Queueing Network Models for Performance Evaluation of Dynamic Multi-Product Manufacturing Systems

by

Girish Jampani Hanumantha

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved January 2020 by the
Graduate Supervisory Committee:

Ronald Askin, Chair
Feng Ju
Hao Yan
Pitu Mirchandani

ARIZONA STATE UNIVERSITY

August 2020

ABSTRACT

Modern manufacturing systems are part of a complex supply chain where customer preferences are constantly evolving. The rapidly evolving market demands manufacturing organizations to be increasingly agile and flexible. Medium term capacity planning for manufacturing systems employ queueing network models based on stationary demand assumptions. However, these stationary demand assumptions are not very practical for rapidly evolving supply chains. Nonstationary demand processes provide a reasonable framework to capture the time-varying nature of modern markets. The analysis of queues and queueing networks with time-varying parameters is mathematically intractable. In this dissertation, heuristics which draw upon existing steady state queueing results are proposed to provide computationally efficient approximations for dynamic multi-product manufacturing systems modeled as time-varying queueing networks with multiple customer classes (product types). This dissertation addresses the problem of performance evaluation of such manufacturing systems.

This dissertation considers the two key aspects of dynamic multi-product manufacturing systems - namely, performance evaluation and optimal server resource allocation. First, the performance evaluation of systems with infinite queueing room and a first-come first-serve service paradigm is considered. Second, systems with finite queueing room and priorities between product types are considered. Finally, the optimal server allocation problem is addressed in the context of dynamic multi-product manufacturing systems. The performance estimates developed in the earlier part of the dissertation are leveraged in a simulated annealing algorithm framework to obtain server resource allocations.

ACKNOWLEDGMENTS

I express my sincere gratitude to everyone who made this dissertation possible.

I am deeply indebted to my advisor and committee chair, Dr. Ronald Askin for his guidance and support throughout my Ph.D. I appreciate his patient mentorship for over five years. I am grateful to my dissertation committee members, Dr. Feng Ju, Dr. Hao Yan, and Dr. Pitu Mirchandani who graciously agreed to serve on my dissertation committee. I am thankful to them for their insightful comments and valuable suggestions for my dissertation.

I also thank the Center for Accelerating Operational Efficiency (CAOE) at ASU and the Department of Homeland Security (DHS) for funding part of my dissertation.

Finally, I express my gratitude to friends and family who were a constant source of support and encouragement. Thank you to Sangdi Lin, Vishwanath Potluri, and Ghazal Shams for their help and guidance along the way. This dissertation wouldn't have been possible without them.

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION AND BACKGROUND

1.1 Introduction

The ability to accurately analyze the performance of a manufacturing system is key to making sound operational and tactical decisions for any manufacturing organization. These analyses help managers understand the impact of tactical decisions such a capacity plans and buffer sizing. Performance analyses also form the basis for due-date setting and scheduling decisions on an operational level. Queueing networks are one of the analytic tools widely used for such analyses. Queueing models allow the decision maker to capture uncertainty in arrival and service processes as well as the flow of material through the manufacturing system. Each machine can be viewed as a server and the staging queue in front of the machine as a waiting area. Each job to be produced in the system is a customer with desired service. Efficient algorithms are available for the analysis of open and closed queueing networks under steady state assumptions and are often employed in practice. However, steady state assumptions do not hold true for most modern manufacturing systems. Varying product mixes and demand create nonstationary arrival processes, equipment, engineering change orders and staffing changes cause nonstationary service. Efficient algorithms to analyze queueing networks under nonstationary conditions are crucial to be able to adapt and plan for such dynamic demand and service conditions. A significant amount of work has been done on performance analysis of single customer class, single service station, nonstationary queues motivated by applications in manufacturing, telecommunications, healthcare, call–center staffing and air traffic management. However, there has been very limited research on analysis of nonstationary, multi-class queueing networks particularly with an emphasis on manufacturing systems. For manufacturing systems, incorporating the dynamic demand and network structure of the system helps understand how congestion evolves over time at different points of the system. Networks also provide a basis for modelling the impact of class priorities and buffer sizing. The objective of this research is to develop computationally efficient algorithms to obtain performance approximations of such multi-class networks.

1.1.1 Impact of Nonstationarity

The need for efficient algorithms for the performance analysis of nonstationary queueing networks is motivated through the following examples.

1.1.1.1 Example 1

Consider a three stage serial line of unit rate servers with nonstationary seasonal Poisson arrivals as shown in Fig. 1.1 with arrival rate intensity, $\lambda(t)$, given by

$$\lambda(t) = 0.8\Big[1 + sin\Big(\frac{2\pi t}{100}\Big)\Big],\ t \in [0, 1000]$$

Note that this system has a time-varying arrival but an overall average utilization $\bar{\rho} = \frac{\bar{\lambda}}{\mu} = 0.8$, where μ is the exponential service rate. Also note that as depicted in Figure 1.1 the second station has two parallel servers each with service rate of 0.5 jobs/min.

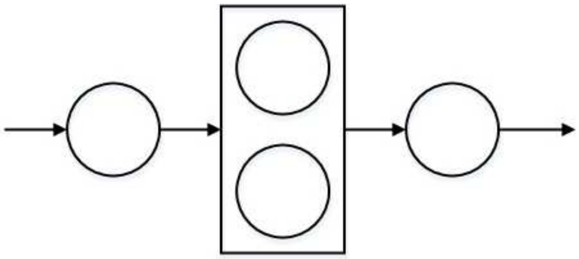


Figure 1.1. Three Stage Serial Line

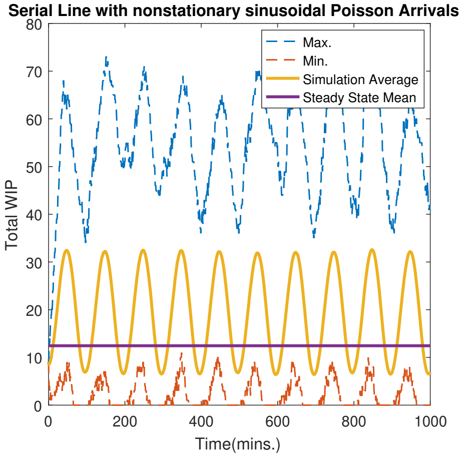


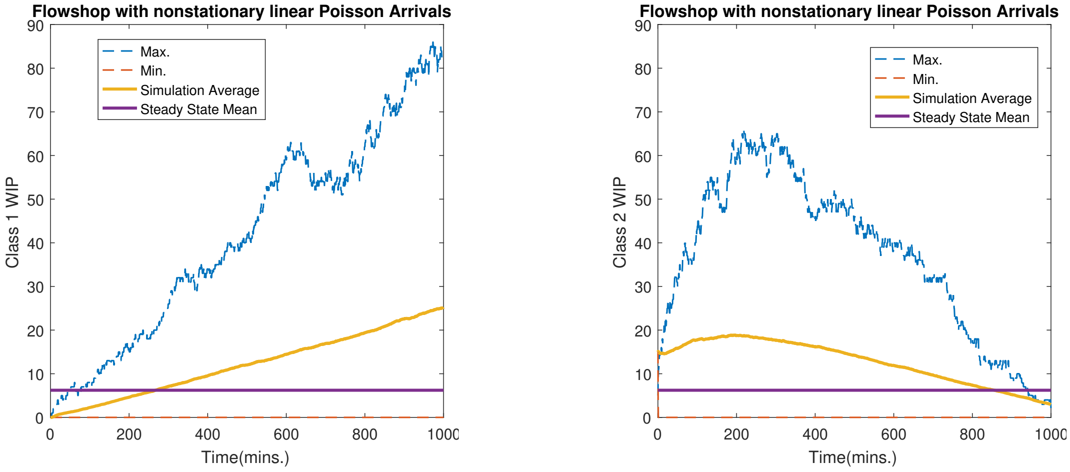
Figure 1.2. Three Stage Serial Line with Sinusoidal Poisson Arrivals

