estimates or promises are made. When offered or called upon in real time, these service promises must also consider the number of customers already in the system, that is, the system *state*. In the examples above, the state of the system is defined by other callers or orders already in queue.

The relevance of state-dependent waiting times for call centers is well-recognized (Whitt, 1999). These problems are typically modeled with a single, multi-server queue, but many service systems require a multi-stage representation. For example, the motivation for our research is a large distribution center, which operates as a series of multi-server queues corresponding to the picking, packing, and shipping operations. To make real-time decisions about workforce allocation, we sought the probability that a particular order would complete its processing before the last truck departed for the day. This probability can only be computed with a state-dependent sojourn time distribution.

Models such as the one we develop here seem all the more important when one considers that simulation is an ineffective means of developing these distributions. For general service times (which we assume), the state of the system must include not only queue lengths and numbers of busy servers, but also the remaining processing time for each order being served. To our knowledge, there is no accepted method of simulating the “remaining times” of orders already in process. The brute force simulation method, which we describe and use below, observes the state of the system upon each arrival and records results only for orders encountering the system state of interest. Such simulations can take hours or even days to generate a single distribution. We believe our models offer a better approach.

Research on sojourn time distributions can be roughly classified into two categories: steady state waiting time and state-dependent waiting time. A steady state sojourn time distribution only has meaning when a customer or order arrives to the system, whereas a state-dependent sojourn time distribution can be computed at any time while in the system, not just upon arrival. Our interest in this paper is a state-dependent sojourn time distribution, which includes the waiting time and service time distributions.

There is a rich literature on steady state sojourn time distributions. Neuts (1981), Sengupta (1989) and Luh and Zheng (2005) showed how to generate the sojourn time distribution of a single stage queue with a single server using a matrix geometric method and a "continuous analog" of the matrix-geometric method, respectively. Asmussen and O'Cinneide (1998) and Asmussen and Møller (2001) extended Sengupta's analysis of the GI/PH/1 queue to the multi-server case. They showed the steady state waiting time distribution in a GI/PH/c queue is also phase-type.

Shanthikumar and Sumita (1988) and You et al. (2002) suggested an approximation model for queueing networks of single servers. Shanthikumar and Sumita (1988) approximated the sojourn time distribution as a phase-type distribution based on the "service index," and You et al. (2002) introduced an approximation using the convolution property of the phase-type distribution. Steady state sojourn time distributions for queueing networks of multiple servers was studied by Mandelbaum et al. (1998) and Gue and Kim (2009a). Mandelbaum et al. (1998) addressed diffusion approximations for M/M/c queueing networks, and Gue and Kim (2009a) developed an approximation model for G/G/c queueing networks based on the characteristics of the phase-type distribution.

A state-dependent sojourn time distribution provides managers or system controllers with the ability to post real-time information to customers or to take real-time actions to improve system performance. Call centers are one of the most active research areas on this subject. Given the system state at the time of estimation, Whitt (1999) proposed a method of estimating the waiting time distribution of each customer in a single stage G/G/c queue.

He estimated the waiting time distribution using a Normal approximation for a large call center with many servers. Nakibly (2002) suggested an approximation model of the waiting time distribution in a multi-server queue by calculating iteratively the waiting time of each customer in the queue. To increase customer satisfaction by announcing expected waiting time to a customer, Jouini and Dallery (2006) investigated the waiting time distribution for multiclass, multi-server call centers with exponential arrival and service times.

Our work differs from existing research in important ways. First, for single-stage systems, our method is effective for small and medium sized systems; whereas the method of Whitt (1999) is effective only for large systems. Second, existing research addresses only single stage systems, whereas we extend our methodology to address multi-stage systems, and even simple, acyclic networks.

We focus on the sojourn time instead of waiting time to extend our interest to manufacturing and warehousing systems. Our approximation model is based on the phase-type distribution. In contrast to the steady state sojourn time distribution, we do not need to consider the arrival process for a state-dependent sojourn time distribution, because it does not affect the sojourn time distribution of a order in the system. Throughout this paper, we assume interarrival times and service times follow general distributions and that all servers in a workstation are homogeneous; that is, they have same processing time distributions.

The rest of this paper is organized as follows: in Section 3.2, we introduce characteristics of the phase-type distribution, which is the fundamental concept behind our models. Also, we introduce the method of fitting a general distribution as a corresponding phase-type distribution. In Section 3.3, we introduce an approximation model of the sojourn time distribution for a multi-server queueing system with exponential service times. In Section 3.4, we present an approximation model for general service times and apply it to some numerical examples. In Section 3.5, we compare an approximation model for queueing networks with simulation; in Section 3.6, we discuss the results and implications of our work.

3.2 Preliminaries

3.2.1 Phase-type distributions

The phase-type distribution is composed of a finite sum or a finite mixture of exponentially distributed components, or a combination of both. When the interarrival time and service time follow the exponential distribution, we call it a Markovian queueing system and solve the system using Markov processes. In addition, if we approximate the interarrival time and service time as a corresponding phase-type distribution, we can analyze the system using the Markov property.

The continuous phase-type distribution used in this paper was defined by Neuts (1981). The following relationships are developed in that book and are repeated here for clarity of exposition. A phase-type distribution is the distribution of time to reach absorbing state $m+1$ in a finite Markov process having infinitesimal generator

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

where the $m \times m$ matrix T satisfies $T_{ii} < 0$, for $1 \le i \le m$, and $T_{ij} \ge 0$, for $i \ne j$.

The equation $\mathbf{Te} + \mathbf{T^0} = \mathbf{0}$ is satisfied, and the initial probability vector of Q is $(\boldsymbol{\alpha},\ \alpha_{m+1})$, with $\alpha\mathbf{e} + \alpha_{m+1} = 1$. Here, $\mathbf{0}$ is a row vector of zeros and $\mathbf{e}$ is a column vector of ones. The pair $(\boldsymbol{\alpha}, \mathbf{T})$ specifies the phase-type representation.

Given initial probability vector $\boldsymbol{\alpha}$, the cumulative distribution function of the time to reach state $m+1$ is

$$F(x) = 1 - \boldsymbol{\alpha} e^{\mathbf{Tx}}\mathbf{e}. \tag{3.1}$$

The density function is

$$f(x) = \alpha \mathbf{e}^{\mathbf{Tx}} T^0 = \alpha \mathbf{e}^{\mathbf{Tx}}(-\mathbf{T})\mathbf{e},$$

and the moments are defined by

$$E(X^k) = (-1)^k k! \boldsymbol{\alpha} \mathbf{T}^{-\mathbf{k}} \mathbf{e}. \tag{3.2}$$

3.2.2 Fitting distribution

In order to analyze our system as a Markovian queueing system, we approximate a general service time as a corresponding phase-type distribution. Mapping general distributions to phase-type distributions has been studied by Sauer and Chandy (1975), Marie (1980), Tijms (1994), You et al. (2002) and Osogami and Harchol-Balter (2003). We follow the matching two moments method of Tijms (1994) and You et al. (2002):

A general distribution is fitted as one of three phase-type distributions: Erlang-k, balanced hyper-exponential and the exponential distribution, based on the squared coefficient of variation $C^2 = \sigma^2/\mu^2$. If $C^2 < 1$, a general distribution is approximated as an Erlang distribution, Erlang (k, λ), of order k. The density function is given by

$$f(x) = \lambda^k \frac{x^{k-1}}{(k-1)!} e^{-\lambda x}, x \geq 0.$$

The shape parameter k is computed by $\lceil \frac{1}{C^2} \rceil$ and $\lambda = k/E[X]$, where $E[X]$ is the mean of the general distribution.

If $C^2 > 1$, a general distribution is converted to a balanced hyper-exponential distribution, HE_2. The density function of HE_2 is described by

$$f(x) = p_1 \lambda_1 e^{-\lambda_1 x} + p_2 \lambda_2 e^{-\lambda_2 x}, x \geq 0,$$

where $p_1 = \frac{1}{2}\left(1 + \sqrt{\frac{C^2-1}{C^2+1}}\right)$, $p_2 = 1 - p_1$, $\lambda_1 = 2p_1/E[X]$ and $\lambda_2 = 2p_2/E[X]$.

If $C^2 = 1$, we use the exponential distribution, with density function,

$$f(x) = \lambda e^{-\lambda x}, x \geq 0,$$

where $\lambda = 1/E[X]$.

3.3 Exponential service times

In this section, we assume all servers in a workstation are identical and have the same exponential distribution of processing times. Whitt (1999) suggested that the waiting time distribution in such a system follows an Erlang-$(k+1)$ distribution with mean $1/c\mu$ when the system has c servers and there are k customers ahead. For now, we are interested in the complete sojourn time distribution (waiting plus processing), so we develop an approximation model for multi-server queueing systems with exponential service times.

3.3.1 Approximation model

The state dependent sojourn time distribution does not rely on the interarrival time distribution because the waiting time distribution is determined only by the number of servers, the number of orders ahead and the service rate. Also, we do not need to consider the remaining service time for exponential service, due to the memoryless property. Every order arriving to the system has Erlang waiting time in queue and is processed in exponential service time.

We approximate the sojourn time distribution of an order in the system based on the convolution property of the phase-type distribution, because the waiting times and service time are phase-type. Neuts (1981) introduced the convolution property of the phase-type distribution. If $H(\cdot)$ and $I(\cdot)$ are both continuous phase-type distributions with representations $(\boldsymbol{\beta}, T)$ and $(\boldsymbol{\gamma}, S)$ of orders m and n, then the convolution of two distributions, $H * I(\cdot)$ is also a phase-type distribution with representation $(\boldsymbol{\alpha}, Q)$ and the infinitesimal generator and the initial probability vector are given by

$$Q = \begin{bmatrix} T & T^0\boldsymbol{\gamma} \\ 0 & S \end{bmatrix},$$

$$\boldsymbol{\alpha} = [\boldsymbol{\beta}, \beta_{m+1}\boldsymbol{\gamma}].$$

We construct the initial probability vector and the infinitesimal generator of the system based on this result. The infinitesimal generator is composed of the waiting time representation $(\boldsymbol{\alpha}, W)$ and service time representation $(\boldsymbol{\beta}, S)$. The sizes of the matrix W and vector $\boldsymbol{\alpha}$ are determined by the number of orders ahead in the queue. For example, if there are k orders ahead, the size of the infinitesimal generator and initial probability vector of waiting time are given by $W_{(k+1)\times(k+1)}$ and $\alpha_{1\times(k+1)}$. Thus, the phase-type representation of the sojourn time distribution is determined by

$$K = \begin{bmatrix} W & W^0\beta \\ 0 & S \end{bmatrix},$$

$$\boldsymbol{\gamma} = [1, 0, \cdots, 0].$$

We can generate the cumulative distribution function (CDF) of the system based on the phase-type representation (γ, K)

$$F(t) = P(T \leq t) = 1 - \gamma e^{Kt}\mathbf{e}, (t \geq 0).$$

The probability density function (PDF) is given by

$$f(t) = \gamma e^{Kt}K^0 = \gamma e^{Kt}(-K)\mathbf{e}, (t \geq 0).$$

3.3.2 Numerical results

To test the approximation model, we consider two factors—the number of servers c and the number of orders ahead k. We use

- c ={ 2, 3, 5, 10, 20, 50, 100, 200}, and
- k ={5, 10, 20} for $c \leq 100$, {40, 60, 80} for $c = 200$.

Table 3.1: A comparison of mean sojourn times for exponential service times.

Servers		2			3			5	
Orders ahead	5	10	20	5	10	20	5	10	20
Approximation	20.00	32.50	57.28	15.00	23.33	40.00	11.00	16.00	26.00
% difference	0.08	0.20	-0.01	-0.63	-0.56	-0.44	0.52	0.59	0.58
Simulation	19.98	32.44	57.28	15.09	23.46	40.18	10.94	15.91	25.85
Servers		10			20			30	
Orders ahead	5	10	20	5	10	20	5	10	20
Approximation	8.00	10.50	15.50	6.50	7.75	10.25	5.99	6.81	8.47
% difference	0.17	0.31	0.36	-0.04	-0.01	-0.01	0.04	0.08	0.17
Simulation	7.99	10.47	15.44	6.50	7.75	10.25	5.98	6.80	8.46
Servers		50			100			200	
Orders ahead	5	10	20	5	10	20	40	60	80
Approximation	5.61	6.09	7.10	5.29	5.56	6.04	6.01	6.54	7.01
% difference	0.34	-0.04	-0.06	-0.05	0.31	-0.18	-0.18	0.30	-0.14
Simulation	5.59	6.10	7.10	5.30	5.55	6.05	6.02	6.52	7.02

We test possible combinations of these two factors under the same capacity ($E[T_s] = 5$) and compare with the results of a simulation built in Arena 10.0. We assume processing time in the simulation model is exponentially distributed. Table 3.1 shows the mean sojourn time results from the approximation model and the simulation model. Differences in the means are all less than 1 percent.

Figures 3.1–3.3 show the PDF and CDF of the approximation and simulation results. As with the mean sojourn time, we see nice agreement. To make a more scientific statement, we chose three cases and checked the agreement using the Anderson-Darling (A-D) test. We find that all statistics of the tests are less than the critical value with $\alpha = 5\%$ (see Table 3.2). The percent differences of the 90^{th} and 95^{th} percentile between the two models are also small.

3.4 General service times

In addition to the exponential service time case, Whitt (1999) developed an approximation model for general service times in single stage systems. For a large number of waiting

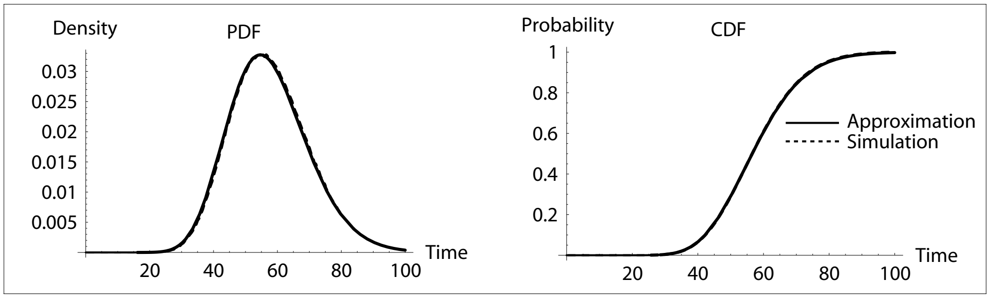


Figure 3.1: Comparison of the PDF and CDF of the approximation model and the simulation model (2 servers and 20 orders ahead).

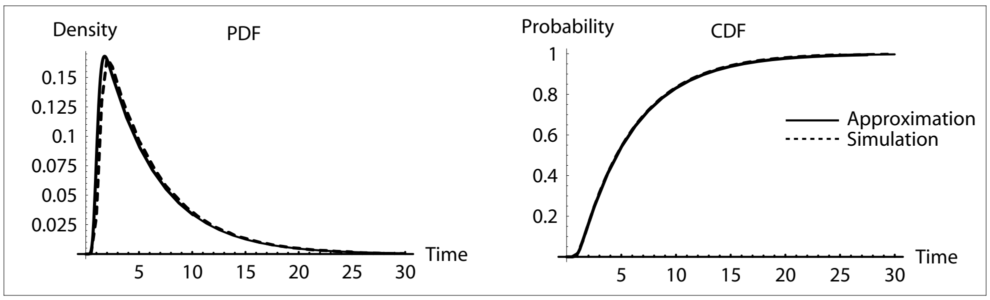


Figure 3.2: Comparison of the PDF and CDF of the approximation model and the simulation model (50 servers and 10 orders ahead).

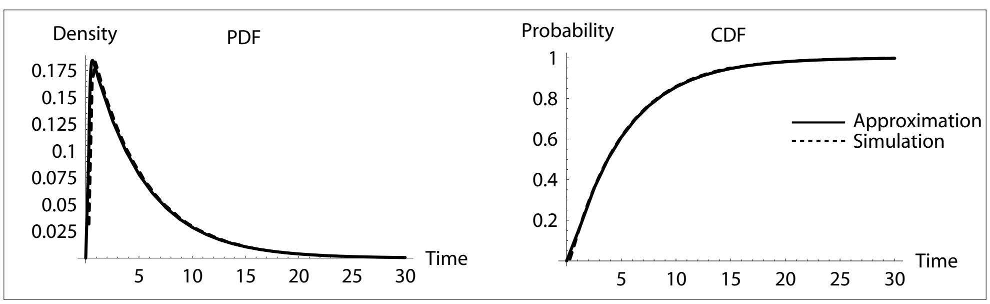


Figure 3.3: Comparison of the PDF and CDF of the approximation model and the simulation model (100 servers and 5 orders ahead).

Table 3.2: Anderson-Darling tests for the exponential service time of single-stage queue.

Servers	Orders ahead	A-D	$\alpha = 5\%$	Decision	% difference	
					90^{th}	95^{th}
2	20	1.405		accept	0.66	2.44
50	10	1.329	2.492	accept	3.10	0.31
100	5	0.631		accept	0.82	0.17

customers k, a Normal approximation can be used. The mean waiting time is given by

$$E[W] \approx \frac{k+1}{\mu c}\left(1+\frac{1}{2c}\right),$$

and the full distribution of waiting time is described by a Normal approximation. Whitt states that his approximation is appropriate only when the number of servers is substantially larger than the number of customers ahead, as is commonly the case in call centers.

In contrast to Whitt (1999), we approximate the sojourn time distribution using phase-type distributions. The procedure is similar to the exponential service times case, except for considering the remaining service times of the orders in service.

The procedure is:

1. Approximate the service time distribution as a corresponding phase-type distribution $(\boldsymbol{\beta}, S)$ based on the C_s^2.

2. Approximate the first waiting time distribution as a corresponding phase-type distribution $(\boldsymbol{\alpha}_1, W_1)$ based on the Markov process.

3. Approximate the second waiting time distribution as a corresponding phase-type distribution $(\boldsymbol{\alpha}_2, W_2)$ based on the Markov process and the first initial probability vector $\boldsymbol{\alpha}_1$.

4. Approximate successive waiting time distributions as corresponding phase-type distributions $(\boldsymbol{\alpha}_i, W_i)$, $i \geq 3$, according to the same procedure.

5. Generate the initial probability vector and infinitesimal generator $(\boldsymbol{\gamma}, K)$ for the system using the convolution property of the phase-type distribution.

6. Solve $F(t) = P(T \leq t) = 1 - \boldsymbol{\gamma} e^{Kt}\mathbf{e}, (t \geq 0)$ to obtain the CDF of the sojourn time distribution.

To analyze a system using continuous time Markov chains, we need to approximate a general service time distribution as a phase-type distribution. We generate the phase-type representation of service time $(\boldsymbol{\beta}, S)$ using the method in Section 3.2.2. The number of phases and the transition rate are determined by the SCV and the number of servers.

If a server is idle, an arriving order enters service immediately, and we do not need to consider waiting time to calculate the sojourn time—the sojourn time is equal to the service time. If there is no idle server and the arriving order finds k orders ahead in the queue, its sojourn time consists of three times — the waiting time for the first order to depart, the waiting time for the next k orders to depart, and the service time. In this condition, if one of the servers finishes its order, the order can go one step forward and then all servers are working immediately. So the order waits $k+1$ "sub-waiting times" to enter service and departs the system after receiving service. Hereafter, we refer to a "sub-waiting time" as an *epoch*.

3.4.1 Approximating the first epoch

Each epoch is the time an arriving order spends in queue until one of the servers finishes its order. That is, an epoch starts when all servers are busy (*all-busy*) and ends when one of the servers finishes its order (*partial-busy*).

To estimate the distribution of waiting time, we introduce a continuous time Markov process $\{N(t); t \geq 0\}$ with some absorbing states to model the all-busy period, where the system-state of the Markov chain is the number of servers in each phase. Asmussen and O'Cinneide (1998) showed that the waiting time distribution in a GI/PH/c queue is phase-type, and that the number of phases is

$$\begin{pmatrix} m+c-1 \\ c \end{pmatrix},$$

where c is the number of servers and m is the number of phases of each server.

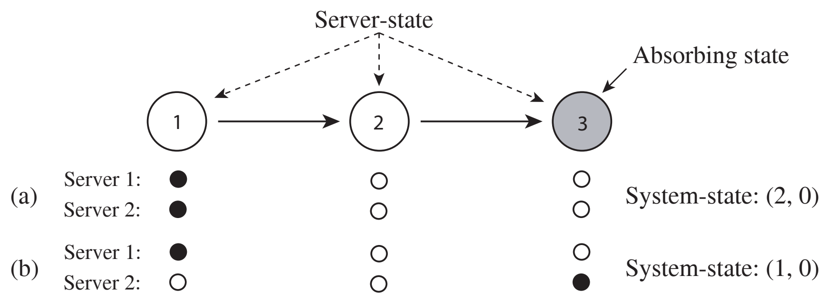


Figure 3.4: System-state and server-state.

Suppose there are c homogeneous servers, each with m phases, and an order finds k orders ahead in the queue. There are $m+1$ server-states, and each server can be in one of those states. A order arrives to find k orders ahead and servers in one of m states (none are idle). We describe the system-state as an m-vector of server-states. The i^{th} element in the vector records the number of servers in state i.

Example 3.1 An order arrives to a system with 2 homogeneous servers and finds 3 orders ahead in the queue. We wish to estimate the sojourn time distribution of this order given $E[T_s] = 2$ and $C_s^2 = 0.8$.

First, we approximate the general service time distribution as a corresponding distribution using the fitting distribution method in Section 3.2.2. We approximate with an Erlang $(2, 1)$ because $m = \left\lceil \frac{1}{C_s^2} \right\rceil = 2,\ \mu = \frac{m}{E[T_s]} = 1$.

Figure 3.4 shows two examples of the relationship between server-state and system-state: (a) shows two servers working in the first server-state, and this is system-state (2, 0); (b) shows server 1 working in the first server-state and server 2 has finished its order, and this is system-state (1, 0). There are 5 system-states: (2, 0), (1, 1), (0, 2), (1, 0), (0, 1). We call (2, 0), (1, 1), (0, 2) *all-busy* states and (1, 0), (0, 1) *partial-busy* states. The time between partial-busy states is an epoch, except for the first epoch which begins upon arrival to the system and therefore to an all-busy state (otherwise a server was empty and the order entered service immediately).