Fig. 1.2 plots the total work-in-process (WIP) in the system for the serial line obtained by 1000 simulation replications each of length 1000 minutes. The figure shows the average WIP level over time along with the maximum and minimum at each point in time across the 1000 replications. The solid horizontal line in the figure indicates WIP levels for a stationary arrival process. Clearly, the sinusoidal pattern of the total WIP is not captured by steady state estimates. Thus, a need exists for models that can track time-varying arrivals as well as other dynamic system properties.

1.1.1.2 Example 2

Flowshops are an ubiquitous class of manufacturing systems where the workstations are placed in series and all product types visit these workstations in the same sequence. Consider a three stage flowshop of unit rate servers with two product classes ($i = 1, 2$) with complementary linear Poisson arrivals with intensities given by

$$\lambda_1(t) = \left(\frac{0.8t}{1000}\right), \ \lambda_2(t) = \left(0.8 - \frac{0.8t}{1000}\right), \ t \in [0, 1000]$$



(a) Class 1 WIP

(b) Class 2 WIP

Figure 1.3. Three Stage Flowshop with Complementary Linear Poisson Arrivals

Fig. 1.3(a) and 1.3(b) plot the total WIP of class 1 and 2 respectively in the flowshop with nonstationary linear Poisson arrivals obtained by 1000 simulation replications.

Comparing the dynamic trajectories to the horizontal steady state values in Fig. 1.2 and Fig. 1.3 shows us that the use of steady state estimates under nonstationary conditions again leads to inaccurate and misleading conclusions. The steady state estimates fail to capture the time-varying nature of the system's performance as arrival conditions vary. Thus, there is a need for computationally efficient performance evaluation methods for systems with time-dependent parameters.

The scope of this research is limited to manufacturing systems that can be analyzed as multi-class queueing networks. The primary goal of the proposed research is to develop analytic approximations for the analysis of dynamic manufacturing systems. In particular, the performance measures of interest are the first-order and second-order moments of throughput rates, work-in-process levels and lead times as a function of dynamic conditions. The focus is primarily on describing performance in response to dynamic demand (order arrival) rates but the ability to adjust service levels to accommodate varying demand is also considered.

1.2 Major Contributions

The major contributions of this dissertation are listed below:

1. Computationally efficient approximations are proposed for the performance evaluation of multi-product dynamic manufacturing systems. The arrival processes are modeled as non-homogeneous Poisson processes and the service time is assumed to be exponentially distributed.
2. Heuristic approximations are presented for lead time estimation in multi-product manufacturing systems which can be used for due-date setting and prioritizing orders.
3. Computationally efficient approximations for incorporating product type priorities and impact of finite buffer space between workstations are developed. Understand-

ing the impact of finite buffers and priorities forms the basis for decisions such as capacity planning and buffer sizing.

4. Optimization schemes are proposed for the allocation of servers in a dynamic multi-class manufacturing system with multiple servers at each workstation. The allocation and reallocation of servers in a dynamic fashion is explored for adapting to the dynamic nature of the arrivals.

Except where otherwise noted, the system is assumed to be "open" in the sense that arriving orders are dispatched to the shop without constrained release control. In that sense, the contributions of this dissertation are particularly relevant to make-to-order systems with time-varying demand processes for products. However, since make-to-stock systems must also match production to consumption, the results can be applied in that case as well.

1.3 Outline of the Dissertation

The rest of this dissertation is organized as follows. The performance analysis of dynamic manufacturing systems with no buffer space restrictions is studied in Chapter 2. The performance analysis of finite buffer systems and the impact of product type priorities is analyzed in Chapter 3. Optimization schemes for allocation and re-allocation of server resources in dynamic manufacturing systems is explored in Chapter 4. Possible future research directions are presented in Chapter 5.

Chapter 2

DYNAMIC MANUFACTURING SYSTEMS WITH INFINITE BUFFERS

This chapter explores the use of queueing network models for the performance evaluation of manufacturing systems under the assumption of infinite buffer space between workstations. There may be multiple product types with each product having its own route, but all products have the same priority. The products are processed in a first-come first-serve scheme at each workstation. These dynamic manufacturing systems can be modeled and analyzed as a nonstationary queueing network with multiple customer classes. However, the exact analysis of nonstationary queues and queueing networks through Markov chain models is mathematically intractable as it involves sums of Bessel functions. Heuristic approaches for the analysis of dynamic manufacturing systems are presented in this chapter. The first part of the chapter presents heuristics based on equivalences between open and closed queueing networks. The latter part of the chapter presents parametric decomposition based incremental approximations for the analysis of multi-product dynamic manufacturing systems. The primary focus is on obtaining reliable first order estimates of WIP levels and throughput rates as a function of time in a computationally efficient way. Numerical analysis of the performance of the proposed heuristics is performed against discrete-event simulations for practically sized flowshop and jobshop instances.

2.1 Problem Description and Mathematical Formulation

Consider a manufacturing system abstracted as an open queueing network with L workstations and R product types (job classes). A system with a general flow pattern defined by the process plans and production volumes of the shop is envisioned. The service rate of workstation l as a function of number of jobs k at the workstation is given by $\mu_l(k)$ (μ_l if static). Let $S(r)$ be the set of workstations visited by part type r. The part types which visit workstation l are denoted by the set $R(l)$. The routing of product type r through the system is given by the matrix $P_r = [p^r_{ij}] \; \forall i \in \{1, \ldots, L\} \; and \; j \in \{1, \ldots, L\}$, indicating the probability a type r job leaving service station i goes to service station j. Let the total observation window be of length T time units which is divided into small time steps. The time step length chosen for approximation is t_s. The instantaneous arrival intensity of part type r at workstation l at time $t, t \in [0, T]$ is given by $\lambda_{rl}(t)$. The average intensity of the arrival process between time t_i and t_j is given by $\lambda_{rl}(t_i, t_j) = \left(\frac{1}{t_j - t_i}\right) \int_{t_i}^{t_j} \lambda_{rl}(t)dt$. The mean number of parts of type r at station l is denoted by $n_{rl}(t)$ and the total number of jobs of type r in the system is given by $n_r(t) = \sum_{l=1}^{L} n_{rl}(t)$. Let $\boldsymbol{n_l}(t)$ be the vector of mean distribution of jobs by part type at workstation l at time t, i.e. $\boldsymbol{n_l}(t) = (n_{1l}(t), \ldots, n_{Rl}(t))$. The mean throughput rate for part type r at time t is denoted by $X_r(t)$ and the mean throughput rate at workstation l for product type r is denoted by $X_{rl}(t)$. Let $\chi_r(t)$ be cumulative production of part type r up to time t. The expected visit counts of a part of type r to workstation l from dispatch until completion are denoted by ν_{rl}. The mean internal arrivals of part type r at workstation l at time t are given by $\gamma_{rl}(t)$. Let $\psi_{rl}(t)$ be the mean number of jobs of part type r leaving workstation l at time t.