We are interested in the distribution of the first epoch, which is the time until the process enters the first partial-busy state, given the process started from an all-busy state. The infinitesimal generator W_1 of the first epoch includes state changes from the all-busy states to partial-busy states. In Example 3.1 ,

$$W_1 = \begin{bmatrix} -2 & 2 & 0 \\ 0 & -2 & 1 \\ 0 & 0 & -2 \end{bmatrix}.$$

We also need to compute the initial probability vector $\boldsymbol{\alpha}_1$ of the the first epoch, for which we use the stationary distribution of the all-busy states, because an arriving order finds the system in one of these all-busy states. In our approximation model, if the system reaches a partial-busy state, the system-state is changed immediately to an all-busy state because there is another order in the queue. Thus we should consider two kinds of state changes for $\boldsymbol{\alpha}_1$. One is state changes from the all-busy states to partial-busy states, and the other is state changes from partial-busy states to all-busy states. If we sequence the states such that the all-busy states precede the partial-busy states, the former state changes have infinitesimal generator W_1, which has zero value for the lower triangle below the diagonal. The latter state changes have transition rate matrix H, which has the same size as W_1 and has zero value for the upper triangle and diagonal. The transition rate matrix $Q = W_1 + H$ includes all possible state changes when an order arrives at a certain position in queue. In Example 3.1, H and Q are

$$H = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix}, Q = \begin{bmatrix} -2 & 2 & 0 \\ 1 & -2 & 1 \\ 0 & 2 & -2 \end{bmatrix}.$$

Now we can compute $\boldsymbol{\alpha}_1$ using the transition rate matrix Q. The transition probability matrix function $P(t)$ of the continuous time Markov chain is given by

$$P(t) = e^{Qt} = I + \sum_{n=1}^{\infty} \frac{1}{n!} Q^n t^n,$$

where I is the identity matrix. The stationary distribution is given by

$$\pi_j = \sum_i \pi_i P_{ij}(t).$$

In Example 3.1, the initial probability vector $\boldsymbol{\alpha}_1$ of the all-busy states in the first epoch is

$$\boldsymbol{\alpha}_1 = \{\pi_j\} = (\pi_{20}, \pi_{11}, \pi_{02}) = (0.25, 0.5, 0.25).$$

3.4.2 Approximating the remaining epochs

The infinitesimal generators W_k of the following epochs are the same as the infinitesimal generator of the first epoch, because the state changes from the all-busy to partial-busy states are the same. But the initial probability vectors $\boldsymbol{\alpha}_k$ of the following epochs are not the same. Rather, they must be derived successively from the initial probability in the previous epoch, because these affect which partial-busy state was reached. As we mentioned above, if the system reaches a partial-busy state, the system state is changed immediately to an all-busy state. This means the initial probability vectors $\boldsymbol{\alpha}_k$ of the all-busy states in epoch k come directly from the stationary distribution $\boldsymbol{\beta}_{k-1}$ of the partial-busy states in the former epoch $k-1$.

Now we introduce a method to compute the stationary distribution $\boldsymbol{\beta}_{k-1}$ of partial-busy states in epoch $k-1$. Every epoch starts from one of the all-busy states and ends in one of the partial-busy states, so if we know the stationary distribution of the all-busy states and the absorbing probability from the all-busy states to partial-busy states, then the stationary probability of the partial-busy state j

$$\beta_j = \sum_{all-busy \;\; i} \pi_i u_{ij},$$

where π_i is stationary probability of all-busy state i and u_{ij} is the absorbing probability from all-busy state i to partial-busy state j. From the Chapman-Kolmogorov equations,

$$u_{ij} = \sum_{all-busy \;\; h} P_{ih} u_{hj}.$$

Finally, we get the elements of $\boldsymbol{\alpha}_k$ directly from the corresponding elements in $\boldsymbol{\beta}_{k-1}$. In Example 3.1, the initial probability vector $\boldsymbol{\alpha}_2$ of the all-busy states in the second epoch is

$$\boldsymbol{\alpha}_2 = (\beta_{10}, \beta_{01}, 0) = (0.375, 0.625, 0).$$

Note that elements of the initial probability vector $\boldsymbol{\alpha}_k$ converge to the same value as the epoch k increases.

3.4.3 Approximating the sojourn time distribution of the system

The sojourn time of an order with $c+k$ orders ahead in the system has $k+1$ different sub-waiting times and one service time. So, the state-dependent sojourn time distribution is given by the convolution of these $k+2$ distributions.

We construct the initial probability vector and the infinitesimal generator of the queueing system based on the work of Neuts (1981). The infinitesimal generator is composed of the waiting time representation $(\boldsymbol{\alpha}_k, W)$ of each order's waiting time in the system and the service time representation $(\boldsymbol{\beta}, S)$. If there are k orders ahead in the queue, the infinitesimal generator and initial probability vector are given by

$$K = \begin{bmatrix} W & W^0\alpha_2 & 0 & \cdots & 0 & 0 \\ 0 & W & W^0\alpha_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & W & W^0\alpha_{k+1} \\ 0 & 0 & 0 & \cdots & 0 & S \end{bmatrix},$$

$$\boldsymbol{\gamma} = [1, 0, \cdots, 0].$$

The CDF and PDF are given by

$$F(t) = P(T \leq t) = 1 - \gamma e^{Kt}\mathbf{e}, (t \geq 0),$$

$$f(t) = \gamma e^{Kt} K^0 = \gamma e^{Kt}(-K)\mathbf{e}, (t \geq 0).$$

3.4.4 Normal Approximation Model

One drawback of our approximation model is computation time for very large systems. For example, models with up to 30 servers and fewer than, say, 20 orders ahead, generally solve within 1 minute; problems with 50 servers can take 3 minutes; problems with 100 servers can take 8.5 hours and problems with 200 servers, more than a day. To address these larger problems, we introduce a second approximation called the Normal Approximation Model (NAM).

The NAM uses the same phase-type representation of waiting and processing times as the approximation we have already developed, except that the CDF and PDF are not calculated with the matrix exponential expression in Equation 3.1. Instead, we assume the final distribution is Normally distributed, and that we can compute the required first and second moments with Equation 3.2, which requires only a matrix power computation.

3.4.5 Numerical results

Consider the example in Section 3.3.2, but now with general service times and $C^2 = 0.5$. We compare our results with Whitt's method and with simulation in Table 3.3. For the comparison, we add the mean processing time to the waiting time of Whitt's results because he computes only mean waiting time. In the simulation model, we used the Gamma distribution for the service time because it is often an appropriate model for task completion time (Law and Kelton, 2000).

Table 3.3 shows the mean sojourn time results from the approximation model, Whitt's model and the simulation model. As expected, Whitt's model shows good results when the number of servers is high (in this case, greater than 30). With a few exceptions, the approximation model appears to perform well over a wide range of problem instances. We should note however, that the largest problems in this table are very difficult to compute using our approximation model. For example, a problem with 100 servers and 20 orders ahead takes 8.3 hours, and 200 server problems take about 1 day. We record computation time using a 2.4 GHz Intel Core 2 Duo processer, and most of the problems except for those mentioned above are computed in a few seconds. Therefore, one might view Whitt's method and ours as complements—ours performs well for small and medium sized systems, his for large.

Figures 3.5–3.7 compare the PDF and CDF of the approximation model with the simulation model under different numbers of servers and orders ahead in the queue. We use the Anderson-Darling (A-D) test to check the agreement between two distributions. As shown in Table 3.4, all test statistics are significant with $\alpha = 5\%$.

3.4.6 Testing the Normal Approximation Model

Table 3.5 shows the percent differences of percentiles between the NAM and the simulation when $c = 2, 20$ and 100. We see that differences at the 95^{th} percentile are greater than 5 percent for cases $k < c$, where k is the number of orders ahead and there are c

Table 3.3: A comparison of mean sojourn times for general service times.

Servers		2			3			5	
Orders ahead	5	10	20	5	10	20	5	10	20
Whitt	18.75	34.38	65.63	11.67	21.39	40.83	11.60	17.10	28.10
% difference	-4.95	6.51	15.65	-20.99	-7.06	5.21	5.23	6.43	8.27
Approximation	19.38	31.88	56.87	14.71	23.20	40.17	10.97	16.24	26.82
% difference	-1.79	-1.24	0.21	-0.39	0.79	3.51	-0.53	1.06	3.34
Simulation	19.73	32.27	56.75	14.77	23.01	38.81	11.02	16.07	25.95
Servers		10			20			30	
Orders ahead	5	10	20	5	10	20	5	10	20
Whitt	8.15	10.78	16.03	6.54	7.82	10.38	6.02	6.86	8.56
% difference	2.25	2.81	3.42	0.81	1.20	1.49	0.68	0.93	1.32
Approximation	8.02	10.79	16.45	6.48	7.81	10.67	5.99	6.84	8.63
% difference	0.62	2.99	6.19	-0.04	1.04	4.26	0.15	0.53	2.19
Simulation	7.96	10.48	15.43	6.46	7.70	10.26	5.97	6.79	8.43
Servers		50			100			200	
Orders ahead	5	10	20	5	10	20	40	60	80
Whitt	5.61	6.11	7.12	5.30	5.55	6.06	6.03	6.53	7.03
% difference	0.61	0.52	0.82	0.21	0.26	0.27	0.16	0.22	0.25
Approximation	5.59	6.09	7.12	5.30	5.55	6.04	6.02	6.51	7.02
% difference	0.39	0.21	0.73	0.19	0.19	0.09	-0.01	-0.01	0.05
Simulation	5.56	5.96	7.06	5.28	5.53	6.03	5.99	6.49	6.99

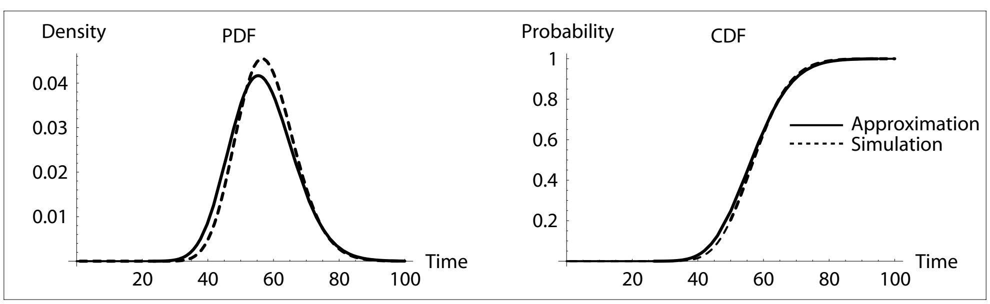


Figure 3.5: Comparison of the PDF and CDF of the approximation model and the simulation model (2 servers and 20 orders ahead).

Table 3.4: Anderson-Darling tests for general service time of the single-stage queue.

Servers	Orders ahead	A-D	$\alpha = 5\%$	Decision	% difference 90^{th}	% difference 95^{th}
2	20	1.755		accept	0.92	1.78
50	10	1.828	2.492	accept	1.56	0.51
100	5	2.327		accept	0.22	0.27

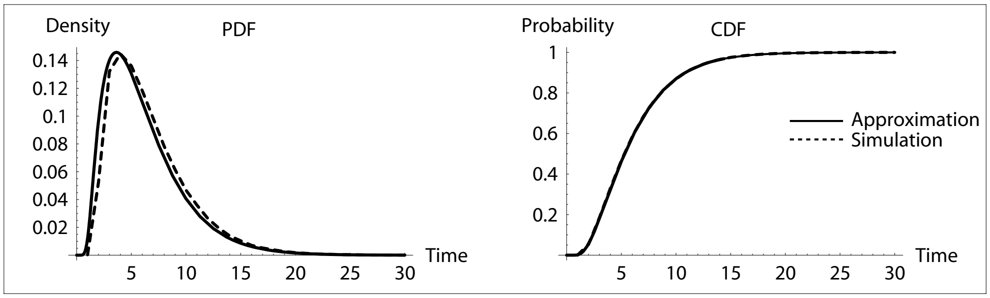


Figure 3.6: Comparison of the PDF and CDF of the approximation model and the simulation model (50 servers and 10 orders ahead).

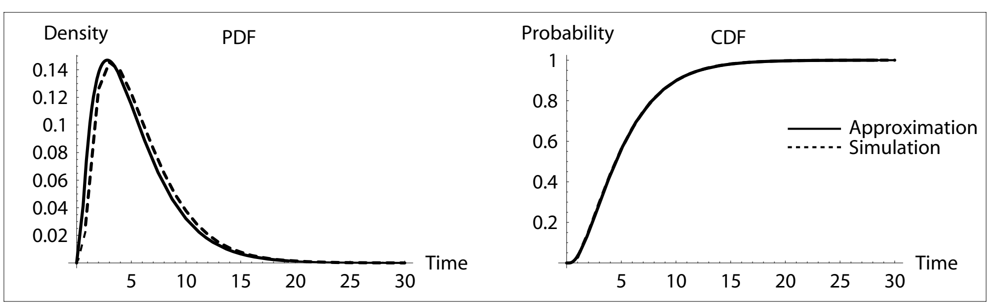


Figure 3.7: Comparison of the PDF and CDF of the approximation model and the simulation model (100 servers and 5 orders ahead).

servers, which suggests that the model does not perform well under these conditions. The explanation has two parts: First, the Normal approximation will perform well, in general, when (1) successive waiting times have the same distribution, and (2) there are many of them. Second, successive waiting times in our case are *not* identically distributed because of the initial set of remaining times (Section 3.4.2). However, successive waiting time distributions do converge as memory of the initial remaining times is "lost," which is reflected by successive initial probability vectors $\boldsymbol{\alpha}_k$ converging to the same set of values. Systems with fewer servers converge faster, so, in general, the NAM performs better when $k \gg c$.

Table 3.5: The percent differences of percentiles between NAM and Simulation for the single stage queue.

Servers		2			20			100	
Orders ahead	5	10	20	5	10	20	5	10	20
90^{th}	-1.69	0.46	1.48	-1.47	-0.53	2.53	-9.23	-1.89	-1.76
95^{th}	-2.73	0.24	2.34	-7.33	-5.84	-2.00	-14.06	-8.16	-7.74

3.5 Multi-Stage Systems

3.5.1 Approximation model

We extend our work to serial lines and acyclic queueing networks based on the single stage approximation model. Consider first a simple, 3-station serial line, and suppose an order is in queue at the first workstation. The order will experience three sojourn times, one at each queue, and the total sojourn time is their convolution. After each of the first two sojourn times, the status of the remaining queues changes, so we must estimate how many orders are in these queues when the order of interest arrives. After each sojourn time, the remaining queues change in two ways: orders arrive from the upstream workstation, and some orders are completed and depart. During each stage, then, we must estimate how many orders arrive to and depart from each remaining workstation (Figure 3.8).

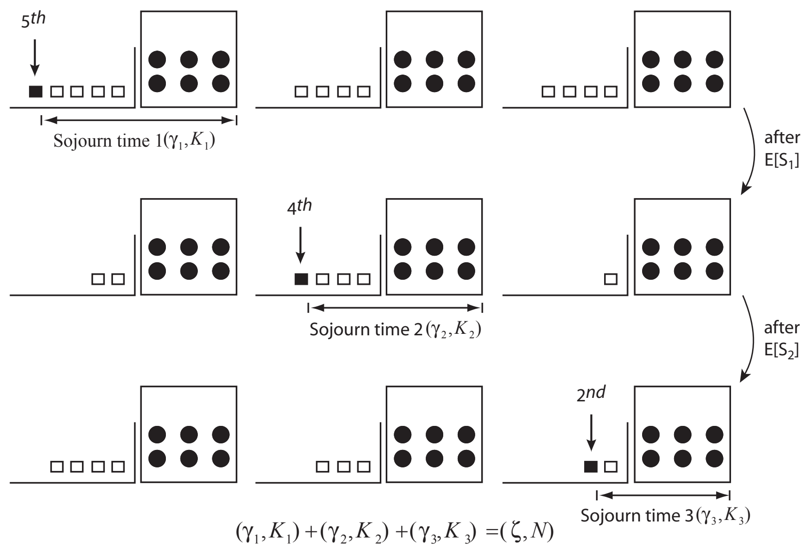


Figure 3.8: State-dependent queueing network model. Black discs represent workers; squares represent orders.

For the first workstation, we simply apply the single-stage model, giving us a sojourn time distribution and its mean $E[S_1]$. Now, let q_j^i be the estimated queue length of workstation j after the i^{th} sojourn time. Waiting time at the second workstation is based on the starting queue length q_2^0, plus arriving orders, minus departing orders. We assume that during $E[S_1]$, all $q_1^0 + c_1$ orders in front of the order of interest arrive to workstation 2. If all servers were busy during this time, then we would estimate the number processed at workstation 2 by $c_2 \times E[S_1]/E[T_2]$. However, it is possible that some servers could go idle during this time, so we correct for this by using a floor function, $\lfloor c_2 \times E[S_1]/E[T_2] \rfloor$. The expected number of orders ahead upon arrival to the second workstation becomes,

$$q_2^1 = \max(0, q_2^0 + (q_1^0 + c_1) - \lfloor c_2 \times E[S_1]/E[T_2] \rfloor).$$

During each stage of calculation, we must revise the number in queue at remaining workstations. For the third and following workstation, this leads to

$$q_j^i = \max\left(0, q_j^{i-1} + \left\lfloor c_{j-1} \times \frac{E[S_i]}{E[T_{j-1}]} \right\rfloor - \left\lfloor c_j \times \frac{E[S_i]}{E[T_j]} \right\rfloor\right), \text{ for } i = 1, \ldots, j-2.$$

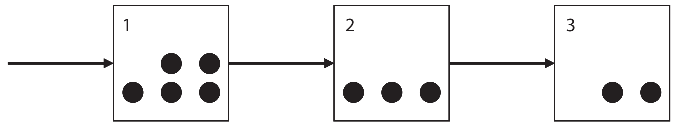


Figure 3.9: A serial line with 3 stations.

The final revision $(j-1)$ for workstation j is

$$q_j^{j-1} = \max\left(0, q_j^{j-2} + q_{j-1}^{j-2} + c_{j-1} - \left\lfloor c_j \times \frac{E[S_{j-1}]}{E[T_j]} \right\rfloor\right).$$

These approximations assume heavy traffic and are far from exact. However, they are sufficient for our purposes, as we are about to show.

For large serial systems, an alternative approach is to modify the NAM by "collapsing" the system into a single-stage queue. If we wish to know the state-dependent sojourn time distribution of an order in front of the first station in a serial line with 3 stations, we assume our system is a single stage queue with c_3 servers, and the number of orders ahead is $q_1 + c_1 + q_2 + c_2 + q_3$, assuming all servers are busy. Now we can generate the state-dependent sojourn time distribution using the single stage model directly.

3.5.2 Numerical results

We apply the approximation model to a serial line with 3 stations (Figure 3.9) and an acyclic queueing network with 4 stations (Figure 3.10). The black disc in each station represents an occupied server, and in the case of the acyclic queueing network, an order departing from the first station selects its follow on station with probability p. We test state-dependent sojourn time distributions for both systems when the order is located in front of the first station. We also test the acyclic queueing network when the order is located in front of the second station. The system information is described in Table 3.6.

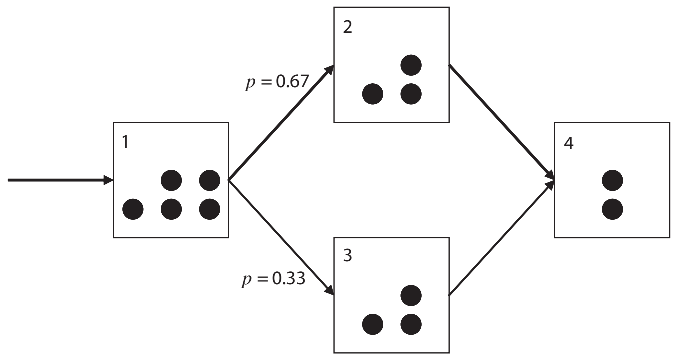


Figure 3.10: An acyclic queueing network with 4 stations. Black discs represent workers.

Table 3.6: The system information of the serial line and acyclic queueing network.

Serial line	$E[T]$	C^2	Servers	ρ
Interarrival	1.05	0.7		
Station 1	5	0.8	5	0.95
Station 2	3	0.8	3	0.95
Station 3	2	0.8	2	0.95
Acyclic queueing network	$E[T]$	C^2	Servers	ρ
Interarrival	1.05	0.7		
Station 1	5	0.8	5	0.95
Station 2	4.47	0.8	3	0.95
Station 3	9.09	0.8	3	0.95
Station 4	2	0.8	2	0.95

Table 3.7: The comparison of the mean sojourn time of the serial line.

Orders in queues	Approximation	% difference	NAM	% difference	Simulation
11-12-13	47.98	0.11	46.75	-2.46	47.93
6-12-13	42.70	-0.12	41.75	-2.34	42.75
16-12-13	53.27	0.50	51.75	-2.37	53.01
11-20-13	55.13	-1.62	54.75	-2.30	56.04
17-12-17	58.33	1.15	56.75	-1.60	57.67
7-20-4	41.90	-2.28	41.75	-2.64	42.88
31-22-29	93.32	0.30	92.75	-0.31	93.04

We compare the mean sojourn time and the distribution among the approximation, the NAM, and the simulation model under different numbers of orders in each station. We assume service times and interarrival times in the simulation model are Gamma distributed. To estimate a sojourn time distribution in simulation for problems given in Table 3.7, we collect sojourn times only for orders seeing the required state conditions upon arrival. When an order arrives to the system, we check the number in queue at each workstation. If this vector of values corresponds to the state of interest problem condition, we record the sojourn time; otherwise we do not. This "brute force" method alleviates the problem of estimating upon arrival the remaining times of orders in process. However, because of occurrences of a particular state are rare, these simulations can take a very long time to run. For example, the first problem in Table 3.7 (11-12-13) takes 2 days for 50 runs, with 40 million (simulated) hours in each run.

Table 3.7 shows the results. In the table, the column " Orders in queues" represents the number of orders in the respective queues. The approximation model is extremely close to the simulation results for nearly every case. The NAM is acceptable, but not quite as good. We believe the NAM underestimates the mean (notice the negative differences) because it does not account for potential starving situations.

Figures 3.11 and 3.12 compare the PDF and CDF of the approximation model with the simulation model under different numbers of servers and orders ahead in the queue. Again, the Anderson-Darling (A-D) tests reveal that there is no significant difference between the

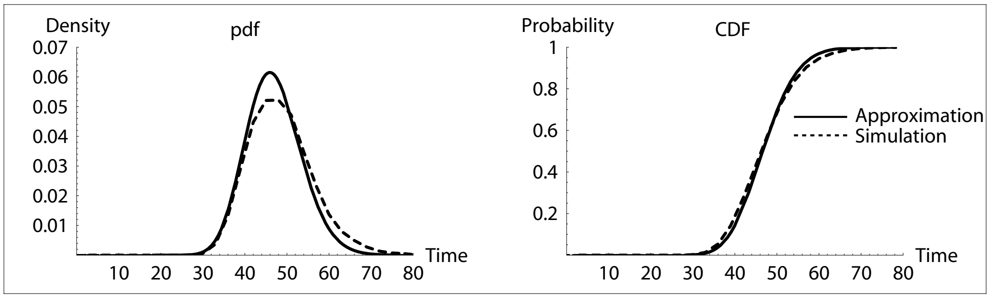


Figure 3.11: Comparison of the PDF and CDF of the serial line (11-12-13).

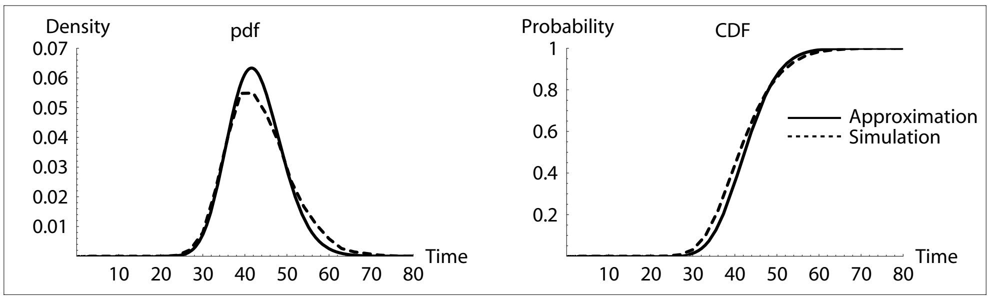


Figure 3.12: Comparison of the PDF and CDF of the serial line (6-12-13).

simulation and approximation results. However distributions generated by the NAM do not exactly fit the simulation (Table 3.9). Most of the percent differences are greater than 5%, except when the numbers of orders at each station are 31, 22 and 29. This suggests that the NAM works well only when the total orders ahead is significantly greater than the number of servers at the last station, because of reduced chances of starving.

Table 3.10 shows the mean sojourn time of the approximation model and simulation for an acyclic queueing network. We test two cases: the order is in front of the first station

Table 3.8: Anderson-Darling tests for the serial line.

Orders in queues	A-D	$\alpha = 5\%$	Decision	% difference	
				90^{th}	95^{th}
11-12-13	2.22	2.492	accept	-3.08	-4.13
6-12-13	1.79		accept	-1.61	-2.52
16-12-13	0.86		accept	-2.29	-2.95
17-12-17	1.85		accept	-0.84	-2.14

Table 3.9: The percent differences of percentiles between the NAM and simulation for the serial line.

Orders in queues	Total orders ahead	90^{th}	95^{th}
11-12-13	44	-5.24	-7.83
6-12-13	39	-6.64	-9.24
16-12-13	49	-5.28	-7.16
17-12-17	54	-4.59	-6.82
31-22-29	90	-2.43	-3.98

Table 3.10: The comparison of the mean sojourn time of the acyclic queueing network.

Order location	Orders in queues	Approximation	% difference	Simulation
The first station	11-12-8-13	60.87	1.86%	59.77
	6-4-16-13	55.59	0.76%	55.18
	16-12-15-13	72.26	0.84%	71.66
	11-20-12-13	73.10	0.73%	72.57
	17-12-7-17	70.54	2.78%	68.64
The second station	11-12-8-13	39.56	-1.28%	40.07
	6-8-12-13	33.48	-1.76%	34.08
	16-12-15-13	39.56	-1.81%	40.29
	11-20-12-13	51.72	-0.53%	51.99
	17-12-7-17	43.56	2.25%	42.60

and in front of the second station. The approximation model appears to work well for both cases.

Figures 3.13–3.16 compare the PDF and CDF of the approximation model with the simulation model under different locations of the order of interest. Notice the unusual distribution in Figure 3.14. This is caused by the mixture of the distributions of the two potential serial line paths (path 1: 1-2-4 and path 2: 1-3-4) because an arriving can take either path to depart the system. If the difference between the two mean sojourn times is large ($E[S]$ of the two paths are 44.44 and 75.06 hours), the sojourn time distribution has two peaks, as in Figure 3.14. Notice that our model reflects this because we estimate the sojourn time distribution for a random order by approximating the CDF of all possible "serial lines" (paths in the network) and mixing those CDFs according to the probabilities of taking those

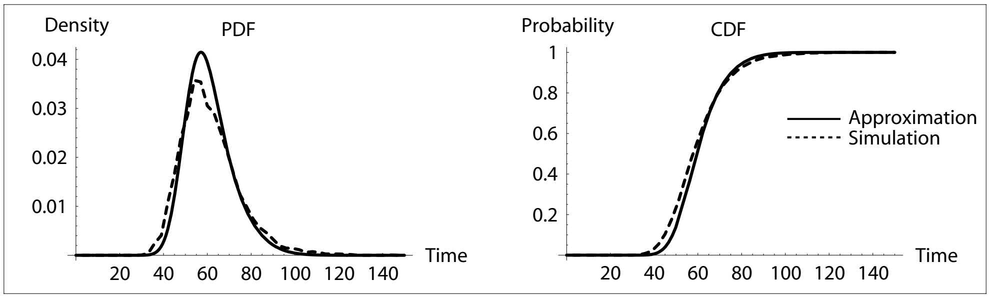


Figure 3.13: Comparison of the PDF and CDF of the acyclic queueing network (The first station, 11-12-8-13).

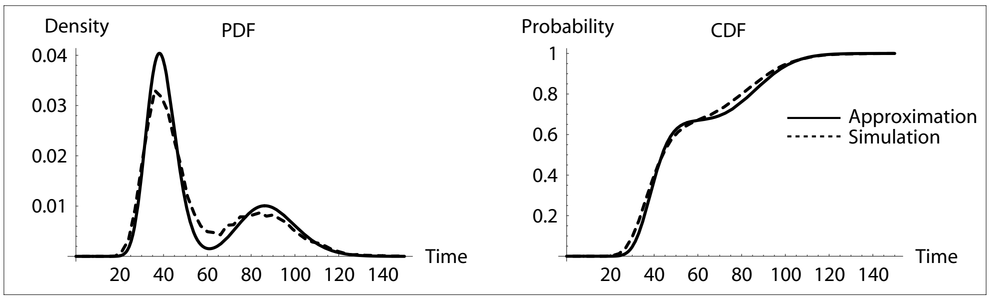


Figure 3.14: Comparison of the PDF and CDF of the acyclic queueing network (The first station, 6-4-16-13).

paths. We use the Anderson-Darling (A-D) test to check the agreement between the two distributions. As shown in Table 3.11, all test statistics are significant with $\alpha = 5\%$.

3.6 Chapter conclusions

We have developed an approximation model for state-dependent sojourn time distributions of queueing systems with multiple servers, using the characteristics of the phase-type distribution. We develop an approximation model for single-stage queues with exponential and general service time distributions and an approximation model for the serial lines and acyclic queueing networks with general service time distributions.

The waiting time of the single-stage G/M/c follows the Erlang distribution and has only one phase-type representation. So, it is easy to generate the CDF of the state-dependent

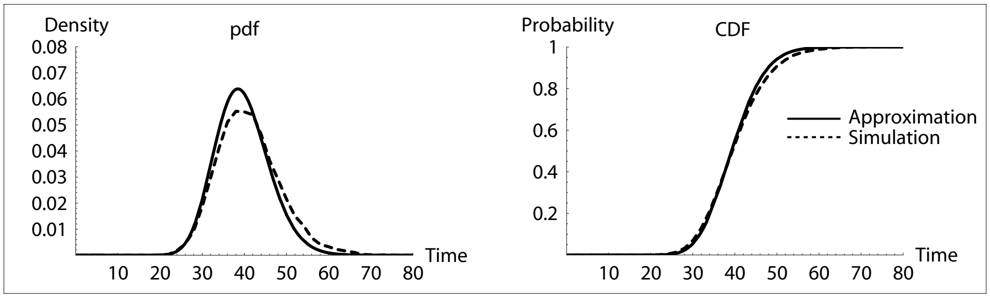


Figure 3.15: Comparison of the PDF and CDF of the acyclic queueing network (The second station, 11-12-8-13).

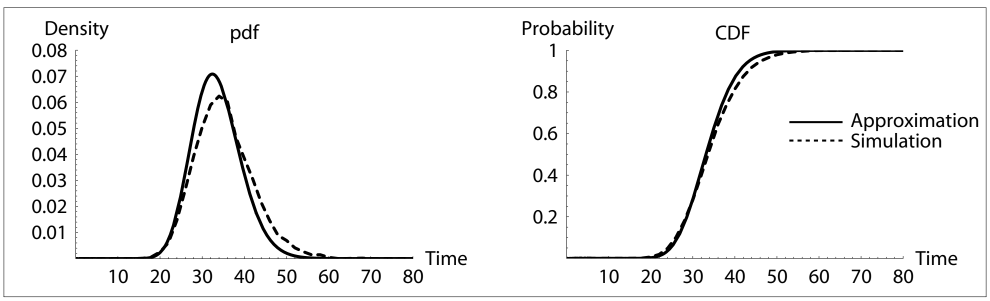


Figure 3.16: Comparison of the PDF and CDF of the acyclic queueing network (The second station, 6-8-12-13).

Table 3.11: Anderson-Darling tests for the acyclic queueing network.

Order location	Orders in queues	A-D	$\alpha = 5\%$	Decision	% difference	
					90^{th}	95^{th}
The first station	11-12-8-13	2.01	2.492	accept	-2.53	-4.40
	6-4-16-13	1.65		accept	1.75	-0.39
	16-12-15-13	1.21		accept	-0.91	-1.75
	11-20-12-13	2.31		accept	-3.72	-2.99
	17-12-7-17	1.74		accept	-1.98	-3.63
The second station	11-12-8-13	2.01	2.492	accept	-3.50	-4.15
	6-8-12-13	2.36		accept	-4.85	-6.61
	16-12-15-13	2.46		accept	-3.89	-5.43
	11-20-12-13	1.35		accept	-2.25	-2.65
	17-12-7-17	1.02		accept	-0.81	-0.98

sojourn time distribution, whereas the G/G/c single-stage queue has many phase-type representations of waiting time. For the waiting time distribution of the G/G/c queue, we introduce a Markov process $\{N(t), t \geq 0\}$ to describe the waiting time distribution considering remaining service time. The approximation model for a network was generated by convoluting sojourn time distributions in the routing of the order.

In contrast with Whitt's model, our approximation model performs well over a wide range of problem instances, whereas Whitt's model shows good results when the number of servers is high. However our single-stage approximation model is limited to problems with fewer than about 200 servers, because the computation time of our model increases as the number of servers and orders ahead increases. This problem is caused by the matrix exponential calculation.

The approximation allows us to estimate the probability that the state-dependent sojourn time will be less than a desired time t, $p_s = P(T < t)$. This result can be used in various ways based on the state of the system. For example, an order fulfillment system can do dynamic order promising to customers, at the point of order. During times of heavy loading, when the probability of an order making it onto a premium shipping mode is low, the retailer might declare to offer this mode to the customer.

Service systems could inform customers of their wait time with an estimated probability. For example, in an amusement park such as Disney world, we see the phrase "From this point, expected time is 1 hour" in a waiting line. Disney could develop this time at a higher probability of success and therefore "exceed expectations." Also, it could be used as a criterion for staff scheduling. If a service system has non-stationary arrival streams, we can schedule staff based on the sojourn time distribution of a peak time and non-peak time.

Chapter 4

Dynamic Worker Allocation Policies to Improve Service in an Order Fulfillment System

We propose several dynamic worker allocation policies to maximize service performance of an order fulfillment system. Our policies make use of the state-dependent sojourn time distribution for an order, which we compute with a model based on phase-type distributions. Our approach differs from other work on dynamic worker allocation in that we focus on service performance as perceived by the customer, instead of traditional system performance measures such as cycle time, throughput, and WIP. Our results suggest that order fulfillment systems can significantly improve their service performance by moving the right number of workers to the right place, at the right time.

4.1 Introduction

The most important function for most distribution centers is order fulfillment, which is the demand-side activity that satisfies customer requests. In many distribution centers, there are three primary activities: picking, packing, and shipping. Orders often arrive to such systems throughout the day and are prepared for shipping through picking and packing processes by workers. Customers usually want to receive their orders as soon as possible. Thus, how companies respond to customer orders is a matter of primary concern for this system.

Several years ago, we noticed from data of a large distributor in San Diego that on average, 5 percent of all orders arrived at the shipping dock in the 30 minutes *following* the departure of the last truck each night. In this chapter we ask, could workers have been redeployed somehow, perhaps in the few hours before the truck departure, to "flush" the system of orders that were almost completed, thereby increasing the service performance of the system? For example, workers could move from picking to shipping in order to flush the system of nearly completed orders right before the truck departs. The problem, of course, is that this causes an imbalance of the system and reduces the throughput of the picking area temporarily, which has reverberating effects later. Whether or not the system can be gamed in this way is what we explore in this chapter.

Our goal is to maximize a performance metric called *Next Scheduled Departure* (NSD), which was introduced by Doerr and Gue (2006) to measure service performance in a distribution center. NSD records the percentage of orders arriving in the 24 hours between cutoff times on successive days that make it onto the truck in the current day. The cutoff time is established by managers in the distribution center, and any order arriving before this time is due in the current day. Doerr and Gue (2006) claim that an increase in the NSD means a direct increase in customer satisfaction. We might say that NSD is a system performance measure from the customer's perspective, whereas existing metrics such as WIP, cycle time and throughput are from the distributor's perspective.

To introduce the dynamics of a worker allocation policy, consider a naïve policy, which moves a fixed number of workers at a fixed time close to the deadline everyday. We model the order fulfillment system as a serial line of 3 workstations (picking, packing, shipping) with 10, 12, and 9 workers per workstation. We move 8 workers from picking to shipping one hour before the deadline everyday, in an attempt to flush the system of almost completed orders. On some days, the policy is effective, but on others it is not. Because the naïve policy ignores the state of the system when moving workers, it is possible that workers are moved from a busy work center (picking) to a work center with little or no work to be done (shipping). This is a double error: (1) busy workers are made idle after the shift and (2) orders in picking back up during the shift, thereby increasing their chances of being late the *next* day.

Simulation experiments on this simple policy led us to consider more complex, but intuitively appealing policies based on *sojourn time*. The idea is to consider the likelihood (probability) that an order in a system will make it on the next departing truck, if the allocation of workers does not change. When this probability drops below a threshold value, then some workers in picking should be moved so as to improve the probability of success for other, more promising orders.

We introduce several dynamic worker allocation policies designed to maximize customer satisfaction without disturbing system balance. In the next section, we describe previous research on dynamic worker allocation policies. Section 4.3 presents the meaning of the Next Scheduled Departure (NSD) metric in terms of the sojourn time distribution. We also define the probability of success (p_s), which is the crucial criterion for our worker allocation policies. In Section 4.4 and 4.5, we introduce and evaluate several policies. We offer conclusions in Section 4.6.

4.2 Literature Review

There is a rich literature on workers moving among tasks in a dynamic fashion in manufacturing and, to a lesser extent, warehousing. Research in dynamic worker allocation falls under several names, such as worksharing systems, worker cross-training or cross-utilization, collaborative and non-collaborative work systems, and agile workforces. The use of cross-trained workers to improve the performance of manufacturing systems has been investigated by several researchers. Askin and Chen (2006) categorize worksharing policies as two types: Dynamic assembly-Line Balancing (DLB) and Moving Worker Modules (MWM). MWM systems have fewer workers than machines, so workers share every machine and move everywhere within a zone, whereas DLB has the same number of machines and workers. DLB has fixed tasks and shared tasks; the former are assigned to a designated worker, and the latter can be carried out by either of an adjacent pair of workers. In this review, we classify the floating worker system studies by MWM or DLB based on the number of workers per machine, the extent of skill per worker, and the amount of WIP.

Toyota Sewn-products management System (TSS) and the Bucket Brigade System are the representative MWM (Askin and Chen, 2006). These systems have fewer workers than machines. In a TSS system, a worker moves downstream doing his or her job at every machine within a restricted zone until either the job is finished or the downstream worker takes over the job. Bischak (1996) studied a U-shaped manufacturing line under the TSS system. She developed a simulation model and showed that the throughput of moving worker modules is higher than the fixed worker system with and without the use of buffers. Zavadlav et al. (1996) also examined a U-shaped serial line under the TSS system using a Markov decision process and simulation. They compared three systems: fixed assignments, overlapping (shared) worker assignments, and free-floating worker assignments based on the same constraints. They showed that the free-floating system is the most efficient way to lower WIP.

Bartholdi and Eisenstein (1996) and Bartholdi et al. (2001) developed the bucket brigades protocol based on TSS. They showed that sequencing workers from slowest to fastest can make a balanced partition of work emerge without management intervention, and that it produces the highest possible production rate. McClain et al. (2000) showed that bucket brigades does not work well with random processing times and nearly-equal worker velocities, because these conditions can lead to idle time for workers when the bucket brigade rule is followed strictly. To increase system performance, they suggested relaxing bucket brigade's "wait" restriction and allowing small amounts of inventory.

In the DLB problem, Gel et al. (2007) developed a zone policy in CONWIP production systems with hierarchical cross-training. They suggest the "fixed-before-shared" policy, which states that a cross-trained worker should do his unique task before helping a fixed worker with a shared task. They showed that partial cross-training in hierarchical patterns may yield substantial benefit if cross-trained workers are significantly faster than static workers under the "fixed-before-shared" policy. Gel et al. (2002) studied factors affecting the opportunity for worksharing under the same conditions. They showed that the ability to preempt the shared task, granularity of the shared task, and less variability of the task times increases the system effectiveness.

The half full buffer (HFB) control policy, which provides enough work for the downstream worker not to starve and enough empty space for the upstream worker not to be blocked, was studied by Ostalaza et al. (1990), McClain et al. (2000), Gel et al. (2002), Chen and Askin (2006) and Askin and Chen (2006). Ostalaza et al. (1990) suggested an SPT/RSPT rule, and showed that the system performance increases when the number of shared tasks is controlled by the HFB control policy. McClain et al. (2000) considered zoned worksharing and bucket brigades with a variety of situations. Using simulation, they tested the HFB control policy to move flexible workers for shared tasks. Following this research, Gel et al. (2002) proposed a more delicated HFB control policy considering general processing times and different worker speeds. They also suggested the 50-50 work content heuristic rule that an upstream machine transfers the shared task to the downstream machine in the case where the ratio of work content at the upstream machine to the total work content in the system is greater than 50 percent. Askin and Chen (2006) and Chen and Askin (2006) confirmed again the effect of the HFB control policy and suggested another threshold heuristic rule, smallest R no starvation (SRNS), to calculate the threshold value R under CONWIP levels. They showed that cross-training can increase system effectiveness with a work content heuristic rule such as SRNS and HFB.

In contrast to existing research, we offer a dynamic worker allocation policy with the following distinctives: First, the purpose of most former research on worker allocation has been to maximize system throughput, to minimize cycle time, or to decrease WIP in a manufacturing system. Our objective is to improve customer service in an order fulfillment system, which we measure with the *Next Scheduled Departure* (NSD) performance metric (Doerr and Gue, 2006).

Second, most of the existing research is limited to small tandem lines, such as 2 or 3-station tandem lines with one or two servers in the system. Existing results show that dynamic capacity allocation between adjacent workstations works well based on those simple conditions. However, real systems may not be so simple: there may be many servers in each workstation and perhaps many workers need to move among workstations to improve system performance. For example, a typical distribution center can have dozens of workers in each functional area (picking, packing, shipping). Our policies are effective in these larger systems.

Third, most of the methods which have been used by existing research are Markov decision processes, heuristics, or simulation. We compute the probability that an order in the system will make it on the next departing truck without worker allocation based on its sojourn time distribution in a multi-server queueing network. When this probability drops below a threshold value, we move some workers from one workstation to another to improve the probability of success for more promising orders.

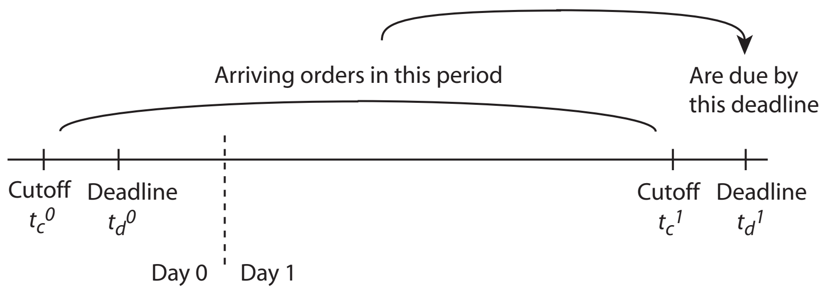


Figure 4.1: The relationship between the cutoff time and NSD (Doerr and Gue, 2006).

4.3 Preliminaries

Before designing worker our allocation policies, we introduce three fundamental concepts integral to their development: Next Scheduled Departure (NSD), state-dependent sojourn time distributions, and the probability of success.

4.3.1 Next Scheduled Departure (NSD)

Doerr and Gue (2006) define NSD as the fraction of orders arriving in 24 hours between cutoff times t_c on successive days that make it onto the truck in the current day. An earlier cutoff time means more orders arriving before the cutoff time will make it on the departing truck and therefore NSD will be higher; thus, NSD depends directly on the cutoff time. Figure 4.1 illustrates the relationship between the cutoff time and NSD, where t_c^n is the cutoff time and t_d^n is the deadline of day n.