The following assumptions are made:

1. The process plan for each part type is known along with operation setup and variable processing time.
2. Service discipline at each workstation is First Come First Serve (FCFS) for jobs of the same part type and is dictated by part priorities for jobs of different part types. For this chapter, all priorities are assumed to be equal.
3. The capacity of the staging queues are infinite (to be subsequently relaxed).
4. Process routes are known and jobs relate to part types.
5. Workstations are assumed to be reliable. Unreliable workstations can be modeled by appropriately modifying the exponential service rate assuming no jobs are lost.

2.2 Review of Dynamic Queueing Models

The performance evaluation of dynamic manufacturing systems is viewed as the analysis of a multi-class nonstationary queueing network with non-homogeneous Poisson arrivals and exponential service. The first two subsections present a review of existing work on the analysis of nonstationary queues and queueing networks. The later part of this section reviews work on lead time forecasting in manufacturing systems.

2.2.1 Nonstationary Queues

The differential-difference equations describing the dynamics of a nonstationary Markovian queue $M_t/M_t/c_t$ do not have a closed form solution, thus, resulting in a number of approximations being proposed. The majority of the approximations proposed for the analysis of nonstationary queues can broadly be classified into three categories, namely, systems approximations, numerical approximations and process approximations (Schwarz *et al.* (2016), Gross (2008)). Systems approximations usually employ a period-by-period analysis approach where the nonstationary queue of interest is approximated by an equivalent stationary queue for a given period. Some notable systems approximations are the pointwise stationary (PSA) approximation (Green and Kolesar (1991)), effective arrival rate (EAR) approximation (Thompson (1993)), lagged Stationary independent period-by-period (SIPP) approximation (Green *et al.* (2001), Green *et al.* (2003)) and the stationary backlog carryover (SBC) approximation (Stolletz (2008)). Whitt (1991) proved that the PSA is asymptotically correct for $M_t/M_t/s$ queues as the service rates and arrival rates increase such that instantaneous traffic intensity remains constant. Numerical approximations usually make some simplifying assumptions to make the differential-difference equations tractable and employ numerical procedures to approximate system performance. Some noteworthy numerical approximations are the closure approximation (Rothkopf and Oren (1979), Clark (1981), Taaffe and Ong (1987)) and the randomization method (Grassmann (1977)). Process approximations use limiting arguments to substitute the stochastic process of interest with a continuous fluid or diffusion process under heavy traffic assumptions (see Di Crescenzo and Nobile (1995), Mandelbaum and Massey (1995), Massey and Whitt (1998), Massey (2002), Liu and Whitt (2011), Whitt (2014), Pender (2014)). Wang *et al.* (1996) proposed the pointwise stationary fluid flow approximation (PSFFA) which combines the PSA with a simple fluid flow model to approximate performance of nonstationary single server queues.

2.2.2 Nonstationary Queueing Networks

Only limited research has appeared on nonstationary networks and most of that has had limited computational justification. Duda (1986) proposed a diffusion approximation for the transient analysis of $GI/GI/1$ queue as the basis for the analysis of an open, general nonstationary single server queueing network. Tipper and Sundareshan (1990) suggested a numerical approximation and a nonlinear state model. Massey and Whitt (1993) study networks of infinite-server queues. Malone (1995) proposed a decomposition approximation for open nonstationary networks of single-server queues. Mandelbaum and Massey (1995) investigated fluid and diffusion limits for large scale Markovian service networks. Whitt (1999) presented a decomposition approximation framework for time-dependent Markovian queueing networks based on a generalization of a Jackson queueing network. Nelson and Taaffe (2004) developed a numerically exact method for the analysis of a multi-class network of $PH_t/PH_t/\infty$ queues. Alnowibet and Perros (2009) developed an iterative scheme for estimating mean number in system and blocking probability for the nonstationary $M_t/M/s/s$ loss queue and extended it to the analysis of a network of nonstationary loss queues. Loss queues and loss networks are not directly applicable to the problem being addressed in this chapter since all arrivals to the system are served through their entire process plan before leaving the system. The primary interest is in determining system throughput times and queue lengths for time-varying arrival processes and operating plans, potentially with limited space or WIP goals. Izady and Worthington (2011) presented approximations for the analysis of nonstationary loss queues and networks of nonstationary loss queues. Izady and Worthington (2012) proposed a heuristic staffing algorithm for time dependent queueing networks motivated by staffing problems in Accident and

Emergency (A&E) departments. Networks of time-varying many-server fluid queues were studied in Liu and Whitt (2013). Pender (2016a) analyzed a two stage tandem queue with coupled processors through a closure approximation for the functional Kolmogrov forward equations. Pender (2016b) presented a sampling approach for approximating expectations employed in closure approximations for the analysis of nonstationary queueing networks. More recently, Askin and Jampani Hanumantha (2017) and Askin and Hanumantha (2018) discuss stochastic systems approximations for the analysis of nonstationary queueing networks.

2.2.3 Lead Time Estimation in Dynamic Manufacturing Systems

Estimating the throughput time from the release of a job for manufacture until its availability for use is a key requirement for ensuring smooth and efficient, lean manufacturing. Planned lead times as derived from throughput time estimates are key drivers of MRP systems and form the basis for setting shop priorities and quoting delivery dates to customers (Vig and Dooley (1993), Askin and Goldberg (2002)). Accordingly, lead or throughput time estimation has been investigated by many researchers. Most of this work has dealt with the impact of scheduling rules (see for instance Baker (1984), Lee (2009), Cheng and Gupta (1989)). Duenyas (1995) studied the problem of due-date setting and sequencing of orders in a production system with multiple customer classes with different lead time preferences through a Semi-Markov Decision Process model of a single stage queue. Duenyas and Hopp (1995) modeled the lead time quotation problem under three different assumptions - namely infinite plant capacity, finite plant capacity and a combination of finite plant capacity and scheduling constraints. Weng (1996) presented optimal policies for order-acceptance and lead times for a make-to-order system with two different customer classes - lead time sensitive customers and lead time insensitive customers. Spearman and Zhang (1999) investigated optimal lead time policies in a single class manufacturing system with infinite buffers and general service. They compared the effects of constraining the fraction of tardy jobs against constraining the average tardiness of jobs. Weng (1999) prescribed optimal lead time policies for a make-to-order production system by modeling flow times as a generalized phase type distribution dependent of idle capacity. Webster (2002) analyzed optimal capacity and pricing policies for a single stage production system with lead time and price dependent demand. Öztürk *et al.* (2006) developed simulation metamodels based on regression trees to estimate lead times in make-to-order manufacturing systems. A deterministic flow graph method for determining throughput times for arriving jobs based on current shop status was proposed in Ioannou and Dimitriou (2012). Queueing analysis and experience indicate a nonlinear relationship between workload and lead time. A few researchers have addressed broader stochastic models. Stochastic approximation was used to determine optimal release times for a fixed set of jobs in Hasan and Spearman (1999). A static open queueing network model was used to predict lead time in Vandaele *et al.* (2002).