Expected NSD for a system can be computed based on the steady-state sojourn time distribution. Every arriving order sees the same steady-state sojourn time distribution to depart the system. If an order arrives to a system close to the deadline, the probability that it will make it on the next departing truck will be low. Those arriving early in the 24 hour window have the highest chance of making it on the truck. As shown in Figure 4.2, given cutoff time t_c and deadline t_d, the time average of the probability of being completed on time for orders arriving in this 24 hour period is the expected NSD. We can compute this

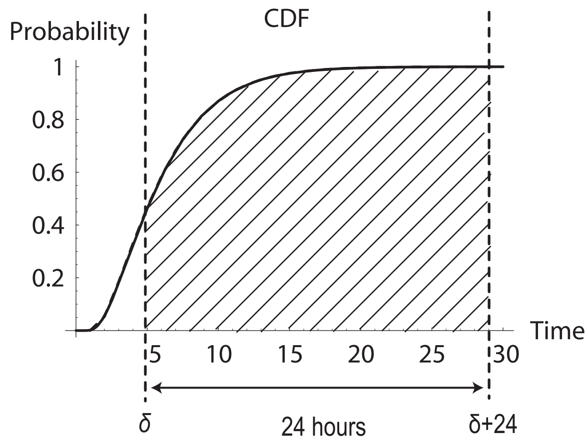


Figure 4.2: The meaning of NSD based on the steady-state sojourn time distribution.

steady-state sojourn time distribution for multi-server queueing networks using methods we develop in Gue and Kim (2009a).

Formally, if $P(t) = P[\ \textit{event occurs at time } t\ ]$, then the time average probability is given by $p = \frac{1}{t}\int_0^t P(t)dt$.

Proposition 4.1 *For an order fulfillment system with deadline* t_d *and cutoff time* t_c*,*

$$NSD = \frac{1}{24}\int_{\delta}^{\delta+24} P[T < t]dt, \tag{4.1}$$

where $0 \le t \le 24$ *and* $\delta = t_d - t_c$*.*

Proof. Let $P(t)$ be the probability that a job arriving at time t is completed before the deadline, then $P(t) = P[T < \delta + 24 - t]$ and by the definition of time average probability,

$$NSD = \frac{1}{24}\int_0^{24} P[T < \delta + 24 - t]dt.$$

where t is elapsed time since the most recent cutoff time t_c.

Let $u = \delta + 24 - t$ and differentiate both sides with respect to t. We get $du/dt = -1$, or $du = -dt$.

$$\begin{aligned} NSD &= \frac{1}{24}\int_0^{24} P[T < \delta + 24 - t]dt \\ &= \frac{1}{24}\int_{\delta+24}^{\delta} P[T < u](-du) \\ &= \frac{1}{24}\int_{\delta}^{\delta+24} P[T < u]du. \end{aligned}$$

□

Given the desired baseline NSD (see Doerr and Gue (2006)) when there is no worker allocation, we compute δ by simple search using Equation 4.1 and decide the cutoff time $t_c = t_d - \delta$.

4.3.2 State-dependent sojourn time distribution

A state-dependent sojourn time distribution provides real-time information about a system, where the state of the system is defined by the number of busy servers, the number of orders in queue, and the processing rate of the servers.

Gue and Kim (2009b) develop an approximation model for the state-dependent sojourn time distribution of orders in a multi-server queueing system, when interarrival and service times can take on general distributions. The model is based on the characteristics of phase-type distributions, and uses a Markov process to represent the all-busy period for a waiting time distribution.

An order in queue (say, the k^{th}) waits $k + 1$ "sub-waiting times" to enter service and departs the system after receiving service. They approximate each "sub-waiting time" and service time as corresponding phase-type distributions $(\boldsymbol{\alpha_k}, W_k)$, $(\boldsymbol{\beta}, S)$ respectively and generate the initial probability vector and infinitesimal generator of the sojourn time distribution of the system $(\boldsymbol{\gamma}, K)$ using the convolution property of the phase-type distribution.

$$K = \begin{bmatrix} W & W^0\alpha_2 & 0 & \cdots & 0 & 0 \\ 0 & W & W^0\alpha_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & W & W^0\alpha_{k+1} \\ 0 & 0 & 0 & \cdots & 0 & S \end{bmatrix},$$

$$\boldsymbol{\gamma} = [1, 0, \cdots, 0].$$

Based on $(\boldsymbol{\gamma}, K)$, the CDF and PDF are given by

$$F(t) = P(T \leq t) = 1 - \gamma e^{Kt}\mathbf{e}, (t \geq 0),$$

$$f(t) = \gamma e^{Kt} K^0 = \gamma e^{Kt}(-K)\mathbf{e}, (t \geq 0),$$

where $\mathbf{e}$ is a column vector of ones. See Gue and Kim (2009b) for details. For our purpose, it is sufficient to know that for any order in a system, we can compute its remaining sojourn time distribution based on the actual state of the system.

4.3.3 The probability of success

In Section 4.1, we reported that a simple worker allocation policy (moving a fixed number of workers at fixed time everyday) without considering the system state is not promising. An alternative, which we show below to be more effective, is to decide the number of workers to switch based on the current system state. To do so, we introduce probability of success p_s as a tool to help determine how many workers to switch.

Probability of success is decided by the system state, the number of orders ahead, the number of servers, the service time distribution, and the remaining time before the deadline. We can compute p_s of an order in the system based on the state-dependent sojourn time distribution model (Gue and Kim, 2009b), as follows. Let a continuous random variable T be the sojourn time

Definition 4.1 *Probability of success*

$$p_s = \begin{cases} P[T < t_r] & \textit{if an order is in queue} \\ P[T < t_\ell | T > t_e] & \textit{if an order is in service} \end{cases} \tag{4.2}$$

where t_r the remaining is the remaining time, t_ℓ is the duration of time between entry into service and the deadline, and t_e is the elapsed time since an order entered service.

We can compute p_s for an order in service from the sojourn time CDF as follows.

$$\begin{aligned} P[T < t_\ell | T > t_e] &= 1 - P[T > t_\ell | T > t_e] \\ &= 1 - \frac{\int_{t_\ell}^{\infty} f_T(u)du}{\int_{t_e}^{\infty} f_T(u)du} \\ &= \frac{\int_{t_e}^{\infty} f_T(u)du - \int_{t_\ell}^{\infty} f_T(u)du}{\int_{t_e}^{\infty} f_T(u)du} \\ &= \frac{\int_{t_e}^{t_\ell} f_T(u)du}{\int_{t_e}^{\infty} f_T(u)du} \\ &= \frac{F(t_\ell) - F(t_e)}{1 - F(t_e)}. \end{aligned}$$

To illustrate, we record p_s for a particular order in a simulated system.

Example 4.1 Consider a single-stage queue with 30 identical servers, with processing times having mean 5 hours and squared coefficient of variation 0.8. An order arrives at 10:00 and sees 19 orders ahead. The last truck departs the system at 17:00. What is the probability of success, and how does it change with time?

To illustrate how p_s changes over time, we run a simulation model and compute p_s and the mean sojourn time $E[S]$ at 15 minute increments, based on a recalculated state-dependent sojourn time distribution. For example, to compute the probability of success of this order at 10:00 (remaining time t_r=7 hours), we generate the phase-type representation of the system $(\boldsymbol{\gamma}, K)$ using the approximation model of Gue and Kim (2009b) based on the system state. Then the probability of success is computed using Equation 4.2,

$$p_s = P(T \leq t_r = 7) = 0.41.$$

Figure 4.3 shows p_s for the order over time. In this case, the order is still in the system at t_d=17:00, when its p_s=0. We see that p_s increases or decreases according to the condition of the system. However, once an order enters service, its p_s can only decrease (see Equation 4.2). Once an order is seized by a worker, there is no way to help its process time by worker allocation because we do not consider worker collaboration on an order.

Our worker allocation policies are based on the insight that as time progresses, the probability of success for orders far from completion goes down and so effort devoted to them might be better spent on orders "on the bubble" of making the next truck. We formalize this idea below.

4.4 Worker allocation policies

We introduce dynamic worker allocation policies that assign cross-trained workers to tasks in an order fulfillment system such that performance against the Next Schedule Departure (NSD) metric is improved. We suggest two types of policies: a *flushing* policy and a *cascade* policy. We further classify a flushing policy as either *single* or *multi*-flush in accordance with the number of switching events.

A single flush policy moves workers based on the state of the shipping area at a defined time in order to flush the system of nearly completed orders right before the truck departs

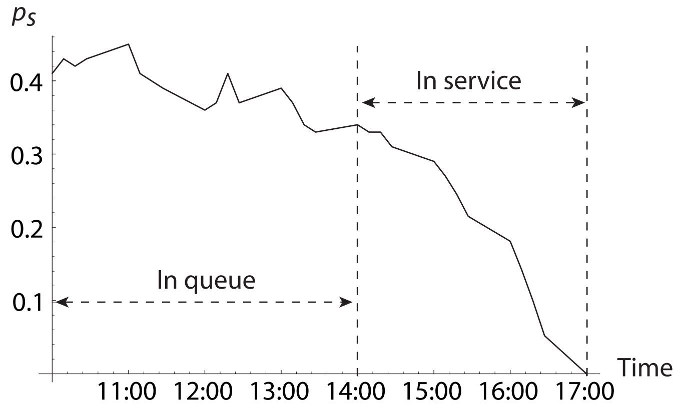


Figure 4.3: A change of p_s of an order while in the system. The probability of success p_s increases or decreases with time because the system state, the number of orders ahead, the number of servers, and the service time distribution change; but p_s of an order always decreases in service because it is a conditional probability of elapsed time t_e.

(Figure 4.4). For this reason, this policy only focuses on the state of the shipping area and the sojourn time distribution of orders in it. One limitation of the single flush policy is that we have only one opportunity per day to correct an overloaded shipping area. In the multi-flush policy, we move workers periodically during a day, considering the state of the shipping area. Multi-flush is the same as single flush, except for the number of switching events.

A cascade policy moves workers sequentially from picking to packing, then from packing to shipping based on the state of the packing and shipping areas at a certain times. The cascade policy is an attempt to flush across more than one workstation. Figure 4.5 shows the concept. At a fixed time, say, 15:00 each day, move workers from picking to packing. At a later time, after flushing orders in the queue of the packing area, move workers from packing to shipping. The goal of the cascade policy is to increase the p_s of an order in the system regardless of its location in a relatively short time.

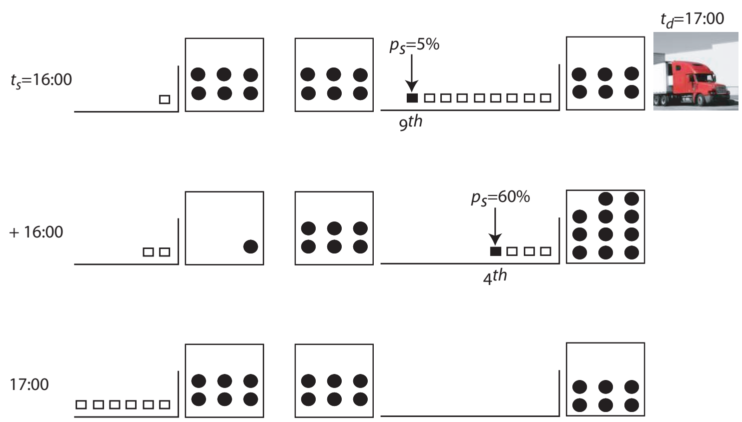


Figure 4.4: Flushing policy

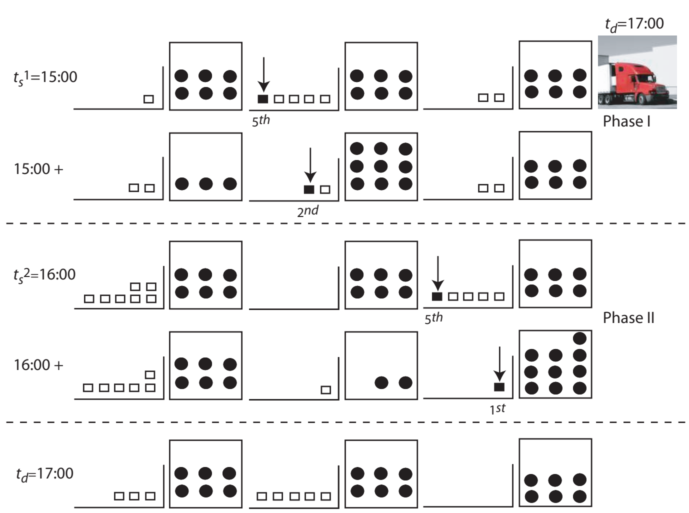


Figure 4.5: Cascade policy

Table 4.1: The characteristics of 12 systems.

	High ρ High SCV	High ρ Low SCV	Low ρ High SCV	Low ρ Low SCV
Small system with short $E[S]$	System 1	System 2	System 3	System 4
Large system with short $E[S]$	System 5	System 6	System 7	System 8
Small system with long $E[S]$	System 9	System 10	System 11	System 12

4.5 Experiments

To make definitive statements about which policy is "best" would require that we test it on an infinite number of potential systems. We settle for a more limited experiment on 12 carefully constructed systems, which we believe stress the model along important dimensions. All of the example systems are 3-stage serial lines, in keeping with the picking-packing-shipping structure of many order fulfillment systems. The systems vary according to size (number of workers), mean sojourn time, variability of processing times, and utilization. Table 4.1 summarizes the systems.

We select three basic candidate systems according to size and mean sojourn time, then divide each system into four different systems based on two levels of the squared coefficient of variation (SCV: 0.5, 0.9) and utilization (ρ=0.85, 0.95). For the squared coefficient of variation, we deal exclusively with the case SCV<1 because it is common in practice. We consider three basic serial lines. The first is a small system (total 31 workers) with short mean sojourn time (6.52 hours); the second is a large system (total 126 workers) with short mean sojourn time (7.31 hours); and the third is a small system (total 44 workers) with long mean sojourn time (19.76 hours). Table 4.2 shows system information of the representative systems 1, 5 and 9.

To implement a worker allocation policy correctly, we need to answer the questions: how many workers should we move and when? Switching workers too early and switching

Table 4.2: System information for Systems 1, 5, and 9.

System 1	$E[T]$	C^2	ρ	Workers
Interarrival	0.117	0.75		
Picking	1.07	0.9	0.91	10
Packing	1.3	0.9	0.93	12
Shipping	1.0	0.9	0.95	9
System 5	$E[T]$	C^2	ρ	Workers
Interarrival	0.053	0.75		
Picking	2.7	0.9	0.92	56
Packing	1.5	0.9	0.95	30
Shipping	2.0	0.9	0.95	40
System 9	$E[T]$	C^2	ρ	Workers
Interarrival	0.23	0.75		
Picking	3.3	0.9	0.96	15
Packing	4.3	0.9	0.93	20
Shipping	2.0	0.9	0.97	9

too many workers can cause imbalance in the system, whereas switching too late or too few might not have an effect on NSD.

The exact number of switching workers is decided by the target p_s (the probability of success of an order of interest that we want to reach by worker allocation), and switching time t_s, so we test the 12 systems to find best t_s and the target p_s for each policy. For example, in Figure 4.4, the probability of success p_s for the last order in the shipping queue is 5% based on the number of orders ahead 8, the number of workers 6, and the remaining time t_r=1 hour ($t_d - t_s$). To reach the target p_s=60% for this order, we add 5 workers to shipping, then the system state is changed: the number of orders ahead is 4, and the number of workers 11. The number of workers required is computed by simple search, based on the method of Gue and Kim (2009b)

$$\text{Target } p_s = 60\% = P(T \leq t_r = 1) = 1 - \gamma e^{Kt_r}\mathbf{e}.$$

where (γ, K) depends on the system state.

We use 6 different levels of the target p_s; 10%, 30%, 50%, 60%, 70%, 90%, and use switching time t_s from early in the morning to close to the deadline. We do not include a switching time that makes the remaining time t_r less than the mean processing time of the last workstation because this situation limits the effect of the policy (more on this below).

We run simulation models built in Arena 12.0. Each scenario runs 100 times and each run has 50 days' NSD results. For easy implementation of the simulation model, in advance, we compute the exact number of workers to reach the given target p_s based on current p_s for possible combinations between target p_s and switching time t_s for each system. Then, we just move required workers based on the number in queue in the shipping area based on this given information. For example, in the case with the target p_s=60% and switching time t_s=16:00 for System 1, suppose there are 4 orders in the shipping queue at 16:00, then we just move 4 workers from picking to shipping based on the given information. In our simulation model, we assume two important points: We do not consider worker transition time when we move workers from one workstation to other workstations. This is not true in practice, of course, but transition times are highly application specific, so we ignore them for this analysis. We also assume that the picking process is not a batch process. For many order fulfillment systems, this will not be true, but for others (part-to-picker systems, in particular) it is reasonable.

4.5.1 Single Flush

The truck departure time for all experiments is 17:00. To find the best t_s and target p_s for the single flush policy, we use

- Target p_s: {10%, 30%, 50%, 60%, 70%, 90%}, and
- Switching time t_s: {16:00, 15:00, 14:00, 13:00, 07:00} for Systems 1–4, {15:00, 14:00, 13:00, 12:00, 07:00} for Systems 5–8, {15:00, 14:00, 13:00, 12:00, 07:00, 02:00} for Systems 9–12.

We choose candidate t_s from early in the morning (02:00 or 07:00) to close to the deadline (16:00 or 15:00). We do not consider t_s later than 16:00 (Systems 1–4) or 15:00 (Systems 5–12) because mean service times of the shipping area $E[T_3]$ are 1 or 2 hours. If the remaining time t_r is shorter than $E[T_3]$, then p_s of the switching time is too low to have an effect on NSD. The algorithm is:

Algorithm 1 Single Flush (SF)

Step 1: Decide the cutoff time t_c using the steady-state sojourn time distribution, based on the baseline NSD.
Step 2: Check the number of orders in front of the shipping area at a given switching time t_s.
Step 3: Compute p_s of the last order in the shipping queue using the state-dependent sojourn time distribution.
Step 4: If p_s is below a given target, compute the number of switching workers required to reach the target p_s and move these workers from the picking to the shipping area. Otherwise, do not move workers.
Step 5: Return workers to their original place when clock time reaches the deadline t_d.

First, we decide cutoff time t_c using the steady-state sojourn time distribution model under baseline NSD 77% (fixed model without worker allocation of System 1):

$$\begin{aligned} \text{Baseline NSD} = 77\% &= \frac{1}{24}\int_{\delta}^{24+\delta} P[T < t]dt \\ \delta &\approx 1 \text{ hour} \\ t_c &= 16:00 \end{aligned}$$

Second, we test the single flush policy under possible combinations of t_s and target p_s. There are total 384 different scenarios. Table 4.3 shows the results of System 1; other test results are shown in the appendix. We compare single flush and the fixed worker model in terms of the mean sojourn time $E[S]$, expected number of switching workers $E[SW]$, and NSD. We see that NSD of the single flush policy with any combination of t_s and target p_s is

Table 4.3: Results of the single flush policy for System 1.

	Fixed	t_s=16:00					
		p_s=10%	30	50	60	70	90
$E[S]$	6.53	5.76	5.70	5.69	5.71	5.71	5.71
$E[SW]$	0	2.27	3.60	4.50	5.12	5.12	5.12
NSD	**77.25**	**80.56**	**81.10**	**81.05**	**81.26**	**81.26**	**81.26**

	Fixed	t_s=15:00					
		p_s=10%	30	50	60	70	90
$E[S]$	6.53	5.94	5.88	5.83	5.85	5.84	6.10
$E[SW]$	0	0.81	1.22	1.66	1.99	2.17	5.24
NSD	**77.25**	**80.10**	**80.41**	**80.89**	**80.72**	**81.03**	**80.77**

	Fixed	t_s=14:00					
		p_s=10%	30	50	60	70	90
$E[S]$	6.53	6.08	6.03	6.01	5.99	6	6.06
$E[SW]$	0	0.34	0.52	0.65	0.79	1.02	1.83
NSD	**77.25**	**79.35**	**79.51**	**79.88**	**79.90**	**80.24**	**80.45**

	Fixed	t_s=13:00					
		p_s=10%	30	50	60	70	90
$E[S]$	6.53	6.10	6.15	6.10	6.11	6.08	6.05
$E[SW]$	0	0.32	0.23	0.29	0.36	0.44	0.75
NSD	**77.25**	**79.08**	**78.91**	**79.42**	**79.32**	**79.42**	**79.87**

	Fixed	t_s=07:00					
		p_s=10%	30	50	60	70	90
$E[S]$	6.53	6.37	6.46	6.46	6.46	6.46	6.43
$E[SW]$	0	0.14	0.00	0.00	0.00	0.01	0.01
NSD	**77.25**	**78.24**	**77.40**	**77.53**	**77.55**	**77.48**	**77.54**

better than the fixed model. NSD is improved by 0.25–4.01%. Moreover, $E[S]$ decreases by 0.07–0.84 hours compared with fixed model.

In Table 4.3 notice that the target p_s=60, 70, 90% for t_s=16:00 have the same results. We can explain this phenomenon with Figure 4.6. Given the remaining time t_r=1 hour, the maximum achievable p_s of any order in the shipping queue using a worker allocation policy is about 60%. In other words, if we switch workers when the remaining time equals the mean processing time of the final workstation, we can never increase the p_s of the last order in the shipping queue beyond approximately 60 percent, *no matter how workers we switch.* Why is this? Suppose there are m orders in the shipping queue at the switching time, and

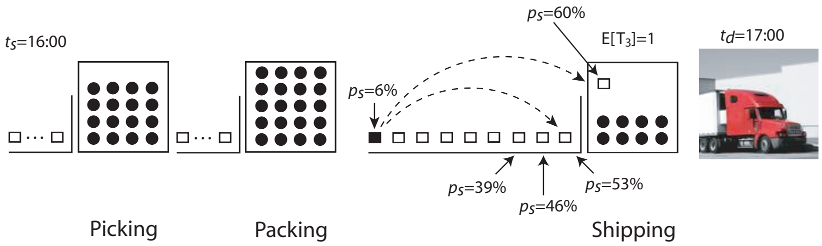


Figure 4.6: The maximum p_s that an order can reach by worker allocation: If we add 6 workers to shipping, the p_s of the last order in the shipping queue increases to 39%. If we add 9 workers, then the p_s of the last order reaches 60%. Adding more than 9 workers creates idle workers.

that the time remaining equals the mean processing time of shipping. Moving more than m workers from picking to shipping makes little sense because switching workers $m + 1$ and beyond would have no order to work on. If we move m workers, then the last order in the shipping queue goes into service with exactly $E[T_3]$ time remaining, and we would expect its p_s immediately to jump to $P[T_3 < E[T_3]]$. For a symmetric distribution this would be 50 percent, but our model of T_3 is Erlang (k,μ) because we expect SCV<1 for shipping. For the parameters we have used in our testing, $P[T_3 < E[T_3]] \approx 0.6$ for the Erlang distribution. We can conclude, then, that target p_s values greater than about 60% are not achievable when switching workers with only $E[T_3]$ time remaining.

In Table 4.3, we see that NSD in the same target p_s decreases as we set switching time t_s earlier. Table 4.4 includes results of all 12 systems and it shows this trend. Earlier switching is ineffective because switched workers are often left in shipping after the conditions which caused their movement have dissipated. Moreover, very few workers tend to move at all because at this early time the p_s values of orders in the shipping queue are usually quite high.

Unlike switching time t_s, it looks as if there is no difference on NSD among policies at the same t_s with different target p_s (Table 4.3). To find the best target p_s, we do a series of t-tests among scenarios with different p_s based on 100 runs of each scenario (Kelton, 2003). For switching time 07:00, there is no statistical difference among p_s values because very few

Table 4.4: Change of NSD according to switching times t_s.

t_s	Ex.1	Ex.2	Ex.3	Ex.4	Ex.5	Ex.6	Ex.7	Ex.8	Ex.9	Ex.10	Ex.11	Ex.12
15:00(16:00)	81.1	82.7	87.5	88.0	78.0	78.6	81.9	82.0	80.4	77.6	70.0	70.7
14:00(15:00)	80.7	82.1	87.3	87.8	77.7	78.2	81.9	82.0	79.7	76.3	69.8	70.5
13:00(14:00)	79.9	81.7	87.1	87.7	77.3	77.9	81.8	81.9	79.3	76.1	69.5	70.3
12:00(13:00)	79.3	81.1	87.0	87.7	76.9	77.8	81.8	81.9	78.1	74.9	69.3	70.1
07:00	77.6	80.2	86.9	87.6	76.7	77.7	81.8	81.9	75.5	72.1	68.8	69.9
02:00									73.6	70.3	68.8	69.9

workers move at all ($E[SW] \leq 0.14$ workers), and it is difficult to increase NSD beyond that achieved by the fixed worker model. For later switching times, however, higher p_s values are the best. We can see this trend in Figure 4.7, which shows the best scenarios for all 12 systems based on a series of t-test. We do not include any target p_s which exceeds maximum achievable p_s (empty space in the figure).

For the collection of graphics in Figure 4.7, we detect a pattern (see Figure 4.8): when switching time is close to the deadline, almost any p_s value is effective because the situation of the p_s value of orders in the shipping queue less than the target p_s at switching time occurs frequently. When the switching time is very early, all p_s values are "best" because none are effective at all—the early switching time prevents any significant effect on NSD, irrespective of p_s.

In summary, then, we see that the single flush policy can significantly increase NSD and decrease mean sojourn time $E[S]$. A later switching time t_s and higher target p_s are best in almost every case. In general, we can say that the target $p_s = 60\%$ at switching time $t_s = t_d - E[T_3]$ is the best condition for the single flush. Especially, we should also note that the single flush with the target $p_s = 60\%$ at switching time $t_s = t_d - E[T_3]$ means the number moving of workers equals the number of orders in the shipping queue at the switching time. This suggests a very simple but effective policy for success.

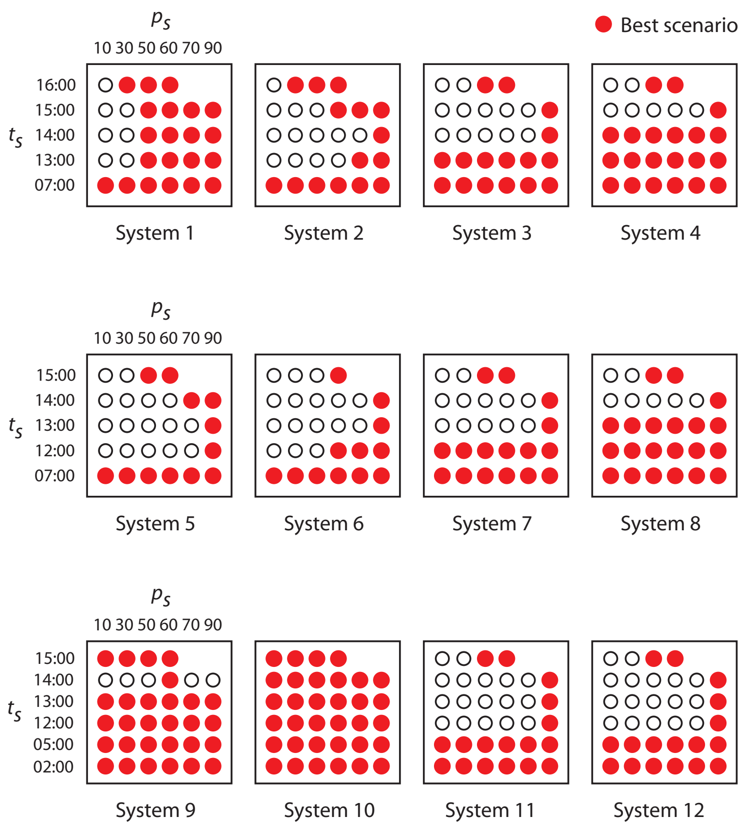


Figure 4.7: Best scenarios for all 12 Systems: Open circles represent good scenario; gray circles represent best scenario; empty space has the same NSD with the next lower p_s.

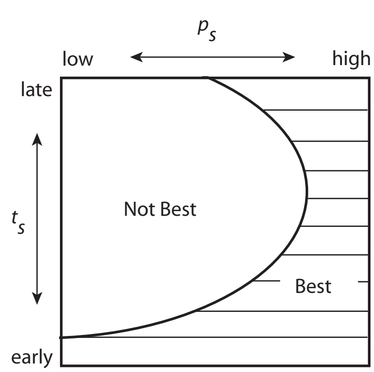


Figure 4.8: An approximate pattern for scenarios in Figure 4.7

4.5.2 Multi-Flush

We set the switching period t_p as the mean service time of the shipping area $E[T_3]$, because we found in our analysis of single flush policies that a late switching time t_s is more effective than an early switching time. Thus the number of workers for each switching period is decided based on the system state, especially under the remaining time $t_r = t_p = E[T_3]$. To find the best target p_s for the multi-flush, we use 6 target p_s values (10%, 30%, 50%, 60%, 70%, 90%) as used in the single flush experiment. The number of switching events varies from two to five. The algorithm is as follows:

Algorithm 2 Multi-Flush (MF)

Step 1: Decide the cutoff time t_c using the steady-state sojourn time distribution model, based on the baseline NSD.
Step 2: Check the number of orders in front of the shipping area every period.
Step 3: Compute p_s of the last order in the shipping queue using the state-dependent sojourn time distribution under $t_r = t_p$.
Step 4: If p_s is below the target p_s, compute the number of switching workers to reach the target p_s and move these workers from the picking to the shipping area. Otherwise, do not move workers.
Step 5: If clock time does not meet the deadline and reaches the periodic switching time, return workers to their original place and go to Step 2. Otherwise, go to Step 6.
Step 6: Return workers to their original place when clock time reaches the deadline t_d.

We compare system performance of the multi-flush with the fixed model in Table 4.5, which shows test results of System 1. Test results of other Systems are shown in the appendix. The multi- flush policy shows good results on NSD (increase by 4.16–5.58%) and $E[S]$ (decrease by 0.88–0.98 hours) compared with the fixed model, regardless of the number of switching events. More switching events means more switching workers, so in practice we would probably favor fewer switching events. Notice that the target p_s=60, 70, 90% for every switching t_s have the same results. This is the same phenomenon we observed for the single flush policy.

To find the best target p_s among 10, 30, 50, 60% for the same switching time t_s, we do a series of t-test among these target p_s values and we find that the higher target $p_s = 60\%$ is

Table 4.5: Multi-flush results for System 1.

	Fixed	2 Times (t_s=16:00, 15:00)					
		p_s=10%	30%	50%	60%	70%	90%
$E[S]$	6.54	5.73	5.68	5.65	5.68	5.68	5.68
$E[SW]$	0.00	2.73	3.89	5.40	6.20	6.20	6.20
NSD	**77.25**	**81.42**	**82.01**	**82.26**	**82.40**	**82.40**	**82.40**
	Fixed	3 Times (t_s=16:00, 15:00, 14:00)					
		p_s=10%	30%	50%	60%	70%	90%
$E[S]$	6.54	5.69	5.65	5.65	5.64	5.64	5.64
$E[SW]$	0.00	3.06	4.32	6.16	7.04	7.04	7.04
NSD	**77.25**	**81.61**	**82.30**	**82.73**	**82.77**	**82.77**	**82.77**
	Fixed	4 Times (t_s=16:00, 15:00, 14:00, 13:00)					
		p_s=10%	30%	50%	60%	70%	90%
$E[S]$	6.54	5.66	5.62	5.62	5.64	5.64	5.64
$E[SW]$	0.00	3.30	4.70	6.79	7.89	7.89	7.89
NSD	**77.25**	**81.99**	**82.83**	**82.75**	**82.65**	**82.65**	**82.65**
	Fixed	5 Times (t_s=16:00, 15:00, 14:00, 13:00, 12:00)					
		p_s=10%	30%	50%	60%	70%	90%
$E[S]$	6.54	5.64	5.62	5.59	5.61	5.61	5.61
$E[SW]$	0	3.52	5.04	7.34	8.68	8.68	8.68
NSD	**77.25**	**81.83**	**82.30**	**82.45**	**82.66**	**82.66**	**82.66**
	Fixed	6 Times (t_s=16:00, 15:00, 14:00, 13:00, 12:00, 11:00)					
		p_s=10%	30%	50%	60%	70%	90%
$E[S]$	6.54	5.62	5.56	5.57	5.63	5.63	5.63
$E[SW]$	0	3.70	5.33	7.94	9.46	9.46	9.46
NSD	**77.25**	**82.28**	**82.35**	**82.46**	**82.49**	**82.49**	**82.49**

always selected as the best scenario. We can conclude that the target $p_s = 60\%$, $t_r = E[T_3]$, and 2 switching times is the best condition for the multi-flush in terms of the system stability and simplicity.

4.5.3 Cascade policy

Based on the results of the flushing policy, we set the target p_s as 60% and the first and second switching time t_s^1, t_s^2 considering the mean processing time of the packing and shipping area $E[T_2]$, $E[T_3]$. For example, first we check the state of the packing area at time $t_s^1 = t_d - (E[T_2] + E[T_3])$, and decide the number of switching workers (from picking to packing) to reach the target p_s based on the remaining time $t_r = E[T_2]$. In the same manner, we compute the number of switching workers (from packing to shipping) to reach the target p_s under $t_r = E[T_3]$ at time $t_s^2 = t_d - E[T_3]$. The algorithm is as follows:

Algorithm 3 Cascade (C)

Step 1: Decide the cutoff time t_c using the steady-state sojourn time distribution model, based on the baseline NSD.
Step 2: Check the number of orders in front of packing area at time $t_s^1 = t_d - (E[T_2] + E[T_3])$.
Step 3: Compute p_s of the last order in the packing queue using the state-dependent sojourn time distribution under $t_r = E[T_2]$.
Step 4: If p_s is below the target p_s, compute the number of switching workers to reach the target p_s and move these workers from the picking to the packing area. Otherwise, do not move workers.
Step 5: Return workers to their original place when clock time reaches $t_s^2 = t_d - E[T_3]$, and check the number of orders in front of the shipping area.
Step 6: Compute p_s of the last order in the shipping queue using the state-dependent sojourn time distribution under $t_r = E[T_3]$.
Step 7: If p_s is below the target p_s, compute the number of switching workers to reach the target p_s and move these workers from the packing to the shipping area. Otherwise, do not move workers.
Step 8: Return workers to their original place when clock time reaches the deadline t_d.

In Table 4.6, we compare the system performance among three policies: cascade, a single flush with the target $p_s = 60\%$ and $t_s = t_d - E[T_3]$, and a multi-flush with the target $p_s = 60\%$, 2 times switching with the switching period $t_r = E[T_3]$ for all 12 systems. In general, there is very little difference in system performance among them. The cascade policy performs poorly in Systems 9 and 10, because the switching duration of workers away from picking is 6.3 hours ($E[T_2]+E[T_3]$), which is quite long compared with the switching duration of the flushing policy ($E[T_3]$=2 hours). The extended absence from the picking station leads to the very high average queue level at picking (44.7, 50.5 orders). Moreover, Systems 9 and 10 have high utilization and high variation, which leads to large queues in front of each workstation. Another observation from Table 4.7 is that dynamic worker allocation policies seem to have more effect on systems with higher utilization ρ and variation (SCV), because these systems have longer queues.

From comparison among the three policies, we see that the multi-flush and cascade policies are not much better than the single flush policy, except for System 9 and 10. However, the multi-flush and cascade policies use more switching workers than the single flush policy, so they could be more difficult to implement. If switching workers is easy, or a small increase, say 1%, is important for the system, then the multi-flush policy might be adopted.

4.6 Chapter conclusions

In this research, we propose several dynamic worker allocation policies for due-date order fulfillment systems. We showed that we can increase service performance (NSD) and decrease mean sojourn time through dynamic worker allocation. This means that we can, in fact, improve service performance using appropriate worker reallocation without disturbing the system balance.

We propose flushing policies (single flush, multi-flush) and a cascade policy as candidate dynamic worker allocation policies. To find the best policy and switching time and the target probability of success for each policy, we performed thorough testing on 12 systems. We find that late switching times (short remaining time) have a greater effect on NSD than earlier times. We also find that the higher target probability of success has more benefit on NSD, because it has more chance for worker allocation using more workers. However, it does not

Table 4.6: Cascade policy results and comparison among three policies.

		System 1				System 2				System 3				System 4			
		F	SF	MF	C	F	SF	MF	C	F	SF	MF	C	F	SF	MF	C
$E[S]$		6.5	5.7	5.7	5.6	5.8	5.2	5.2	5.2	4.2	4.1	4.1	4.0	4.0	3.9	3.9	3.9
$E[SW]$		0.0	5.1	6.2	9.5	0.0	4.7	5.5	8.4	0.0	2.3	2.8	5.0	0.0	1.8	2.4	4.4
NSD		**77.3**	**81.3**	**82.4**	**82.5**	**80.0**	**82.9**	**84.1**	**83.9**	**86.9**	**87.6**	**88.0**	**88.0**	**87.6**	**88.1**	**88.4**	**88.4**
Average	Pick	5.8	7.2	8.2	8.0	5.0	6.1	7.2	6.8	1.7	1.8	1.9	1.9	1.4	1.4	1.5	1.5
orders in	Pack	7.4	7.8	6.0	7.8	5.4	5.6	4.6	5.6	1.9	1.9	1.8	1.9	1.3	1.4	1.3	1.4
queue	Ship	12.6	5.4	5.1	3.8	9.6	4.2	4.0	3.1	2.8	2.1	2.1	1.7	2.0	1.6	1.6	1.3
		System 5				System 6				System 7				System 8			
		F	SF	MF	C	F	SF	MF	C	F	SF	MF	C	F	SF	MF	C
$E[S]$		7.6	7.3	7.2	7.2	7.4	7.1	7.1	7.0	6.3	6.3	6.3	6.3	6.3	6.3	6.3	6.3
$E[SW]$		0.0	7.4	8.4	13.2	0.0	6.1	7.1	11.7	0.0	1.1	1.5	3.5	0.0	0.8	1.3	3.0
NSD		**76.7**	**78.5**	**78.7**	**78.9**	**77.7**	**78.9**	**79.3**	**79.3**	**81.8**	**82.0**	**82.0**	**82.0**	**81.9**	**82.1**	**82.1**	**82.0**
Average	Pick	3.8	4.4	4.4	3.9	3.2	3.8	3.8	3.4	0.3	0.3	0.3	0.3	0.2	0.3	0.3	0.3
orders in	Pack	12.3	12.2	12.2	11.0	10.3	10.2	10.2	9.7	1.3	1.4	1.4	1.4	1.1	1.1	1.1	1.1
queue	Ship	11.9	2.6	2.6	4.9	8.7	2.3	2.3	4.1	1.0	0.6	0.6	0.9	0.8	0.5	0.5	0.7
		System 9				System 10				System 11				System 12			
		F	SF	MF	C	F	SF	MF	C	F	SF	MF	C	F	SF	MF	C
$E[S]$		19.5	16.5	16.6	**21.7**	17.8	15.5	15.6	**22.7**	11.1	10.8	10.8	10.9	10.8	10.6	10.6	10.7
$E[SW]$		0.0	5.5	6.0	11.0	0.0	4.9	5.3	9.9	0.0	2.3	2.4	4.6	0.0	1.9	2.4	3.8
NSD		**70.3**	**80.4**	**81.1**	**75.4**	**68.6**	**77.2**	**77.5**	**74.1**	**68.8**	**70.3**	**70.7**	**70.1**	**69.9**	**71.0**	**71.6**	**70.9**
Average	Pick	13.9	18.4	20.8	**44.7**	13.0	16.9	20.2	**50.5**	2.0	2.0	2.1	2.4	1.6	1.6	1.7	2.0
orders in	Pack	8.7	8.5	8.4	5.0	6.1	5.8	5.7	3.6	1.2	1.3	1.3	1.1	0.9	0.9	0.9	0.8
queue	Ship	20.9	3.6	1.8	3.5	16.8	3.0	1.6	3.0	2.9	1.7	0.9	1.7	2.1	1.4	0.8	1.4

*F: fixed model, SF: single flush, MF: multi-flush, C: cascade.

Table 4.7: The system performance according to the system characteristic.

			High ρ High SCV	High ρ Low SCV	Low ρ High SCV	Low ρ Low SCV
			System 1	System 2	System 3	System 4
Small system with short $E[S]$	Average orders in queue	Pick	5.8	5.0	1.7	1.4
		Pack	7.4	5.4	1.9	1.3
		Ship	12.6	9.6	2.8	2.0
	*Increase in NSD (%)		**4.01**	2.94	0.66	0.51
			System 5	System 6	System 7	System 8
Large system with short $E[S]$	Average orders in queue	Pick	3.8	3.2	0.3	0.2
		Pack	12.3	10.3	1.3	1.1
		Ship	11.9	8.7	1.0	0.8
	Increase in NSD (%)		**1.79**	1.25	0.16	0.11
			System 9	System 10	System 11	System 12
Small system with long $E[S]$	Average orders in queue	Pick	13.9	13.0	2.0	1.6
		Pack	8.7	6.1	1.2	0.9
		Ship	20.9	16.8	2.9	2.1
	Increase in NSD (%)		**10.08**	8.63	1.51	1.08

*Fixed model - Single flush.

disturb the system balance because the number of workers is decided based on the system state at that switching time.

Based on these best conditions of the switching time and target probability of success, we test and compare three policies and find that there is very little difference in system performance among them. Although the multi-flush and cascade policy (except for long switching duration-early switching time) are a bit better than the single flush, we believe the single flush is most likely the best policy in terms of system stability and ease of implementation.

Another important fact is that the system with long waiting time has more room to improve NSD. Utilization and variation are the important factors that cause system delay, so the system with high utilization and variation can be improved significantly by our dynamic worker allocation policies. This suggests that the methods we develop here would have their highest return for systems in heavy traffic.

Although the policies we develop rely heavily on the ability to compute state-dependent sojourn time distributions, the results of our experiments suggest a simple policy that requires no mathematics at all: (1) Establish the switching time at the deadline minus the mean processing time of shipping; (2) At the switching time each day, move a number of workers from picking to shipping equal to the size of the shipping queue, not to exceed the total number in picking. This simple policy is approximately equivalent to the single-flush policy with target p_s=60% and switching time $t_s = t_d - E[T_3]$, so we would expect very good performance with it.

Chapter 5

Conclusions and Outlook

Speed of response to customer orders is one of the most important factors of success for order fulfillment systems. The purpose of this dissertation is to increase customer satisfaction by improving the delivery speed with dynamic worker allocation. To implement dynamic worker allocation correctly, we need to decide the number of switching workers and switching time based on the probability of success of an order in the system. When this probability drops below a threshold value, then we move some workers to improve the probability of success for other, more promising orders, and improve service performance of the system. To compute the probability of success of an order in the system at any time, we need to know state-dependent sojourn time distribution.

In this dissertation, we develop two fundamental approximation models; for steady-state and state-dependent sojourn time distributions. We use these models to design a policy that assigns cross-trained workers to tasks in an order fulfillment system such that performance against the Next Scheduled Departure (NSD) metric is maximized. We summarize the following contributions of this research.

First, we develop the first approximation model for steady-state sojourn time distributions in acyclic multi-server queueing networks. We show that the relationship between NSD and cutoff time t_c for dynamic worker allocation and the NSD is computed by the steady-state sojourn time distribution. Thus, managers can decide the cutoff time t_c of their system based on this approximation model. We also present a method that can decide the optimal cutoff time t_c^* for the maximum shipping profit of a company based on the steady-state sojourn time distribution. However, this method is limited only to the systems with a stationary arrival stream because of the assumption of our steady-state sojourn time distribution model. In future research, we would like to estimate the sojourn time distribution of an arriving order to systems with non-stationary arrival streams.

Second, we develop the first approximation model for state-dependent sojourn time distributions in multi-server single-stage and acyclic queueing networks. We can compute the probability of success p_s of an order in a system at any time, where the system is defined by the number of busy servers and the number of orders in queue at any workstation, and this makes dynamic worker allocation possible. Our approximation model performs well over a wide range of problem instances. However, our single-stage approximation model is limited to problems with fewer than about 200 servers, because the computation time of our model increases as the number of servers and orders ahead in queue increases. Our model is also limited to an acyclic queueing network because we use the expected number of orders ahead at each workstation instead of the number of orders ahead. If the system has re-entrant and external inflow for each workstation, these quantities can be difficult to compute.

The state-dependent sojourn time approximation model allows us to estimate the probability that the state-dependent sojourn time will be less than a desired time t. We can use this result in various ways. For example, an order fulfillment system can do dynamic order promising to customers at the point of order. Service systems such as amusement parks also could inform customers of their wait time with an estimated probability.

Third, we present several dynamic worker allocation policies that assign cross-trained workers to tasks in an order fulfillment system to increase customer satisfaction based on the sojourn time distribution. We test three dynamic worker allocation policies: the single flush, multi flush, and the cascade policy. To select the best policy and conditions, we do thorough experiments based on 12 different systems. We find that the single and multi flush policy work well irrespective of the system characteristics, but the cascade policy is limited to systems with relatively short queues. Also the policies with late switching times and higher target probability of success have a greater effect on NSD. However our policies do not disturb the system balance because the number of switching workers is decided based on the system state. Our results suggest that it is possible to improve service performance—significantly, in some conditions—by moving the right number of workers to the right place at the right time.