2.2.4 Motivation for Research

There has been quite a bit of work done on the performance analysis of single stage, single class, nonstationary queues motivated by applications in manufacturing, telecommunications, healthcare, call–center staffing and air traffic management. However, research is limited on analysis of nonstationary, multi-class queueing networks. For manufacturing systems incorporating the network structure of the system is key to understanding how congestion evolves over time at different points of the system. This understanding is important to estimate throughput times and for determining appropriate release and control strategies. Networks also provide a basis for modeling the impact of class priorities. The majority of work on the performance analysis of nonstationary systems focuses on single class loss networks where customer abandoments and rejections occur. There has been very little research on the analysis of nonstationary manufacturing systems with multiple products. The rest of this chapter is organized as follows. Section 2.3 presents incremental approximation which exploit equivalences between open and closed queueing networks. Section 2.4 presents a parametric decomposition based incremental approximation for the analysis of nonstationary manufacturing systems with multiple products. Section 2.5 builds on the throughput rate and WIP level approximations developed in Sections 2.3 and 2.4 to obtain estimates of lead time by product type. The results of the approximations applied to realistic flowshops and jobshops is presented and discussed in Section 2.6.

2.3 Closed Network Based Performance Approximations

In this section an approximation is developed for performance analysis based on a closed queueing network paradigm. A closed queueing network is a queueing network where the number of customers in the system is kept constant, with new customers being introduced into the system as customers depart the system. Specifically, incremental approximations are presented where the current state of the system is mapped to an equivalent closed queueing network. In practice, closed networks are known to be more efficient than open networks from a production efficiency perspective. The computational advantage is that performance measures for closed networks can be computed offline and stored. An aggregated product level approximation is presented initially, which, is subsequently extended to obtain estimates of WIP and throughput rates by product type and workstations.

The following additional notation is introduced in this section to represent the equivalent closed queueing network(s) employed for approximation purposes. Let the closed product form queueing network with state vector $\boldsymbol{N}$ be $C(\boldsymbol{N})$, where $\boldsymbol{N} = (n_1^c, \ldots, n_R^c)$. Let $n_{rl}^c(\boldsymbol{N})$ be the mean total number of jobs on routing part type r at workstation l for $C(\boldsymbol{N})$ $\left(n_r^c(\boldsymbol{N}) = \sum_{l=1}^{L} n_{rl}^c(\boldsymbol{N})\right)$. Let $\boldsymbol{n_l^c}(\boldsymbol{N})$ be the vector of distribution of jobs at workstation l by routing part type r for $C(\boldsymbol{N})$. Let $X_r^c(\boldsymbol{N})$ be the mean throughput rate for part type r of $C(\boldsymbol{N})$. Efficient algorithms exist for computing $C(\boldsymbol{N})$ (see, for instance, Reiser and Lavenberg (1980)).

2.3.1 Closed Network Based Throughput Model (CNBTM)

The performance of a dynamic manufacturing system at any point in time is approximated by comparing workstation state against a population of pre-computed closed queueing networks varying in constant work-in-process levels. The model operates at the total system level to match progression of each job class. The starting number of jobs in system is set to match the throughput of an equivalent open network. The number of jobs of each part type in system are updated through flow balance equations. The throughput rate for each part type is then obtained by a weighted sum of throughput rates for the closest static closed network using the floor and ceiling

WIP levels. Cumulative production is accumulated for each period. The pseudocode for the approximation is detailed below.

```
1  **Algorithm:** Closed Network Based Throughput Model (CNBTM)
2  Use Mean Value Analysis (MVA) (Reiser and Lavenberg (1980)) to solve the
     closed queueing network C($\mathbf{N^*}$) ($\mathbf{N^*}$ sufficiently large).
3  Initialize $t = 0$. Set $\mathbf{N_0}$ = $(n_1(0), ..., n_R(0))$ to match an equivalent open
     queueing network.
4  Set $X_r(0) = X_r^c(\mathbf{N_0})$ for each $r = 1, \ldots, R$.
5  while $t \leq T$ do
6      $t \leftarrow t + t_s$.
7      for $r \in \{1, \ldots, R\}$ do
8          $n_r(t) \leftarrow max(n_r(t - t_s) + t_s \sum_{j=1}^{L} \lambda_{rj}(t - t_s, t) - t_s X_r(t - t_s), 0)$
9          $X_r(t) \leftarrow (1 - \alpha_r) X_r^c(\boldsymbol{N_l}) + \alpha_r X_r^c(\boldsymbol{N_u})$, where
10         $\boldsymbol{N_l} = (\lfloor n_1(t) \rfloor, ..., \lfloor n_R(t) \rfloor)$
11         $\boldsymbol{N_u} = (\lceil n_1(t) \rceil, ..., \lceil n_R(t) \rceil)$
12         $\alpha_r = n_r(t) - \lfloor n_r(t) \rfloor$
13         $\chi_r(t) \leftarrow \chi_r(t - t_s) + min(t_s X_r(t - t_s), n_r(t - t_s) + t_s \sum_{j=1}^{L} \lambda(t - t_s, t))$
14     end
15 end
```

Mean Value Analysis is used to pre-compute performance estimates for a closed queueing network with sufficiently large constant work-in-process levels. Time is initialized to zero and the starting CONWIP level is set to match an equivalent open queueing network with initial arrival rate conditions. The throughput rate at time zero is initialized in Step 3. Steps 6 through 13 are repeated in increments of t_s until the time horizon of interest is covered completely. Step 8 updates the number in system of part type r by adding the expected number of arrivals of part type r to the system and deducting the expected number of departures from the system in the time interval $(t - t_s, t]$. Step 9 updates the throughput rate for part type r as a weighted combination of throughput rates of static closed queueing networks with the floor and ceiling WIP levels. Step 13 updates cumulative production by part type.

2.3.1.1 Proposition 1

The CNBTM approximation is exact for any stationary closed queueing network for which MVA is exact.

Proof Consider a multi-class closed queueing network with R part types and L workstations. Let the constant number of jobs in system be denoted by $N_r, r \in \{1, \ldots, R\}$.

The starting condition of the network is given by $n_r(0) = N_r,\ r \in \{1, \ldots, R\}$. Note that for a closed queueing network the following equation holds as there are a constant number of jobs in the system.

$$\sum_{j=1}^{L} \lambda_{rj}(0, t_s) = X_r(0),\ r \in \{1, \ldots, R\}$$

Then Step 8 of CNBTM at time $t = t_s$ becomes,

$$\begin{aligned} n_r(t_s) &= max(n_r(0), 0),\ r \in \{1, \ldots, R\} \\ &= N_r,\ r \in \{1, \ldots, R\} \end{aligned}$$

The throughput rate at time $t = t_s$ can be obtained from Step 13 of CNBTM as

$$X_r(t_s) = (1 - 0)X_r^c(N_r) + 0X_r^c(N_r + 1),\ r \in \{1, \ldots, R\}$$

$$= X_r^c(N_r),\ r \in \{1, \ldots, R\}$$

The number of jobs in system and throughput rate remain constant over time and exactly match standard queueing results. Thus, the CNBTM is exact for any closed queueing network for which MVA is exact.