Finally, investigation is needed on approximation models for multi-server queueing networks with batch processes. In our dynamic worker allocation, we assume that the picking process is not a batch process, but this is not reasonable for many systems. We plan to take this up in future research.

Bibliography

Askin, R. and Chen, J. (2006). Dynamic Task Assignment for Throughput Maximization with Worksharing. *European Journal of Operational Research*, (168):853–869.

Asmussen, S. and Møller, J. R. (2001). Calculation of the Steady State Waiting Time Distribution in GI/PH/c and MAP/PH/c Queues. *Queueing Systems*, 37:9–29.

Asmussen, S. and O'Cinneide, C. A. (1998). Representations for Matrix-geometric and Matrix-exponential Steady-state Distributions with Applications to Many-server Queues. *Stochastic Models*, 14:369–387.

Bartholdi, J. J. and Eisenstein, D. (1996). A Production Line That Balances Itself. *Operations Research*, 44(1):21–34.

Bartholdi, J. J., Eisenstein, D. D., and Foley, R. D. (2001). Performance of Bucket Brigades When Work is Stochastic. *Operations Research*, 49(5):710–719.

Bischak, D. (1996). Performance of a Manufacturing Module with Moving Workers. *IIE Transactions*, (28):723–733.

Chen, J. and Askin, R. (2006). Throughput Maximization in Serial Production Lines with Worksharing. *Int.J.Production Economics*, (99):88–101.

Curry, G. L. and Feldman, R. M. (2009). *Manufacturing Systems Modeling and Analysis.* Springer.

Doerr, K. and Gue, K. (2006). A Performance Metric and Goal Setting Procedure for Order Fulfillment Operations. Working paper, Naval Postgraduate School, Monterey, CA.

Gel, E., Hopp, W., and Van Oyen, M. (2002). Factors Affecting Opportunity of Worksharing as a Dynamic Line Balancing Mechanism. *IIE Transactions*, (34):847–863.

Gel, E., Hopp, W., and Van Oyen, M. (2007). Hierarchical Cross-training in Work-in-process-constrained Systems. *IIE Transactions*, (39):125–143.

Gue, K. R. and Kim, H. H. (2009a). An Approximation Model for Sojourn Time Distributions in Acyclic Multi-server Queueing Networks.

Gue, K. R. and Kim, H. H. (2009b). Predicting Departure Times in Multi-Stage Queueing Systems.

Internetretailer (2009). Internet Retailer Top 500 Guide. "`http://www.internetretailer.com`".

Jouini, O. and Dallery, Y. (2006). Predicting Queueing Delays for Multiclass Call Centers. In *valuetools '06: Proceedings of the 1st international conference on Performance evaluation methodolgies and tools*, New York, NY, USA.

Kelton, W. D. Sadowski, R. P. S. D. T. (2003). *Simulation with Arena.* McGraw-Hill Professional.

Law, A. M. and Kelton, W. D. (2000). *Simulation Modeling and Analysis.* McGraw-Hill, third edition.

Luh, H. and Zheng, Z. X. (2005). PH/PH/1 Queueing Models in Mathematica for Performance Evaluation. *International Journal of Operations Research*, 2(2):81–88.

Mandelbaum, A., Massey, W. A., and Reiman, M. I. (1998). Strong Approximations for Markovian Service Networks. *Queueing Systems*, 30:149–201.

Marie, R. (1980). Calculating Equilibrium Probabilities for $\lambda(n)/c_k/1/n$ Queues. *In Proceedings of Performance*, pages 117–125.

McClain, J., Schultz, K., and Thomas, L. (2000). Management of Worksharing Systems. *Manufacturing and Service Operations Management*, 2(1):49–67.

Nakibly, E. (2002). *Predicting Waiting Times in Telephone Service Systems.* Ph.d. thesis, The Senate of the Technion, Haifa, Israel.

Neuts, M. F. (1981). *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach.* Johns Hopkins University Press, Baltimore, Maryland.

Neuts, M. F. (1989). *Structured Stochastic Matrices of* M/G/1 *type and their Applications.* Marcel Dekker, INC., New York, New York.

Osogami, T. and Harchol-Balter, M. (2003). A Closed-form Solution for Mapping General Distributions to Minimal PH Distributions. *Performance Evaluation, Special issue for the selected best papers of TOOLS 2003*, 63(6):524–552.

Ostalaza, J., McClain, J., and Thomas, L. (1990). The Use of Dynamic (State-dependent) Assembly-Line Balancing to Improve Throughput. *J. Manufacturing and Oper. Management*, (3):105–133.

Rueda, J. E. (2003). The Ph_t/Ph_t/s/c Queueing Model and Approximation. Master's thesis, Virginia Tech.

Sauer, C. and Chandy, K. (1975). Approximate Analysis of Central Server Models. *IBM Journal of Research and Development*, 19:301–313.

Sengupta, B. (1989). Markov Processes Whose Steady State Distribution is Matrix-exponential with an Application to the GI/PH/1 Queue. *Advances in Applied Probability*, 21:159–180.

Shanthikumar, J. and Sumita, O. (1988). Approximations for the Time Spent in a Dynamic Job Shop with Applications to Due Date Assignment. *International J. Production Res.*, 26:1329–1352.

Tijms, H. C. (1994). *Stochastic Models: An Algorithmic Approach*. John Wiley & Sons, Chichester.

Whitt, W. (1983). The Queuing Network Analyzer. *Bell Systems Technical Journal*, 62(9):2779–2815.

Whitt, W. (1999). Predicting Queueing Delays. *Management Science*, 45(6):870–888.

Yoon, B. S. (1994). An Approximation Method for Sojourn Time Distributions in General Queueing Networks. *Journal of the Korean Operations and Management Science*, 19(3):93–109.

You, J. U., Chung, M. Y., and Sung, D. K. (2002). Performance Evaluation Using an Approximation Method for Sojourn Time Distributions in an IN/ISDN Signaling Platform. *Computer Communications*, 25:1283–1296.

Zavadlav, E., McClain, J., and Thomas, L. (1996). Self-buffering, Self-balancing, Self-flushing Production Lines. *Management Science*, 42(8):1151–1164.

Appendices

Table 1: Single flush of the System 1 and 2

System 1	Fixed	t_s=16:00						15:00						14:00					
	p_s(%)	10	30	50	60	70	90	10	30	50	60	70	90	10	30	50	60	70	90
$E[S]$	6.5	5.7	5.7	5.6	5.7	5.7	5.7	5.9	5.8	5.8	5.8	5.8	6.1	6.0	6.0	6.0	5.9	6.0	6.0
$E[SW]$	0	2.2	3.6	4.5	5.1	5.1	5.1	0.8	1.2	1.6	1.9	2.1	5.2	0.3	0.5	0.6	0.7	1.0	1.8
NSD	77.2	80.5	81.1	81.0	81.2	81.2	81.2	80.1	80.4	80.8	80.7	81.0	80.7	79.3	79.5	79.8	79.9	80.2	80.4
	Fixed	t_s=13:00						07:00											
	p_s(%)	10	30	50	60	70	90	10	30	50	60	70	90						
$E[S]$	6.5	6.1	6.1	6.1	6.1	6.0	6.0	6.3	6.4	6.4	6.4	6.4	6.4						
$E[SW]$	0	0.3	0.2	0.2	0.3	0.4	0.7	0.1	0.0	0.0	0.0	0.0	0.0						
NSD	77.2	79.0	78.9	79.4	79.3	79.4	79.8	78.2	77.4	77.5	77.5	77.4	77.5						
System 2	Fixed	t_s=16:00						15:00						14:00					
	p_s(%)	10	30	50	60	70	90	10	30	50	60	70	90	10	30	50	60	70	90
$E[S]$	5.8	5.2	5.2	5.3	5.2	5.2	5.2	5.4	5.3	5.4	5.3	5.3	5.6	5.5	5.5	5.4	5.5	5.4	5.4
$E[SW]$	0	1.6	2.9	1.1	4.6	4.6	4.6	0.4	0.7	0.4	1.3	1.5	4.8	0.1	0.2	0.3	0.1	0.5	1.2
NSD	79.9	82.3	82.9	82.1	82.9	82.9	82.9	81.6	81.8	81.6	82.5	82.7	82.4	81.0	81.4	81.7	81.1	82.0	82.4
	Fixed	t_s=13:00						07:00											
	p_s(%)	10	30	50	60	70	90	10	30	50	60	70	90						
$E[S]$	5.8	5.5	5.6	5.5	5.7	5.5	5.4	5.6	5.7	5.7	5.6	5.7	5.7						
$E[SW]$	0	0.1	0.1	0.1	0.0	0.2	0.4	0.0	0.0	0.0	3.9	0.0	0.0						
NSD	79.9	81.1	80.9	81.1	80.0	81.3	81.8	80.6	80.0	80.2	80.1	80.0	80.1						

Table 2: Single flush of the System 3 and 4

System 3	Fixed	t_s=16:00						15:00						14:00					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	4.18	4.13	4.11	4.10	4.09	4.09	4.09	4.16	4.15	4.14	4.14	4.13	4.15	4.17	4.16	4.16	4.16	4.16	4.14
Avg. SW	0.00	0.45	1.09	1.80	2.30	2.30	2.30	0.06	0.14	0.27	0.37	0.47	2.37	0.01	0.02	0.05	0.06	0.10	0.33
NSD	86.94	87.18	87.37	87.55	87.60	87.60	87.60	86.97	87.07	87.25	87.19	87.41	87.69	86.96	86.92	86.97	87.09	87.10	87.28
	Fixed	t_s=13:00						07:00											
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90						
$E[S]$	4.18	4.16	4.17	4.17	4.16	4.16	4.16	4.17	4.18	4.18	4.17	4.18	4.18						
Avg. SW	0.00	0.01	0.01	0.01	0.07	0.02	0.07	0.00	0.00	0.00	0.01	0.00	0.00						
NSD	86.94	86.95	86.93	86.99	87.06	86.94	87.09	86.94	86.94	86.94	86.90	86.94	86.94						
System 4	Fixed	t_s=16:00						15:00						14:00					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	3.99	3.97	3.96	3.94	3.95	3.95	3.95	3.98	3.98	3.97	3.97	3.97	3.98	3.99	3.99	3.98	3.98	3.98	3.97
Avg. SW	0.00	0.21	0.67	1.34	1.85	1.85	1.85	0.02	0.05	0.11	0.16	0.24	1.88	0.00	0.01	0.01	0.02	0.03	0.14
NSD	87.64	87.68	87.92	88.08	88.15	88.15	88.15	87.65	87.70	87.66	87.73	87.86	88.20	87.65	87.66	87.68	87.67	87.67	87.78
	Fixed	t_s=13:00						07:00											
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90						
$E[S]$	3.99	3.99	3.98	3.99	3.98	3.98	3.98	3.99	3.99	3.99	3.98	3.99	3.99						
Avg. SW	0.00	0.00	0.00	0.00	0.02	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00						
NSD	87.64	87.65	87.65	87.65	87.68	87.66	87.69	87.66	87.64	87.64	87.66	87.64	87.64						

Table 3: Single flush of the System 5 and 6

System 5	Fixed	t_s=15:00						14:00						13:00					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	7.59	7.46	7.39	7.30	7.27	7.27	7.27	7.53	7.49	7.44	7.39	7.32	7.27	7.58	7.54	7.52	7.49	7.44	7.33
Avg. SW	0.00	0.67	1.77	4.75	7.36	7.36	7.36	0.21	0.41	0.82	1.46	3.27	7.36	0.06	0.14	0.26	0.39	0.60	6.34
NSD	76.68	77.18	77.74	78.21	78.38	78.38	78.38	76.90	76.99	77.70	77.59	78.35	78.38	76.59	76.82	77.09	77.37	77.47	78.36
	Fixed	t_s=12:00						07:00											
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90						
$E[S]$	7.59	7.58	7.56	7.56	7.52	7.50	7.40	7.61	7.59	7.59	7.58	7.61	7.59						
Avg. SW	0.00	0.02	0.04	0.07	0.10	0.19	1.05	0.00	0.01	0.00	0.00	0.00	0.01						
NSD	76.68	76.70	76.60	76.65	76.76	77.00	77.90	76.66	76.64	76.70	76.67	76.66	76.67						
System 6	Fixed	t_s=15:00						14:00						13:00					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	7.36	7.31	7.24	7.17	7.13	7.13	7.13	7.34	7.33	7.28	7.24	7.20	7.13	7.35	7.34	7.33	7.31	7.29	7.18
Avg. SW	0.00	0.32	0.99	3.47	6.13	6.13	6.13	0.07	0.17	0.42	0.80	2.17	6.13	0.01	0.04	0.09	0.15	0.28	5.07
NSD	77.67	77.83	78.16	78.61	78.92	78.92	78.92	77.71	77.77	77.92	78.25	78.46	78.92	77.66	77.73	77.72	77.77	77.87	78.88
	Fixed	t_s=12:00						07:00											
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90						
$E[S]$	7.36	7.35	7.36	7.35	7.34	7.33	7.26	7.36	7.36	7.36	7.36	7.36	7.36						
Avg. SW	0.00	0.00	0.01	0.02	0.03	0.07	0.56	0.00	0.00	0.00	0.00	0.00	0.00						
NSD	77.67	77.67	77.65	77.66	77.75	77.78	78.05	77.67	77.67	77.67	77.67	77.67	77.67						

Table 4: Single flush of the System 7 and 8

System 7	Fixed	t_s=15:00						14:00						13:00					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	6.32	6.32	6.32	6.32	6.31	6.31	6.31	6.32	6.32	6.32	6.32	6.32	6.31	6.32	6.32	6.32	6.32	6.32	6.32
Avg. SW	0.00	0.00	0.03	0.33	1.07	1.07	1.07	0.00	0.00	0.01	0.02	0.12	1.07	0.00	0.00	0.00	0.00	0.00	0.73
NSD	81.82	81.82	81.85	81.91	81.97	81.97	81.97	81.82	81.82	81.83	81.84	81.87	81.97	81.82	81.82	81.82	81.82	81.81	81.97
	Fixed	t_s=12:00						07:00											
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90						
$E[S]$	6.32	6.32	6.32	6.32	6.32	6.32	6.32	6.32	6.32	6.32	6.32	6.32	6.32	6.319	6.314	6.314	6.314	6.314	6.314
Avg. SW	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.19	1.052	1.052	1.052	1.052	1.052
NSD	81.82	81.82	81.82	81.82	81.82	81.82	81.82	81.82	81.82	81.82	81.82	81.82	81.82	81.851	81.945	81.945	81.945	81.945	81.945
System 8	Fixed	t_s=15:00						14:00						13:00					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	6.33	6.33	6.33	6.33	6.32	6.32	6.32	6.33	6.33	6.33	6.33	6.33	6.32	6.33	6.33	6.33	6.33	6.33	6.33
Avg. SW	0.00	0.00	0.01	0.00	0.81	0.81	0.81	0.00	0.00	0.00	0.00	0.05	0.81	0.00	0.00	0.00	0.00	0.00	0.49
NSD	81.94	81.94	81.94	81.94	82.05	82.05	82.05	81.94	81.94	81.94	81.94	81.95	82.05	81.94	81.94	81.94	81.94	81.94	81.96
	Fixed	t_s=12:00						07:00											
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90						
$E[S]$	6.33	6.33	6.33	6.33	6.33	6.33	6.33	6.33	6.33	6.33	6.33	6.33	6.33	6.328	6.324	6.324	6.324	6.324	6.324
Avg. SW	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.18	0.00	0.00	0.086	0.816	0.816	0.816	0.816	0.816
NSD	81.94	81.94	81.94	81.94	81.94	81.94	81.95	81.94	81.94	81.94	81.98	81.94	81.94	81.964	82.019	82.019	82.019	82.019	82.019

Table 5: Single flush of the System 9 and 10

System 9	Fixed	t_s=15:00						14:00						13:00					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	19.53	16.37	16.29	16.48	16.51	16.51	16.51	16.75	16.68	16.56	16.66	17.13	17.99	17.03	16.97	16.93	16.97	17.04	17.21
Avg. SW	0.00	2.03	2.98	4.93	5.47	5.47	5.47	1.18	1.10	2.31	2.81	3.85	5.46	0.72	1.01	1.58	1.60	2.00	5.41
NSD	70.33	79.84	80.64	80.47	80.40	80.40	80.40	79.50	79.69	80.14	80.77	79.62	78.54	79.17	79.09	79.38	78.93	79.99	78.98
	Fixed	t_s=12:00						07:00						02:00					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	19.53	17.26	17.13	17.16	17.11	17.22	18.86	18.21	18.02	17.86	17.87	18.23	17.80	18.88	18.69	18.57	18.57	18.50	18.24
Avg. SW	0.00	0.49	0.66	1.01	1.00	1.24	2.58	0.10	0.13	0.17	0.18	0.30	0.33	0.03	0.04	0.04	0.05	0.06	0.08
NSD	70.33	77.34	78.81	78.24	78.18	78.66	77.14	74.83	74.99	75.97	75.89	75.41	76.11	72.27	73.23	73.73	73.81	74.23	74.26
System 10	Fixed	t_s=15:00						14:00						13:00					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	17.83	15.27	15.18	15.40	15.48	15.48	15.48	15.67	15.53	15.45	15.63	15.88	17.09	15.86	15.67	15.71	15.69	15.81	15.91
Avg. SW	0.00	1.47	2.30	4.26	4.85	4.85	4.85	0.78	1.16	1.72	2.15	3.10	4.81	0.44	0.66	1.07	1.10	1.44	4.83
NSD	68.58	77.81	77.74	78.51	77.21	77.21	77.21	75.84	76.96	77.19	76.69	76.45	74.39	74.86	75.38	76.94	76.67	76.36	76.33
	Fixed	t_s=12:00						07:00						02:00					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	17.83	16.08	15.95	15.84	15.85	15.92	17.33	16.92	16.74	16.56	16.56	16.71	16.35	17.42	17.26	17.20	17.14	17.01	16.86
Avg. SW	0.00	0.29	0.40	0.64	0.65	0.82	1.94	0.05	0.06	0.09	0.09	0.14	0.18	0.01	0.02	0.02	0.02	0.02	0.04
NSD	68.58	74.11	74.93	75.84	75.76	75.31	73.60	71.04	71.60	72.52	72.61	71.76	73.17	69.35	70.22	70.14	70.32	70.70	71.18

Table 6: Single flush of the System 11 and 12

System 11	Fixed	t_s=15:00						14:00						13:00					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	11.10	10.96	10.88	10.83	10.81	10.81	10.81	11.01	11.02	10.92	10.89	10.85	10.84	11.05	11.01	10.97	10.96	10.93	10.84
Avg. SW	0.00	0.40	0.84	1.92	2.34	2.34	2.34	0.14	0.12	0.58	0.79	1.32	2.32	0.05	0.12	0.28	0.30	0.46	2.32
NSD	68.80	69.25	69.76	70.25	70.31	70.31	70.31	69.13	69.11	69.68	69.98	70.30	70.63	68.94	69.14	69.36	69.46	69.71	70.63
	Fixed	t_s=12:00						07:00						02:00					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	11.10	11.07	11.05	11.01	11.01	10.99	10.91	11.10	11.10	11.10	11.10	11.10	11.08	11.10	11.10	11.10	11.10	11.10	11.10
Avg. SW	0.00	0.02	0.05	0.12	0.12	0.19	0.73	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
NSD	68.80	68.94	69.00	69.24	69.27	69.22	69.87	68.80	68.80	68.79	68.79	68.83	68.90	68.80	68.80	68.80	68.80	68.80	68.80
System 12	Fixed	t_s=15:00						14:00						13:00					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	10.78	10.72	10.67	10.62	10.61	10.61	10.61	10.75	10.72	10.69	10.66	10.64	10.63	10.77	10.75	10.72	10.72	10.69	10.66
Avg. SW	0.00	0.20	0.53	1.47	1.90	1.90	1.90	0.05	0.14	0.34	0.50	0.93	1.89	0.01	0.04	0.13	0.14	0.25	1.90
NSD	69.89	70.08	70.41	70.90	70.97	70.97	70.97	69.93	70.18	70.37	70.60	70.89	71.14	69.93	70.05	70.15	70.14	70.33	71.13
	Fixed	t_s=12:00						07:00						02:00					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	10.78	10.77	10.76	10.75	10.75	10.74	10.68	10.78	10.78	10.78	10.78	10.78	10.77	10.78	10.78	10.78	10.78	10.78	10.78
Avg. SW	0.00	0.01	0.01	0.04	0.05	0.08	0.44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
NSD	69.89	69.92	69.96	70.06	70.06	70.09	70.58	69.89	69.89	69.89	69.89	69.89	69.91	69.89	69.89	69.89	69.89	69.89	69.89