2.3.2 Route-Workstation Based Throughput Model (RWBTM)

A linked period-by-period systems approximation approach is proposed for performance analysis of manufacturing systems with multiple part types under dynamic arrival conditions. At each period, the current state of a workstation represented by a vector of number of jobs by class is compared against a set of static closed queueing networks varying in constant work-in-process (CONWIP) levels. The static closed queueing network solutions are obtained through the Mean Value Analysis (MVA) algorithm (see Reiser and Lavenberg (1980)) for a sufficiently large CONWIP level. The major advantage of this approach is that the MVA algorithm gives estimates of WIP by part type and workstation for all intermediate CONWIP levels and can be pre-computed offline. The closest matching closed queueing network configuration is then used to update throughput rate by workstation for a given period. The state of the system for the following period is subsequently updated through flow balance equations. The pseudo-code for the algorithm is detailed below.

```
1  Algorithm: Route-Workstation Based Throughput Model (RWBTM)
2  Use MVA to solve the closed queueing network C($\mathbf{N^*}$) ($\mathbf{N^*}$ sufficiently large).
3  Initialize $t$ = 0. Set $N_0 = (n_1(0), ..., n_R(0))$ to match an equivalent open
      queueing network.
4  Set $\mathrm{X_r}(0) = \mathrm{X_r}^{\mathrm{c}}(\mathbf{N_0})$ for each $r = 1, \ldots, R$.
5  Set $n_{rl}(0) = n^c_{rl}(\boldsymbol{N_0}),\ \gamma_r l(0) = 0,\ \psi_{rl}(0) = 0,\ X_{rl}(0) = \nu_{rl} X^c_r(\boldsymbol{N_0})$ for each
      $r = 1, \ldots, R$ and each $l \in S(r)$.
6  while $t \leq T$ do
7      $t \leftarrow t + t_s$
8      for $r \in \{1, \ldots, R\}, l \in S(r)$ do
9          $\psi_{rl}(t) \leftarrow min(n_{rl}(t - t_s) + t_s\lambda_{rl}(t - t_s, t), t_s X_{rl}(t - t_s))$
10         $\gamma_{rl}(t) \leftarrow \sum_{j=1}^{L} \psi_{rj}(t) p^r_{jl}$
11         $\chi_r(t) \leftarrow \chi_r(t - t_s) + \sum_{l=1}^{L} \left[\left(1 - \sum_{j=1}^{L} p^r_{lj}\right) \psi_{rl}(t)\right]$
12         $n_{rl}(t) \leftarrow n_{rl}(t - t_s) + t_s\lambda_{rl}(t - t_s, t) + \gamma_{rl}(t) - \psi_{rl}(t)$
13         $\boldsymbol{N_l^*}(t) = \{\boldsymbol{N} :\ \boldsymbol{n_l^*(t)} = min_{\boldsymbol{N} \in \boldsymbol{N^*}} \|\boldsymbol{n}_l(t) - \boldsymbol{n}^c_l(\boldsymbol{N})\|_2\}$
14         $X_{rl}(t) \leftarrow \nu_{rl} X^c_r(\boldsymbol{N_l^*}(t))$
15     end
16 end
```

The MVA estimates are obtained in Step 1 similar to CNBTM with the only difference being that WIP level estimates by product type and wokrstation are stored as well. The throughput rate and other auxillary variables are initialized in Steps 2 through 4. Steps 5 through 14 are repeated in time increments of t_s until the entire time horizon is covered. Step 9 tracks the number of departures by part type and workstation for the time interval $(t - t_s, t]$. Step 10 tracks the internal arrivals by part type at a workstation. Step 11 updates number of jobs produced by part type.

Step 12 updates WIP levels by part type for all workstations. Step 13 compares the current WIP distribution at a workstation against the population of pre-computed closed queueing solutions to find the closest matching solution. Throughput rate is updated by part type and workstation in step 14.

The RWBTM approximation has a few limitations which restrict its application to a wide range of performance evaluation scenarios. The RWBTM approximation does not perform very well for imbalanced networks (non-identical service requirements at each workstation for a given product type). This can be explained by the fact that increasing the CONWIP of a single product results in an increase in the WIP level at just the bottleneck workstation in the closed queueing network. This behavior makes it difficult to capture a wide range of WIP level distributions across workstations for each product type. The other limitation with the RWBTM is that the time and space requirements become prohibitively large for larger queueing networks. Given these limitations of RWBTM an open network based parametric decomposition approximation is proposed in the following section.

2.4 Open Network Based Approximation

In this section the open network representation of the dynamic manufacturing system is approximated through snapshots the system at discrete points in time. The approximation combines an incremental time step by time step approach with parametric decomposition of the network to obtain estimates of throughput rate and WIP as a function of time.

2.4.1 Open Network Based Throughput Model (ONBTM)

A decomposition based approach is employed to approximate the performance of the system in an open network format. For purposes of computational efficiency the approach implicitly assumes stochastic independence between workstations. The model is described for a network of single server queues.

Here $\lambda^r_{ij}(t),\ r \in \{0,\dots,R\},\ i \in \{0,\dots,L\},\ j \in \{1,\dots,L\}$ is used to denote rate of class r arrivals at workstation j from workstation i with workstation 0 representing external arrivals. The pseudo-code for single server workstations is detailed below.

```
1 **Algorithm:** Open Network Based Throughput Model (ONBTM)
2 Initialize $n_{rl}$.
3 **while** $t \leq T$ **do**
4     $X_{rl}(t) \leftarrow min\left[\left(\frac{n_{rl}(t)}{\sum_{p\in R(l)} n_{pl}(t)+1}\right)\mu_l,\ \left(\frac{n_{rl}(t)+t_s\lambda^r_{0l}(t)}{t_s}\right)\right]$
5     $\lambda^r_{kl}(t) = p^r_{kl}X_{rk}(t),\ k \in S(r) \setminus \{0\}$
6     $n_{rl}(t+t_s) \leftarrow max\left[n_{rl}(t)+t_s\left(\sum_{j=0}^{L}\lambda^r_{jl}(t)-\sum_{j=0}^{L}\lambda^r_{lj}(t)\right),0\right]$
7     $t \leftarrow t+t_s$
8 **end**
```

The WIP levels are initialized in Step 1. Steps 2 through 7 are repeated in increments of t_s until the time horizon of interest is covered. At each iteration throughput rate by workstation and product type are updated. This is followed by estimating internal arrival rates between workstations. Finally WIP levels are updated through flow balance and time is incremented. Step 4 adjusts effective service rate for the proportional number of jobs of each part type and overall throughput to the WIP relationship for an $M/M/1$ queue. Specifically, for an $M/M/1$ queue, the number of jobs N in the system is related to the utilization ρ by $N = \frac{\rho}{(1-\rho)}$. Inverting this relationship provides $\rho = \frac{1}{(N+1)}$. Multi-server systems could be adjusted accordingly. Production is also limited by the available jobs.

2.4.1.1 Proposition 2

The ONBTM approximation is exact for a single part type stationary open queueing network with exponential service as $t_s \rightarrow 0$.

Proof Consider a single class stationary open queueing network with L single server workstations having exponential service with rate μ_l. Jobs arrive at the network according to a stationary Poisson process with rate λ. The subscript r is dropped for the rest of this proof as the network has a single job class.