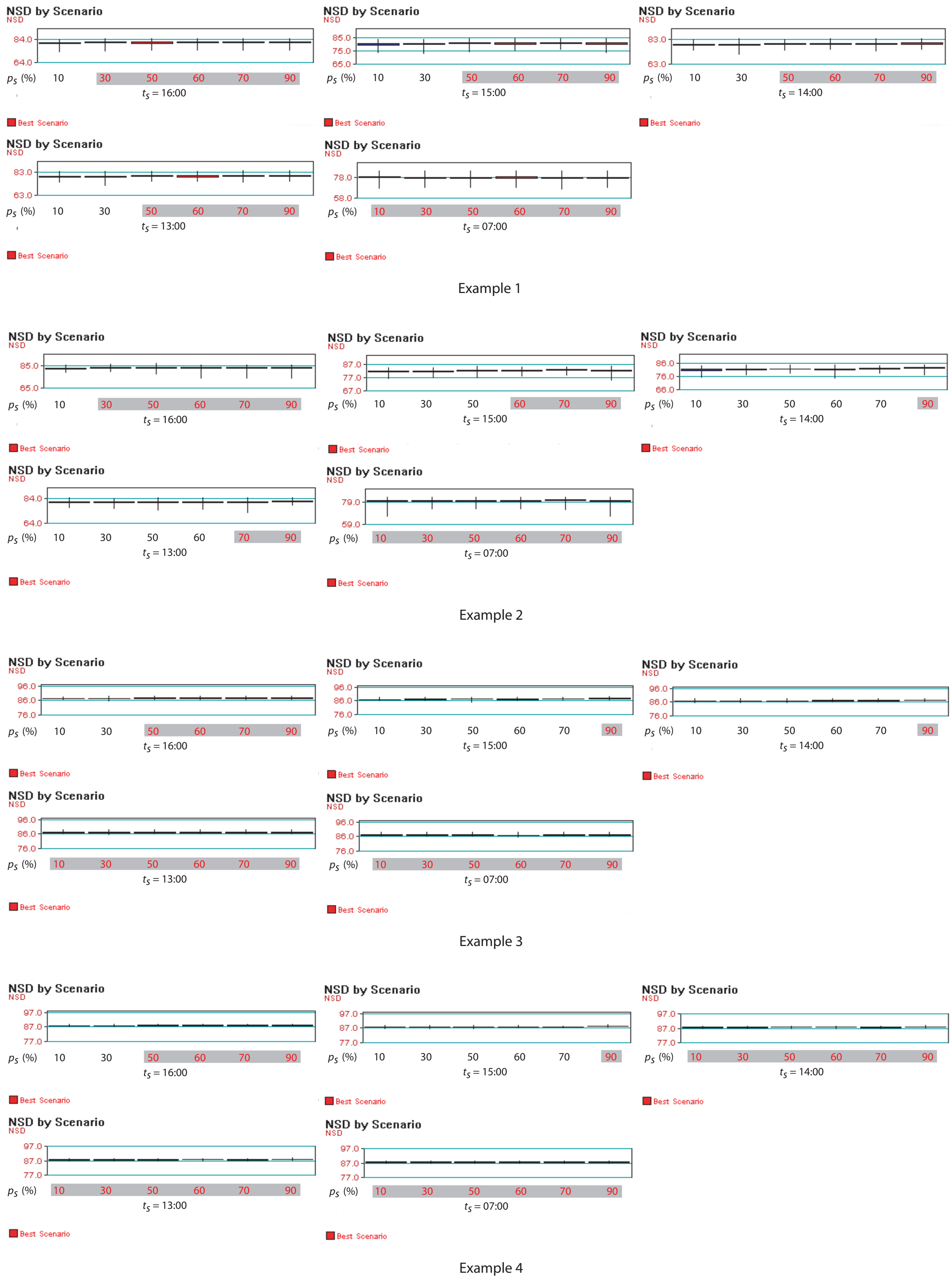


Figure 1: Best scenarios based on box and whisker chart of System 1∼4 using SF

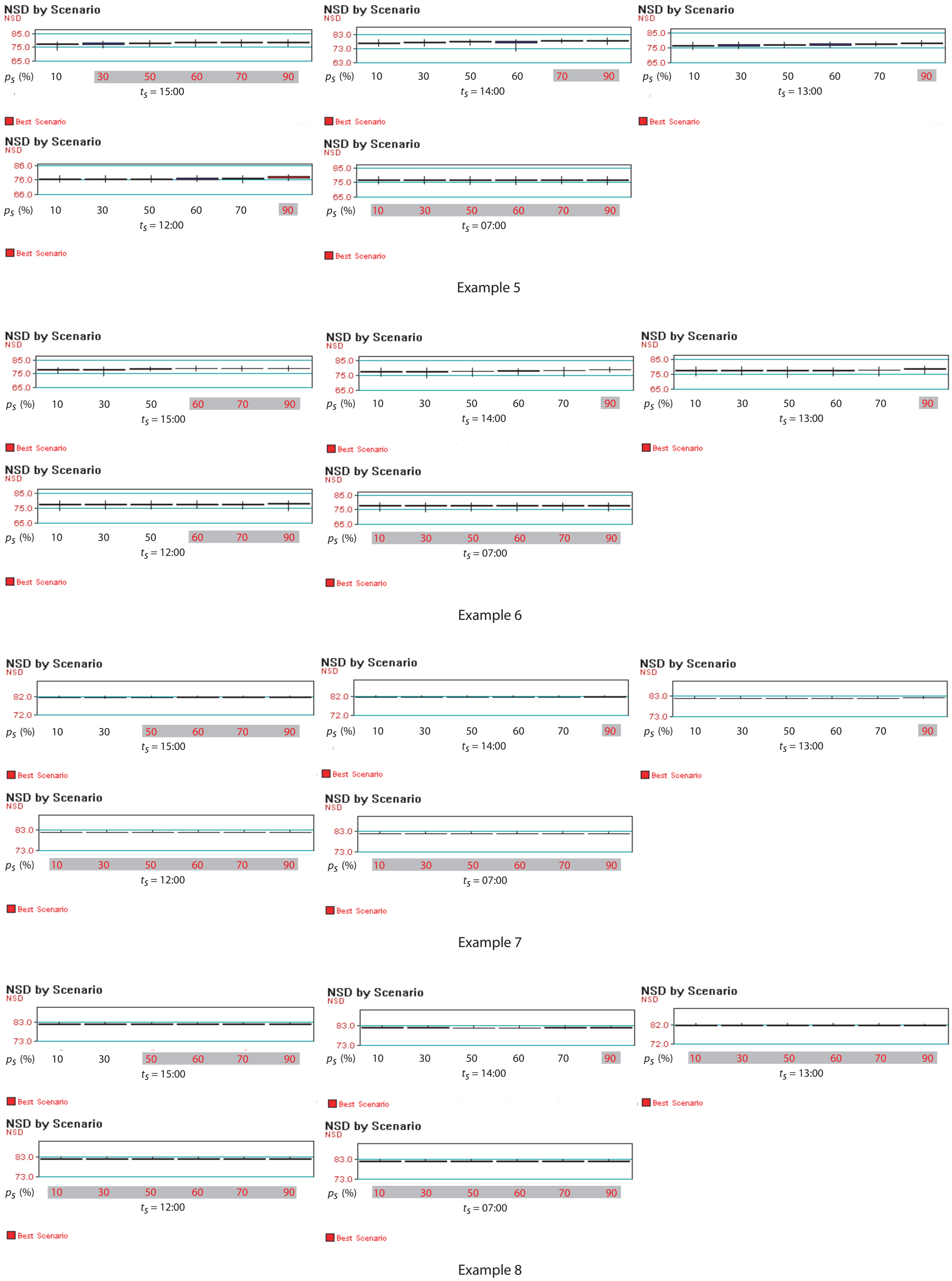
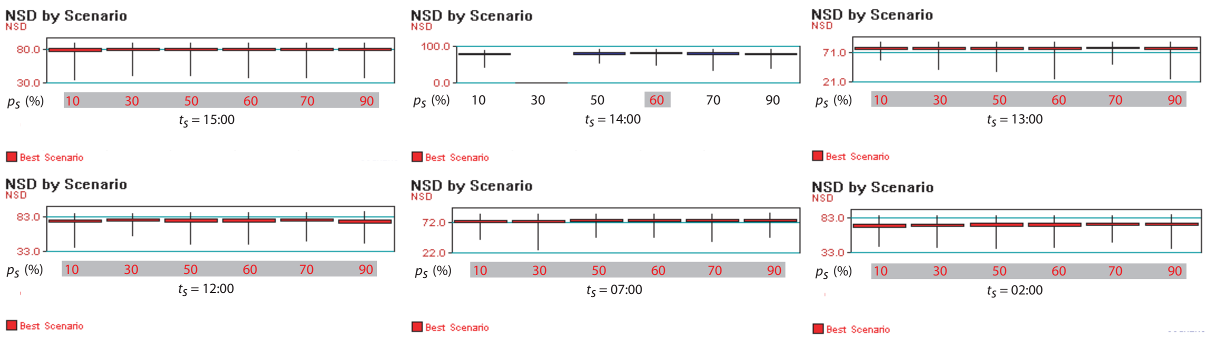
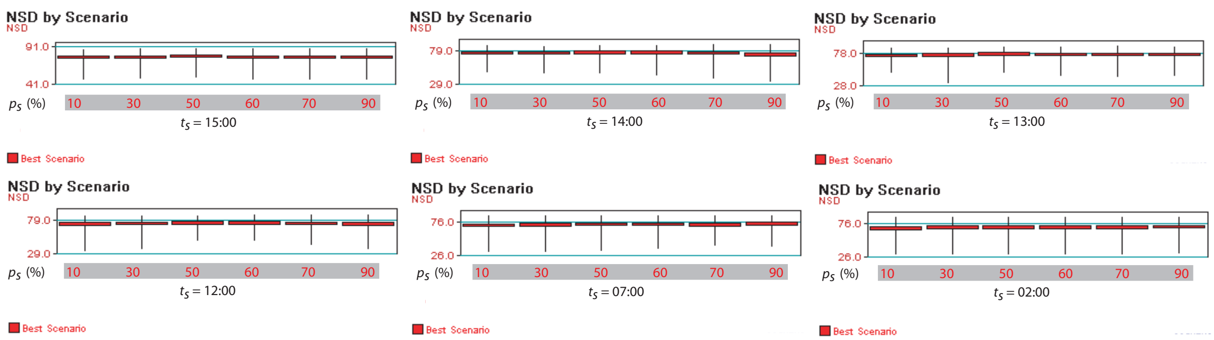


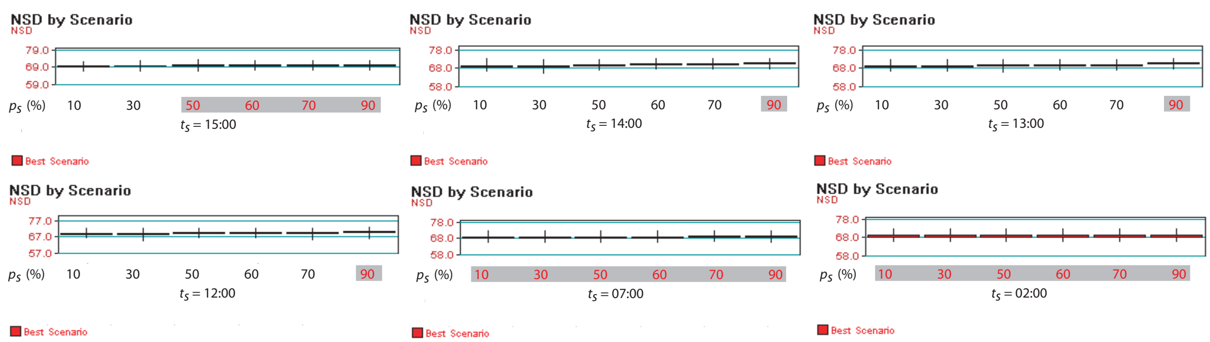
Figure 2: Best scenarios based on box and whisker chart of System 5∼8 using SF



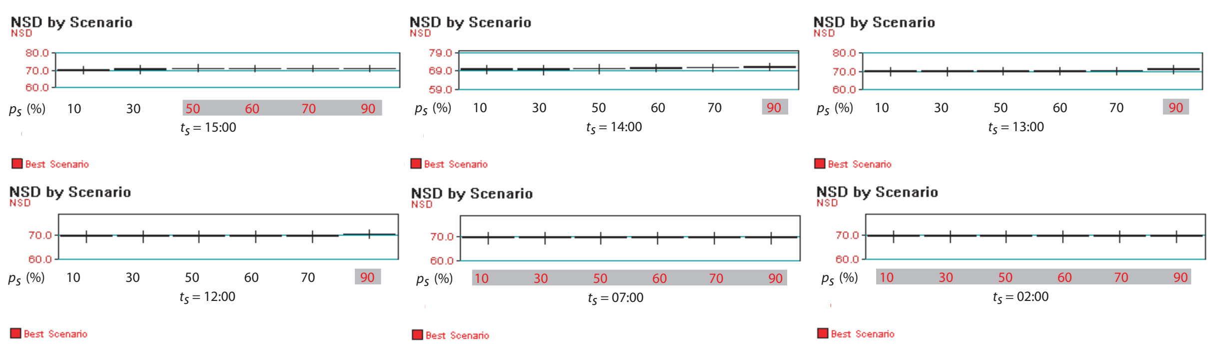
Example 9



Example 10



Example 11



Example 12

Figure 3: Best scenarios based on box and whisker chart of System 9∼12 using SF

Table 7: Multi-flush of the System 1 and 2

System 1	Fixed	2 Times (t_s=16:00, 15:00)						3 Times (t_s=16:00, 15:00, 14:00)						4 Times (t_s=16:00, 15:00, 14:00, 03:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	6.54	5.73	5.68	5.65	5.68	5.68	5.68	5.69	5.65	5.65	5.64	5.64	5.64	5.66	5.62	5.62	5.64	5.64	5.64
Avg. SW	0.00	2.73	3.89	5.40	6.20	6.20	6.20	3.06	4.32	6.16	7.04	7.04	7.04	3.30	4.70	6.79	7.89	7.89	7.89
NSD	77.26	81.42	82.01	82.26	82.40	82.40	82.40	81.61	82.30	82.73	82.77	82.77	82.77	81.99	82.83	82.75	82.65	82.65	82.65

	Fixed	5 Times (t_s=16:00, 15:00, 14:00, 13:00, 12:00)						6 Times (t_s=16:00, 15:00, 14:00, 13:00, 12:00, 11:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	6.54	5.64	5.62	5.59	5.61	5.61	5.61	5.62	5.56	5.57	5.63	5.63	5.63
Avg. SW	0.00	3.52	5.04	7.34	8.68	8.68	8.68	3.70	5.33	7.94	9.46	9.46	9.46
NSD	77.26	81.83	82.30	82.45	82.66	82.66	82.66	82.28	82.35	82.46	82.49	82.49	82.49

System 2	Fixed	2 Times (t_s=16:00, 15:00)						3 Times (t_s=16:00, 15:00, 14:00)						4 Times (t_s=16:00, 15:00, 14:00, 03:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	5.81	5.26	5.21	5.19	5.20	5.20	5.20	5.24	5.18	5.17	5.20	5.20	5.20	5.22	5.16	5.17	5.19	5.19	5.19
Avg. SW	0.00	1.95	3.02	4.64	5.47	5.47	5.47	2.15	3.34	5.21	6.24	6.24	6.24	2.33	3.59	5.75	6.95	6.95	6.95
NSD	79.99	82.91	83.48	83.81	84.14	84.14	84.14	83.24	83.71	84.27	84.12	84.12	84.12	83.32	83.75	84.27	83.99	83.99	83.99

	Fixed	5 Times (t_s=16:00, 15:00, 14:00, 13:00, 12:00)						6 Times (t_s=16:00, 15:00, 14:00, 13:00, 12:00, 11:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	5.81	5.22	5.16	5.15	5.18	5.18	5.18	5.20	5.13	5.15	5.17	5.17	5.17
Avg. SW	0.00	2.51	3.86	6.27	7.65	7.65	7.65	2.64	4.10	6.82	8.38	8.38	8.38
NSD	79.99	83.28	83.80	84.24	84.04	84.04	84.04	83.27	83.89	83.93	83.90	83.90	83.90

Table 8: Multi-flush of the System 3 and 4

System 3	Fixed	2 Times (t_s=16:00, 15:00)						3 Times (t_s=16:00, 15:00, 14:00)						4 Times (t_s=16:00, 15:00, 14:00, 03:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	4.18	4.12	4.10	4.09	4.09	4.09	4.09	4.12	4.09	4.08	4.08	4.08	4.08	4.11	4.08	4.07	4.07	4.07	4.07
Avg. SW	0.00	0.56	1.16	2.23	2.84	2.84	2.84	0.69	1.37	2.64	3.37	3.37	3.37	0.78	1.56	3.05	3.93	3.93	3.93
NSD	86.94	87.38	87.67	87.83	87.97	87.97	87.97	87.38	87.73	87.88	88.04	88.04	88.04	87.47	87.68	87.92	87.95	87.95	87.95

	Fixed	5 Times (t_s=16:00, 15:00, 14:00, 13:00, 12:00)						6 Times (t_s=16:00, 15:00, 14:00, 13:00, 12:00, 11:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	4.18	4.10	4.08	4.06	4.06	4.06	4.06	4.09	4.07	4.05	4.05	4.05	4.05
Avg. SW	0.00	0.87	1.75	3.42	4.49	4.49	4.49	0.96	1.93	3.87	5.04	5.04	5.04
NSD	86.94	87.35	87.76	87.97	87.92	87.92	87.92	87.49	87.73	87.87	88.00	88.00	88.00

System 4	Fixed	2 Times (t_s=16:00, 15:00)						3 Times (t_s=16:00, 15:00, 14:00)						4 Times (t_s=16:00, 15:00, 14:00, 03:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	3.99	3.97	3.95	3.94	3.94	3.94	3.94	3.96	3.95	3.93	3.93	3.93	3.93	3.96	3.94	3.93	3.93	3.93	3.93
Avg. SW	0	0.29	0.73	1.71	2.35	2.35	2.35	0.35	0.87	2.07	2.80	2.80	2.80	0.40	1.00	2.44	3.34	3.34	3.34
NSD	87.64	87.76	88.04	88.34	88.43	88.43	88.43	87.93	88.16	88.28	88.38	88.38	88.38	87.94	88.12	88.34	88.40	88.40	88.40

	Fixed	5 Times (t_s=16:00, 15:00, 14:00, 13:00, 12:00)						6 Times (t_s=16:00, 15:00, 14:00, 13:00, 12:00, 11:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	3.99	3.96	3.94	3.92	3.92	3.92	3.92	3.95	3.93	3.92	3.91	3.91	3.91
Avg. SW	0	0.46	1.13	2.79	3.85	3.85	3.85	0.51	1.26	3.16	4.37	4.37	4.37
NSD	87.64	87.83	88.16	88.37	88.35	88.35	88.35	87.90	88.19	88.30	88.37	88.37	88.37

Table 9: Multi-flush of the System 5 and 6

System 5	Fixed	2 Times (t_s=15:00, 13:00)						3 Times (t_s=15:00, 13:00, 11:00)						4 Times (t_s=15:00, 13:00, 11:00, 09:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	7.59	7.46	7.36	7.27	7.25	7.25	7.25	7.43	7.35	7.24	7.22	7.22	7.22	7.42	7.33	7.24	7.22	7.22	7.22
Avg. SW	0.00	0.84	1.99	5.41	8.36	8.36	8.36	0.94	2.24	5.93	9.49	9.49	9.49	1.04	2.46	6.56	10.83	10.83	10.83
NSD	76.68	77.27	77.91	78.45	78.74	78.74	78.74	77.44	77.86	78.60	78.93	78.93	78.93	77.61	77.98	78.77	78.80	78.80	78.80

	Fixed	5 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00)						6 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00, 05:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	7.59	7.41	7.31	7.21	7.18	7.18	7.18	7.41	7.31	7.19	7.17	7.17	7.17
Avg. SW	0.00	1.14	2.63	7.18	12.07	12.07	12.07	1.23	2.82	7.71	13.24	13.24	13.24
NSD	76.68	77.62	77.95	78.73	78.86	78.86	78.86	77.77	78.04	78.65	78.93	78.93	78.93

System 6	Fixed	2 Times (t_s=15:00, 13:00)						3 Times (t_s=15:00, 13:00, 11:00)						4 Times (t_s=15:00, 13:00, 11:00, 09:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	7.36	7.30	7.23	7.14	7.11	7.11	7.11	7.28	7.22	7.13	7.10	7.10	7.10	7.28	7.20	7.11	7.08	7.08	7.08
Avg. SW	0.00	0.42	1.16	4.01	7.12	7.12	7.12	0.47	1.30	4.51	8.26	8.26	8.26	0.53	1.46	5.00	9.39	9.39	9.39
NSD	77.67	77.92	78.26	78.99	79.29	79.29	79.29	77.90	78.36	78.92	79.21	79.21	79.21	78.05	78.52	79.02	79.29	79.29	79.29

	Fixed	5 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00)						6 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00, 05:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	7.36	7.28	7.20	7.10	7.06	7.06	7.06	7.27	7.19	7.08	7.05	7.05	7.05
Avg. SW	0.00	0.56	1.58	5.42	10.53	10.53	10.53	0.61	1.68	5.91	11.66	11.66	11.66
NSD	77.67	78.00	78.57	79.04	79.29	79.29	79.29	78.04	78.52	79.02	79.30	79.30	79.30

Table 10: Multi-flush of the System 7 and 8

System 7	Fixed	2 Times (t_s=15:00, 13:00)						3 Times (t_s=15:00, 13:00, 11:00)						4 Times (t_s=15:00, 13:00, 11:00, 09:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	6.32	6.32	6.32	6.32	6.31	6.31	6.31	6.32	6.32	6.31	6.31	6.31	6.31	6.32	6.32	6.31	6.30	6.30	6.30
Avg. SW	0.00	0.00	0.04	0.49	1.54	1.54	1.54	0.01	0.06	0.66	2.03	2.03	2.03	0.01	0.07	0.81	2.50	2.50	2.50
NSD	81.82	81.83	81.84	81.95	82.03	82.03	82.03	81.83	81.88	82.01	82.06	82.06	82.06	81.83	81.90	81.97	82.01	82.01	82.01

	Fixed	5 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00)						6 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00, 05:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	6.32	6.32	6.32	6.31	6.30	6.30	6.30	6.32	6.32	6.31	6.30	6.30	6.30
Avg. SW	0.00	0.01	0.09	0.95	3.01	3.01	3.01	0.01	0.10	1.08	3.48	3.48	3.48
NSD	81.82	81.82	81.89	81.98	82.05	82.05	82.05	81.83	81.87	81.96	82.03	82.03	82.03

System 8	Fixed	2 Times (t_s=15:00, 13:00)						3 Times (t_s=15:00, 13:00, 11:00)						4 Times (t_s=15:00, 13:00, 11:00, 09:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	6.33	6.33	6.33	6.33	6.32	6.32	6.32	6.33	6.33	6.32	6.32	6.32	6.32	6.33	6.33	6.32	6.32	6.32	6.32
Avg. SW	0.00	0.00	0.01	0.29	1.26	1.26	1.26	0.00	0.02	0.39	1.70	1.70	1.70	0.00	0.02	0.49	2.15	2.15	2.15
NSD	81.94	81.95	81.93	81.97	82.06	82.06	82.06	81.95	81.93	82.01	82.05	82.05	82.05	81.94	81.94	82.01	82.06	82.06	82.06