Consider an arbitrary workstation, l in this network. The starting number of jobs at workstation l can be computed by using the steady state $M/M/1$ result. Thus,

$$n_l(0) = \frac{\lambda}{\mu_l - \lambda}$$

Given the starting condition, $n_l(0)$ of workstation l use Step 4 of ONBTM to obtain the below relation.

$$\lim_{t_s \to 0} X_l(0) = min\left[\left(\frac{n_l(0)}{n_l(0)+1}\right)\mu_l, \left(\frac{n_l(0)}{t_s} + \lambda\delta(l)\right)\right],$$

$$where\ \delta(l) = \begin{cases} 1, & if\ external\ arrivals\ at\ station\ l \\ 0, & otherwise \end{cases}$$

$$= min(\lambda, \frac{n_l(0)}{t_s} + \lambda\delta(l))$$

$$= \lambda$$

Advancing to the next time step and applying equation Step 6 of ONBTM, the number of jobs at workstation l at time t_s is given by

$$n_l(t_s) = max\left[n_l(0) + t_s\left(\sum_{j=0}^{L}\lambda_{jl}(0) - \sum_{j=0}^{L}\lambda_{lj}(0)\right), 0\right]$$
$$= n_l(0)\ since \sum_{j=0}^{L}\lambda_{jl}(0) = \sum_{j=0}^{L}\lambda_{lj}(0)\ under\ stationary\ conditions$$

The above relations establish that the estimates of mean throughput rate and mean number of jobs at workstation l from ONBTM remain constant over time and exactly match standard queueing results. Since the choice of workstation l is arbitrary, the ONBTM approximation is exact for a single class stationary open queueing network.

2.5 Lead Time Forecast under Dynamic Conditions

Estimates of WIP ($n_r(t)$ or $n_{rl}(t)$) and throughput rates ($X_l(t)$ or $X_{rl}(t)$) are obtained from the CNBTM, RWBTM and ONBTM. These estimates are meaningful for operational control in managing resources. In addition, it may be of interest to estimate flow times for lead time quotation, material planning or other purposes. Four separate lead time forecasts are derived from each of the above estimates as detailed below.

Consider a job on routing part type r entering the system at time t_{curr}. Let v_r be a one-to-one mapping from $\{1, .., |S(r)|\}$ to $S(r)$ such that $v_r(i) = l$ implies the i^{th} workstation visited by a job of product type r is workstation l.

2.5.1 Little's Law Based Lead Time Forecast (LTF1)

The expected completion time for a job on part type r at workstation l, t_{rl}^{comp} can be calculated using Little's Law for the current state of the system. The lead time forecast is then obtained by a weighted average of waiting times (by Little's Law) for all periods covered by the time interval between time of start and expected completion time.

```
Algorithm: Little's Law Based Lead Time Forecast (LTF1)
Compute $X_{rl}(t), n_{rl}(t) \; \forall \; r \in \{1, \dots, R\}, \; l \in S(r), \; t \in \{0, t_s, 2t_s, \dots, \lceil \frac{T}{t_s} \rceil\}$
for $r \in \{1, \dots, R\}$ do
    Initialize $t = t_{curr}$
    for $i \in \{1, \dots, |S(r)|\}$ do
        Compute expected time of completion at workstation $v_r(i)$ as
        $t_{rv_r(i)}^{comp} \leftarrow \left( \frac{n_{rv_r(i)}(t)}{X_{rv_r(i)}(t)} \right)$
        Compute lead time forecast for workstation $v_r(i)$ as
        $w_{rv_r(i)}(t) \leftarrow \left[ \left\lceil \frac{t_{rv_r(i)}^{comp}}{t_s} \right\rceil + 1 \right]^{-1} \sum_{k=0}^{\left\lceil \frac{t_{rv_r(i)}^{comp}}{t_s} \right\rceil} \frac{n_{rv_r(i)}(t+kt_s)}{X_{rv_r(i)}(t+kt_s)}$
        $t \leftarrow t + w_{rv_r(i)}(t)$
    end
    $W_r(t_{curr}) \leftarrow t - t_{curr}$
end
```

The first lead time approximation, denoted $LTF1$, computes the expected lead time for each product type assuming a job of that product enters the system at time t_{curr}. The approximation iterates over the sequence of workstations a job of a certain product type visits and computes the expected completion time at each workstation. The expected completion times are propagated from one workstation in the jobs processing sequence to the next until all workstations in the sequence are exhausted. The estimates of throughput rates and WIP levels are obtained from one of the approximations presented in previous sections in Step 2. Step 3 initializes the time of entry of a job of product type r to t_{curr}. Steps 3 through 10 are repeated until all stations in a jobs processing sequence are covered. Step 6 computes the Little's law estimate for time in system type given current workstation conditions. Step 7 takes an average of Little's Law estimate for every time step of length t_s in the interval $[t, t + t^{comp}_{rv_r(i)}]$ to give a lead time forecast for product type r at workstation $v_r(i)$. The workstation level lead time forecasts are accumulated over steps 3 through 10. The lead time forecast $W_(t_curr)$ is obtained by deducting t_{curr} from the accumulated lead time in Step 10.

2.5.2 Average Work Completion Based Lead Time Forecast (LTF2)

Consider a job on routing part type r entering the system at time t_{curr}. The amount of work completed on this job in subsequent time windows can be estimated and combined to estimate the time at which this job leaves the system. Using time step t_s the model accumulates effective processing time for the job. The model can be viewed as either a processor sharing analogy with average WIP or a clearing function with FCFS processing.

```
1  Algorithm: Average Work Completion based Lead Time Forecast (LTF2)
2  Compute $X_{rl}(t), n_{rl}(t) \ \forall \ r \in \{1, \dots, R\},\ l \in S(r),\ t \in \{0, t_s, 2t_s, \dots, \lceil \frac{T}{t_s} \rceil\}$
3  for $r \in \{1, \dots, R\}$ do
4      Initialize $t = t_{curr}$
5      for $i \in \{1, \dots, |S(r)|\}$ do
6          Compute lead time forecast for workstation $v_r(i)$ as
           $j^* = min\left\{ j \in \mathbb{Z}_+ : \sum_{k=0}^{j} \frac{t_s X_{rv_r(i)}(t+kt_s)}{n_{rv_r(i)}(t+kt_s)} \geq 1 \right\}$
7          $w_{rv_r(i)}(t) = j^* t_s$
8          $t \leftarrow t + w_{rv_r(i)}(t)$
9      end
10     $W_r(t_{curr}) \leftarrow t - t_{curr}$
11 end
```

Step 6 uses a processor sharing analogy to determine the index of the earliest time step, j, at which the proportion of work completed on a given job exceeds 1. Step 7 computes time spent in system by part type and workstation.

2.5.3 Average Time Remaining Lead Time Forecast (LTF3)

LTF3 iterates across time steps until the first period in which Little's Law would imply time at the workstation is less than accumulated time.

```
1  Algorithm: Average Time Remaining Lead Time Forecast (LTF3)
2  Compute $X_{rl}(t), n_{rl}(t)\ \forall\ r \in \{1, \dots, R\},\ l \in S(r),\ t \in \{0, t_s, 2t_s, \dots, \lceil \frac{T}{t_s} \rceil\}$
3  for $r \in \{1, \dots, R\}$ do
4      Initialize $t = t_{curr}$
5      for $i \in \{1, \dots, |S(r)|\}$ do
6          Compute lead time forecast for workstation $v_r(i)$ as
               $j^* = min \left\{ j \in \mathbb{Z}_+ : \left( \frac{n_{rv_r(i)}(t+jt_s)}{X_{rv_r(i)}(t+jt_s)} \right) \leq jt_s \right\}$
7          $w_{rv_r(i)}(t) = j^* t_s$
8          $t \leftarrow t + w_{rv_r(i)}(t)$
9      end
10     $W_r(t_{curr}) \leftarrow t - t_{curr}$
11 end
```

Step 6 computes the earliest time step, j, at which the actual time accumulated by moving from one period to the next exceeds the instantaneous Little's law estimate for time in system. Step 7 computes time spent in system by part type and workstation.

2.5.4 Average Time Remaining Lead Time Forecast (LTF4)

LTF4 is similar to LTF3 except it uses a weighted average of throughput time estimates as time advances.

```
1  **Algorithm:** Average Time Remaining Lead Time Forecast (LTF4)
2  Compute $X_{rl}(t), n_{rl}(t)\ \forall\ r \in \{1, \ldots, R\},\ l \in S(r),\ t \in \{0, t_s, 2t_s, \ldots, \lceil \frac{T}{t_s} \rceil\}$
3  **for** $r \in \{1, \ldots, R\}$ **do**
4      Initialize $t = t_{curr}$
5      **for** $i \in \{1, \ldots, |S(r)|\}$ **do**
6          Compute lead time forecast for workstation $v_r(i)$ as
               $j^* = min\left\{ j \in \mathbb{Z}_+ : \sum_{k=0}^{j} \left( \frac{n_{rv_r(i)}(t+kt_s)}{(j+1)(X_{rv_r(i)}(t+kt_s))} \right) \leq jt_s \right\}$
7          $w_{rv_r(i)}(t) = j^* t_s$
8          $t \leftarrow t + w_{rv_r(i)}(t)$
9      **end**
10     $W_r(t_{curr}) \leftarrow t - t_{curr}$
11 **end**
```

Step 6 computes the earliest time step, j, at which the actual time accumulated by moving from one period to the next exceeds the time averaged Little's law estimate for time in system . Step 7 computes time spent in system by part type and workstation.

Simplified versions of the above lead time forecasts are used for the CNBTM model as it only provides class level estimates of WIP and throughput rates.

2.6 Results and Analysis

The CNBTM and ONBTM approximations are tested on large flowshop and jobshop instances in this section. Approximation estimates are compared against one thousand simulation replications each of length 1000 minutes. The plots in the rest of this section present a comparison of simulation estimates (solid lines) against approximation estimates (dashed lines).

The relative errors presented in the rest of this section are calculated as shown below. Reported values are average normalized relative error equally weighted over the $\frac{1000}{t_s}$ time intervals. Normalized relative error (NRE) at any time t is given by

$$\frac{max(0, |\theta_{APPROX}(t) - \theta_{EXT}(t)| - \epsilon)}{\theta_{EXT}(t) + \epsilon} * 100 \qquad (2.1)$$

where $\theta_{EXT}(t)$ is the exact value at time t (based on the average of the simulation runs), and $\theta_{APPROX}(t)$ is the approximate estimate at time t. The user specified constant ϵ is used to guard against the possibility of very small values in the denominator from skewing the overall relative error. The value of the user specified constant used in the rest of this chapter are $\epsilon = 10^{-3}$ for WIP levels and $\epsilon = 1$ for throughput as deviations in performance estimates below these thresholds are not of practical significance.

2.6.1 Experiment 1: A Large Flowshop

A flowshop with four product types ($\{1, \ldots, 4\}$) and sixteen single server unit rate workstations ($\{1, \ldots, 16\}$) is considered. Jobs are assumed to arrive at the system according to a nonhomogeneous Poisson process with interarrival times averaging 1.25 minutes. Two nonstationary arrival patterns are considered:

1. Sinusoidal with frequency $2\pi = 100$ *mins.* and phase shift between product types.
2. Triangular over $T = 1000$ *mins.* with peaks offset by 200 *mins.*

Configuration	Workload	Bottleneck Station
1	Balanced	-
2	Unbalanced	Workstation 4
3	Unbalanced	Workstation 8
4	Unbalanced	Workstation 13
5	Unbalanced	Workstation4 and Workstation 13

Table 2.1. Flowshop Configurations

The following five workload configurations are considered for the flowshop described above as shown in Table 2.1. The service rate of the bottleneck workstations is set to 0.9 jobs/min. for the imbalanced configurations as opposed to all unit rate servers in the balanced case.

2.6.1.1 CNBTM Results

The throughput estimates from CNBTM are compared against simulation estimates for sinusoidal arrivals in Fig 2.1. The CNBTM approximation tracks throughput very closely for sinusoidal arrivals. The throughput plots are very similar for all five configurations of the flowshop.

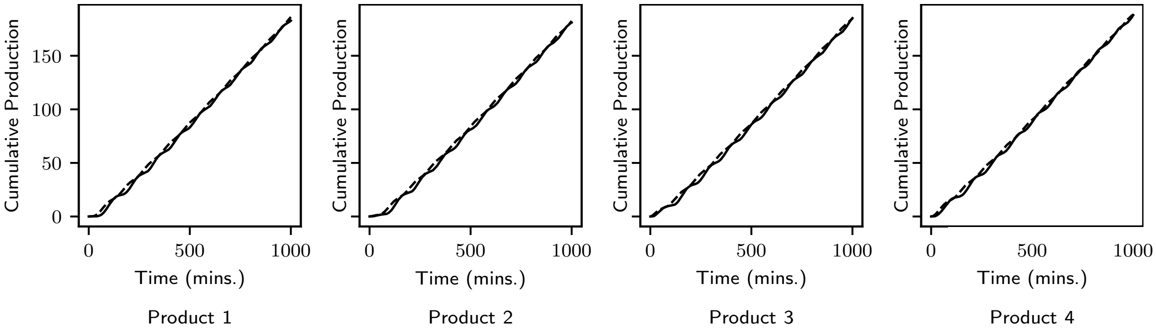


Figure 2.1. Flowshop with Sinusoidal Poisson Arrivals: Throughput

Total WIP levels over time by class for the flowshop with sinusoidal arrivals are displayed in Figure 2.2. The CNBTM only provides class level estimates of WIP without taking into account the distribution of jobs by class at each workstation. It is observed that the CNBTM does a reasonably good job capturing the trend of WIP evolution across time although it tends to underestimate the WIP levels uniformly across all classes for sinusoidal arrivals. This can be explained by the fact that the CNBTM ignores time-varying network effects as it evaluates the queueing network at the aggregated class level. The plots are similar for all five configurations of the flowshop indicating the CNBTM does not significantly distinguish between the five configurations and provides similar class level estimates as the differences in the five configurations are at the workstation level.