	Fixed	5 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00)						6 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00, 05:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	6.33	6.33	6.33	6.32	6.31	6.31	6.31	6.33	6.33	6.32	6.31	6.31	6.31
Avg. SW	0.00	0.00	0.02	0.60	2.61	2.61	2.61	0.00	0.03	0.68	3.04	3.04	3.04
NSD	81.94	81.94	81.93	82.05	82.06	82.06	82.06	81.94	81.93	81.99	82.03	82.03	82.03

Table 11: Multi-flush of the System 9 and 10

System 9	Fixed	2 Times (t_s=15:00, 13:00)						3 Times (t_s=15:00, 13:00, 11:00)						4 Times (t_s=15:00, 13:00, 11:00, 09:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	19.53	16.33	16.21	16.53	16.56	16.56	16.56	16.23	16.21	16.42	16.61	16.61	16.61	16.23	16.24	16.61	16.71	16.71	16.71
Avg. SW	0.00	2.28	3.30	5.35	5.97	5.97	5.97	2.51	3.59	5.86	6.66	6.66	6.66	2.68	3.86	6.41	7.34	7.34	7.34
NSD	70.33	79.96	80.42	80.71	81.09	81.09	81.09	82.01	81.56	81.53	80.43	80.43	80.43	81.40	80.99	80.52	81.32	81.32	81.32

	Fixed	5 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00)						6 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00, 05:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	19.53	16.17	16.11	16.53	16.70	16.70	16.70	16.09	16.11	16.52	16.64	16.64	16.64
Avg. SW	0.00	2.85	4.11	6.93	7.99	7.99	7.99	2.98	4.38	7.42	8.66	8.66	8.66
NSD	70.33	81.16	82.11	79.66	79.73	79.73	79.73	81.79	81.26	80.94	81.03	81.03	81.03

System 10	Fixed	2 Times (t_s=15:00, 13:00)						3 Times (t_s=15:00, 13:00, 11:00)						4 Times (t_s=15:00, 13:00, 11:00, 09:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	17.83	15.22	15.14	15.35	15.55	16.56	16.56	15.17	15.12	15.37	15.57	16.61	16.61	15.16	15.08	15.44	15.73	16.71	16.71
Avg. SW	0.00	1.68	2.61	4.66	5.34	5.97	5.97	1.87	2.86	5.15	6.01	6.66	6.66	2.01	3.11	5.64	6.68	7.34	7.34
NSD	68.58	78.96	78.47	79.67	77.49	81.09	81.09	78.79	77.87	78.14	77.89	80.43	80.43	78.17	79.11	78.09	78.28	81.32	81.32

	Fixed	5 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00)						6 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00, 05:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	17.83	15.16	15.07	15.48	15.73	16.70	16.70	15.08	15.06	15.54	15.87	16.64	16.64
Avg. SW	0.00	2.12	3.32	6.15	7.33	7.99	7.99	2.23	3.50	6.60	7.96	8.66	8.66
NSD	68.58	78.59	78.93	77.64	76.77	79.73	79.73	78.23	79.06	77.44	77.00	81.03	81.03

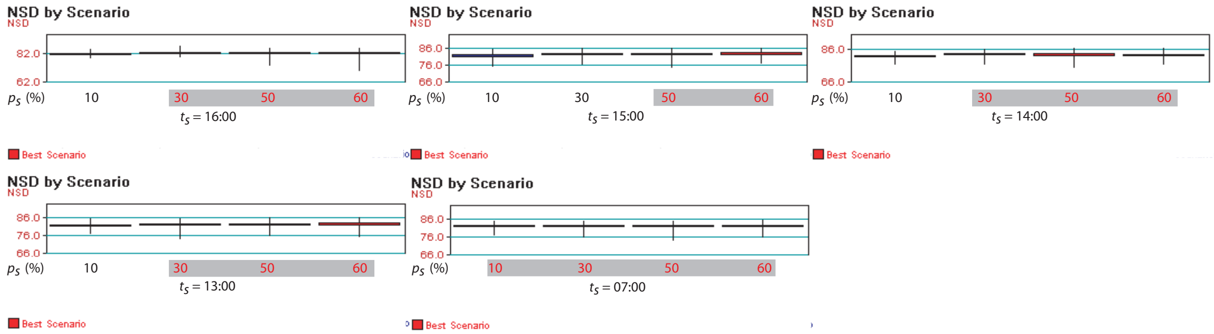
Table 12: Multi-flush of the System 11 and 12

System 11	Fixed	2 Times (t_s=15:00, 13:00)						3 Times (t_s=15:00, 13:00, 11:00)						4 Times (t_s=15:00, 13:00, 11:00, 09:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	11.10	10.92	10.86	10.79	10.78	10.78	10.78	10.90	10.82	10.76	10.74	10.74	10.74	10.87	10.80	10.72	10.71	10.71	10.71
Avg. SW	0.00	0.27	0.69	1.83	2.37	2.37	2.37	0.33	0.84	2.19	2.86	2.86	2.86	0.38	0.97	2.56	3.38	3.38	3.38
NSD	68.80	69.51	70.17	70.77	70.73	70.73	70.73	69.77	70.42	70.97	70.89	70.89	70.89	69.83	70.28	70.89	70.96	70.96	70.96

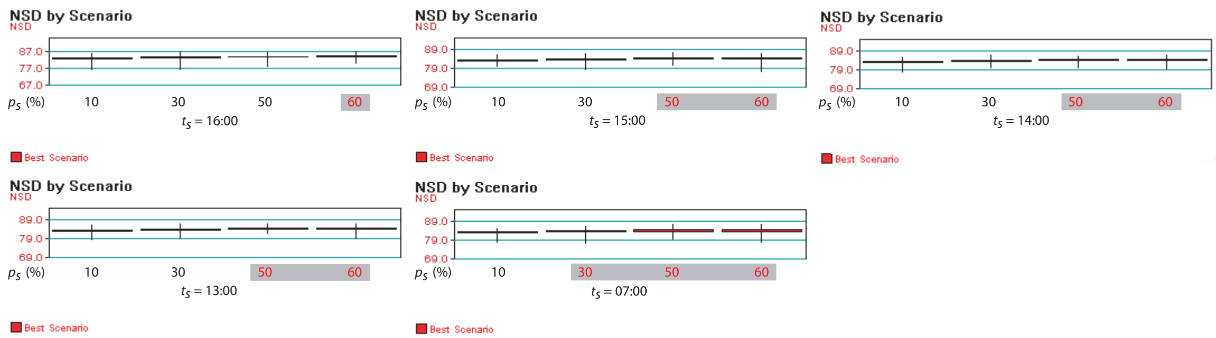
	Fixed	5 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00)						6 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00, 05:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	11.10	10.86	10.77	10.69	10.68	10.68	10.68	10.85	10.75	10.66	10.65	10.65	10.65
Avg. SW	0.00	0.43	1.09	2.92	3.89	3.89	3.89	0.48	1.22	3.27	4.37	4.37	4.37
NSD	68.80	69.93	70.42	70.96	70.93	70.93	70.93	69.83	70.35	70.77	70.91	70.91	70.91

System 12	Fixed	2 Times (t_s=15:00, 13:00)						3 Times (t_s=15:00, 13:00, 11:00)						4 Times (t_s=15:00, 13:00, 11:00, 09:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	10.78	10.70	10.65	10.59	10.58	10.58	10.58	10.69	10.62	10.57	10.56	10.56	10.56	10.67	10.61	10.55	10.53	10.53	10.53
Avg. SW	0.00	0.27	0.69	1.83	2.37	2.37	2.37	0.33	0.84	2.19	2.86	2.86	2.86	0.38	0.97	2.56	3.38	3.38	3.38
NSD	69.89	70.37	70.79	71.40	71.55	71.55	71.55	70.39	70.79	71.43	71.50	71.50	71.50	70.51	70.97	71.40	71.52	71.52	71.52

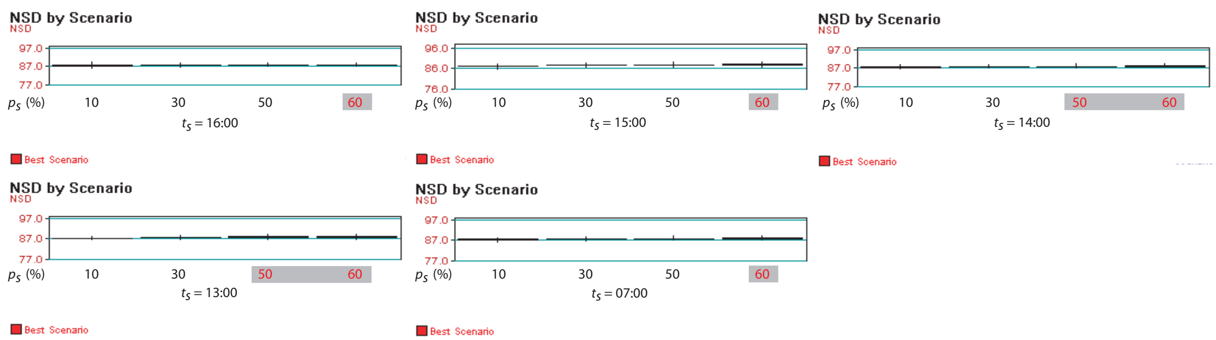
	Fixed	5 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00)						6 Times (t_s=15:00, 13:00, 11:00, 09:00, 07:00, 05:00)					
		p_s=10%	30	50	60	70	90	p_s=10%	30	50	60	70	90
$E[S]$	10.78	10.66	10.59	10.52	10.51	10.51	10.51	10.65	10.58	10.50	10.49	10.49	10.49
Avg. SW	0.00	0.43	1.09	2.92	3.89	3.89	3.89	0.48	1.22	3.27	4.37	4.37	4.37
NSD	69.89	70.56	71.03	71.39	71.54	71.54	71.54	70.56	70.97	71.37	71.51	71.51	71.51



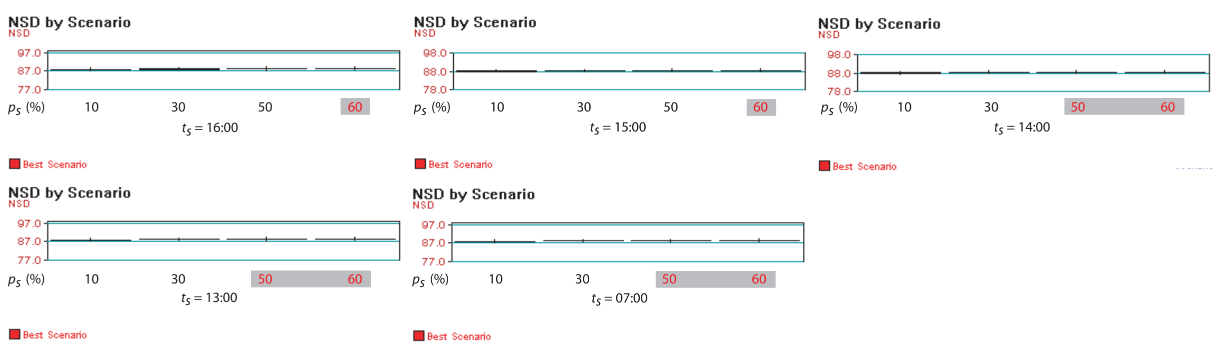
Example 1



Example 2

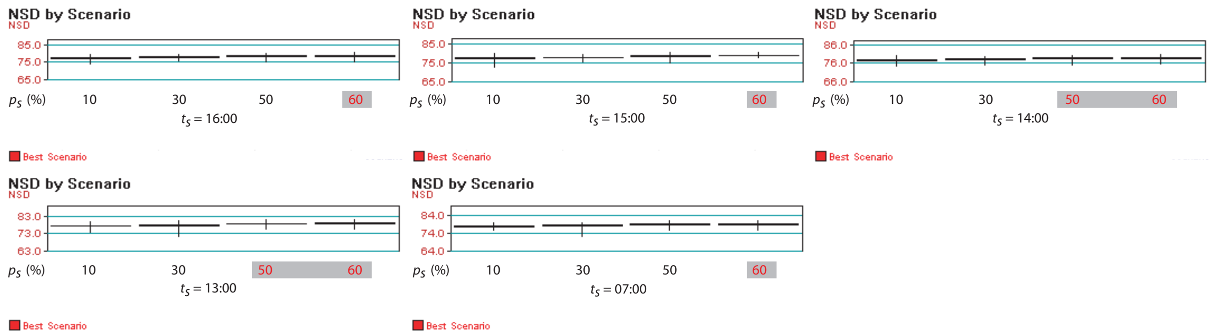


Example 3

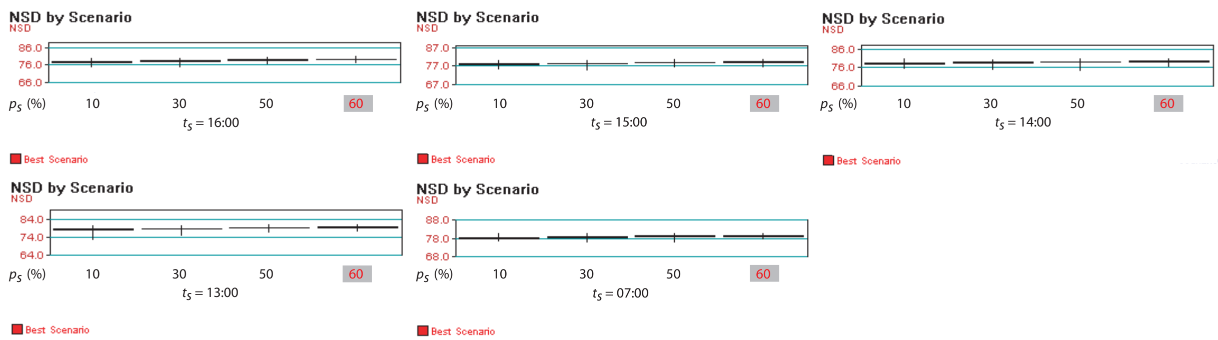


Example 4

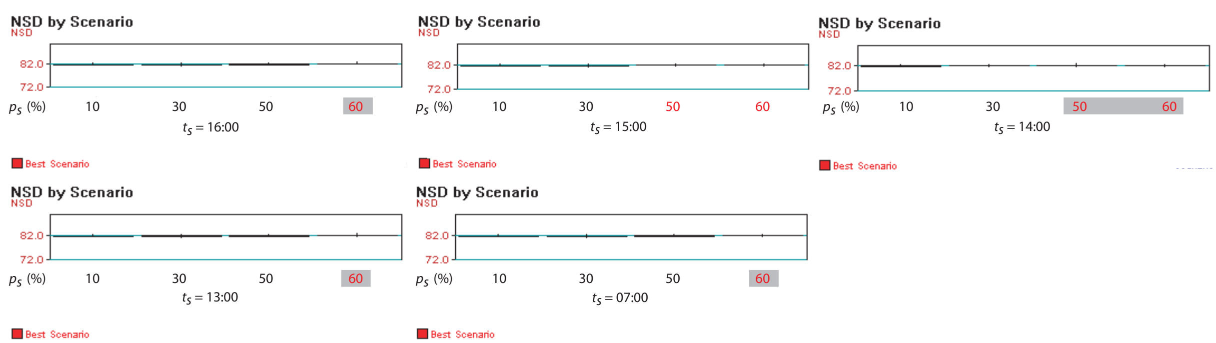
Figure 4: Best scenarios based on box and whisker chart of System 1∼4 using MF



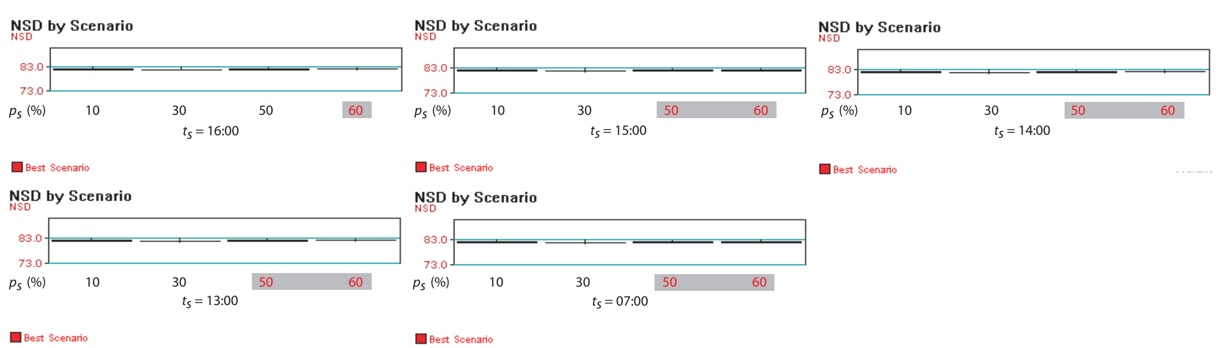
Example 5



Example 6

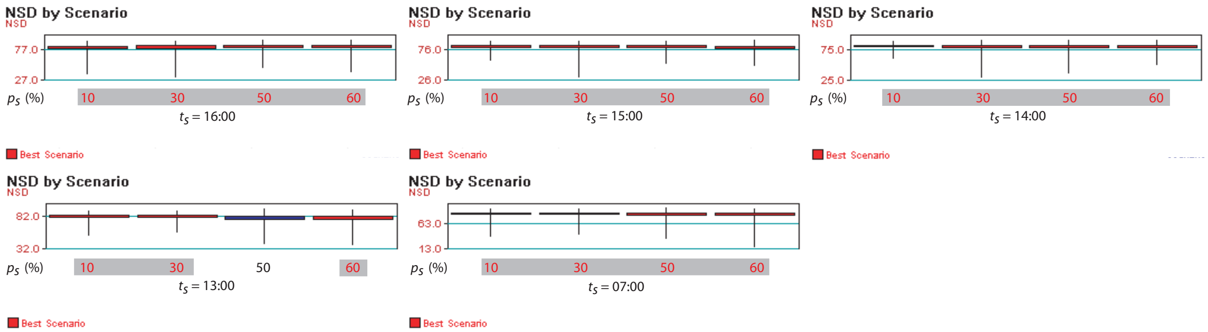


Example 7

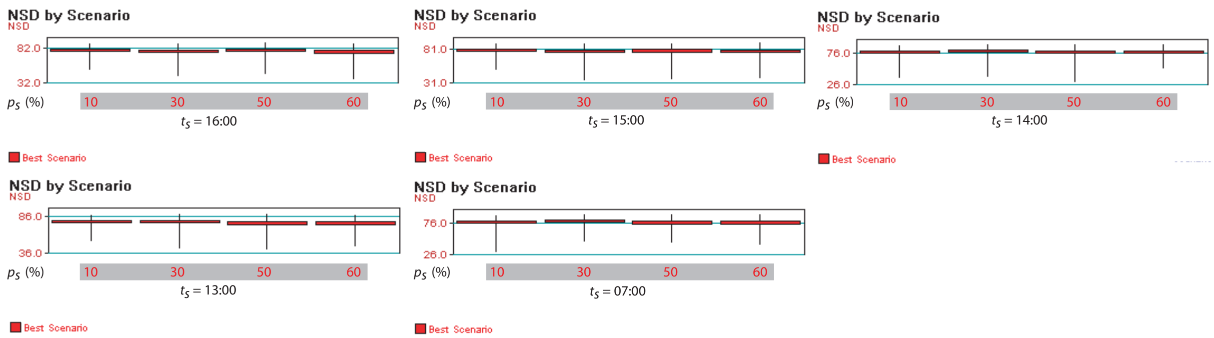


Example 8

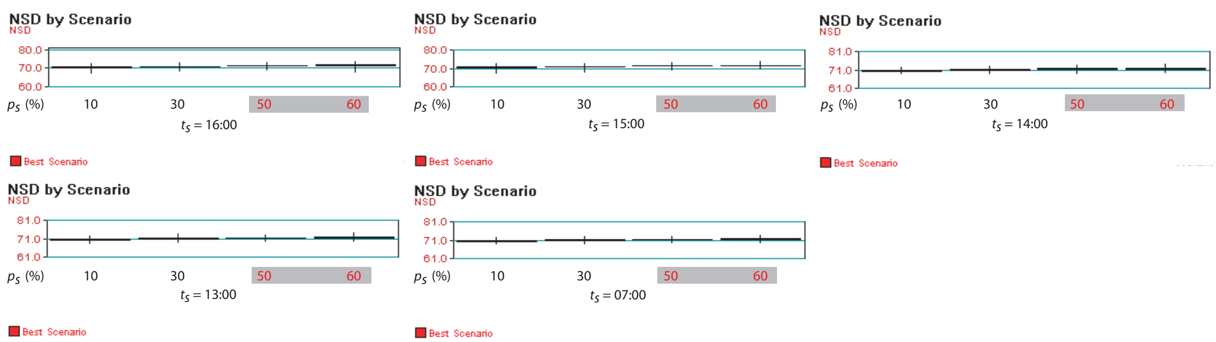
Figure 5: Best scenarios based on box and whisker chart of System 5∼8 using MF



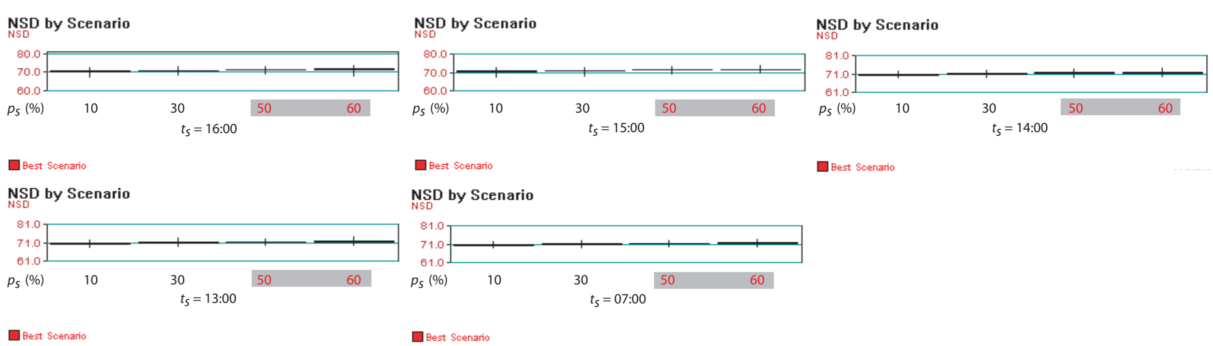
Example 9



Example 10



Example 11



Example 12

Figure 6: Best scenarios based on box and whisker chart of System 9∼12 using MF