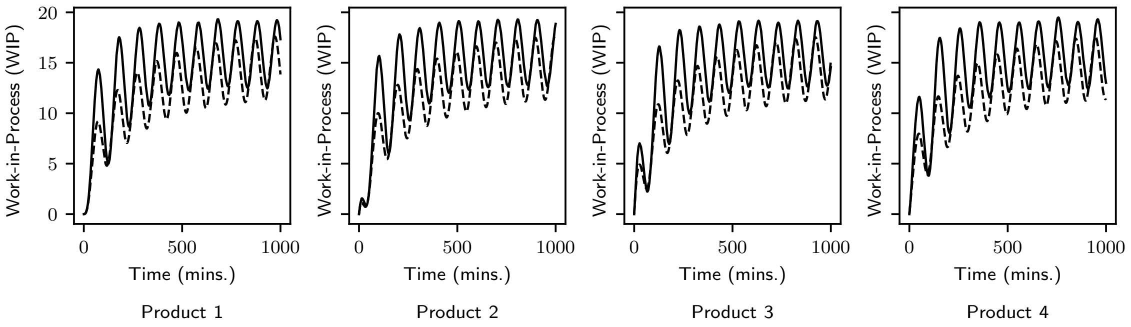


Figure 2.2. Flowshop with Sinusoidal Poisson Arrivals: WIP

Throughput estimates from CNBTM are compared against simulation estimates for triangular arrivals in Figure 2.3. The approximation does very well for class 1 and class 4 but overestimates throughput for all other classes. The throughput plots are very similar for all five configurations of the flowshop.

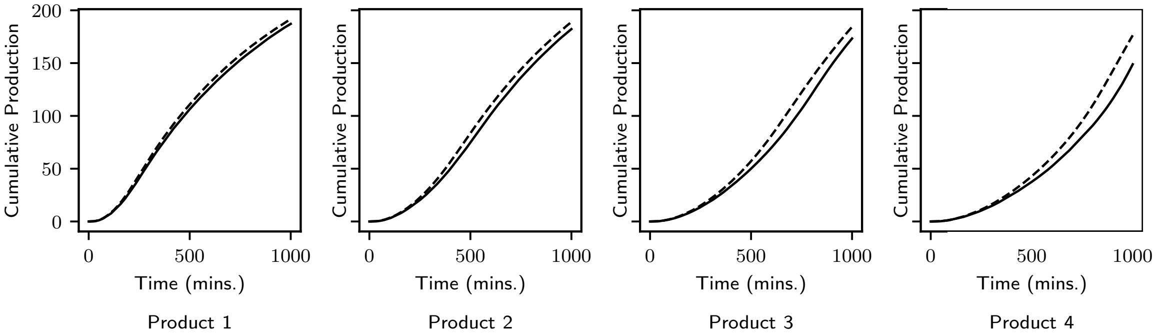


Figure 2.3. Flowshop with triangular Poisson arrivals: Throughput

Figure 2.4 plots class level WIP against time for triangular arrivals. In this case, it is seen that the CNBTM approximation performs very well in capturing the trend as well as the magnitude of WIP levels across time. However, as observed in the sinusoidal arrival case the approximation provides similar estimates of total WIP by class for all the five different configurations. Another interesting observation is that the approximation tends to lag or lead the simulation WIP estimates in the triangular case, particularly noticeable in the class 3 plot.

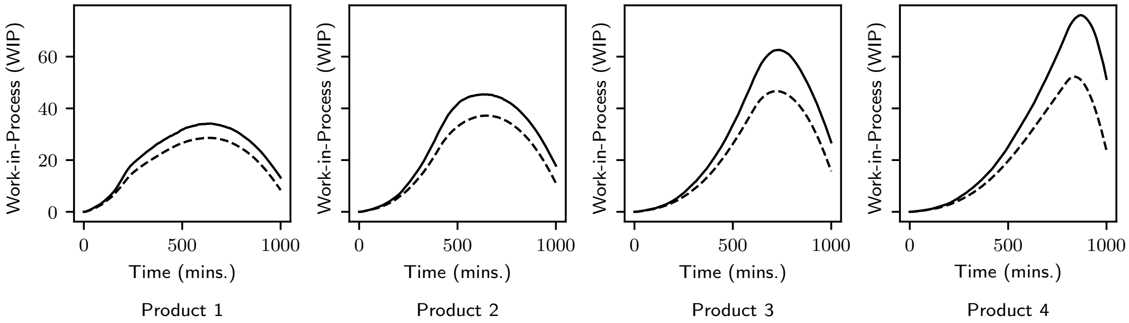
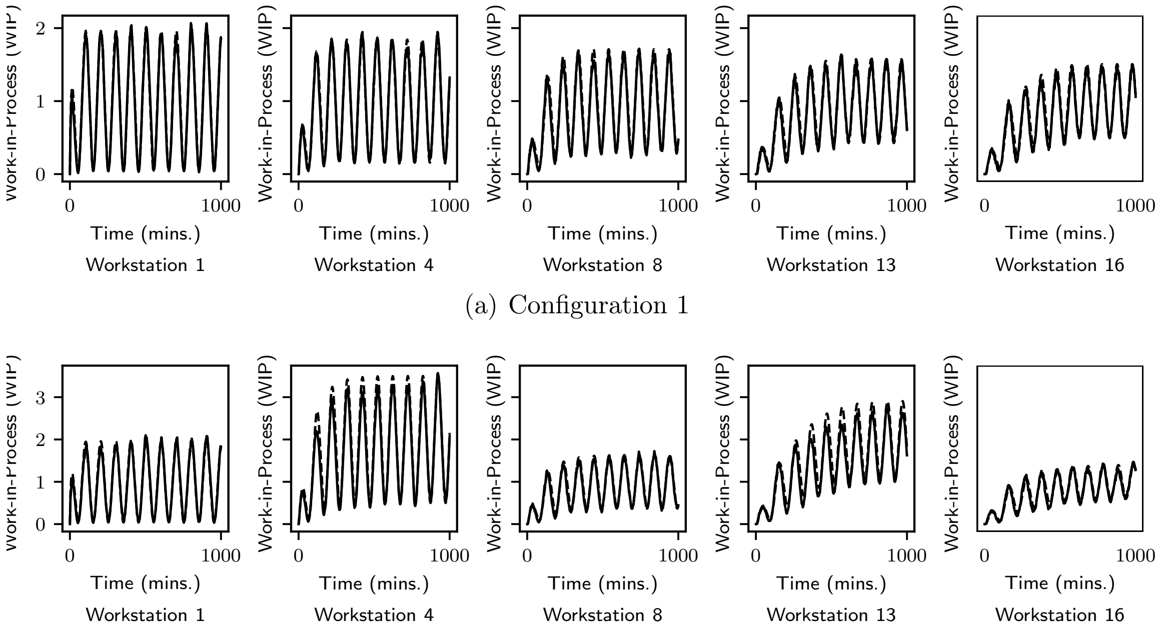


Figure 2.4. Flowshop with triangular Poisson arrivals: WIP

2.6.1.2 ONBTM Results

Unlike the CNBTM model, the open network ONBTM model tracks WIP levels by customer class at each workstation over time. In this subsection, plots are provided for WIP levels by workstation and class over time for class 3, an arbitrarily chosen class to analyze the performance of the ONBTM approximation. The plots are similar for all other three classes.

Figure 2.5 plots the WIP levels at five different points in the system for configuration 1 (balanced) and configuration 5 (imbalanced with bottlenecks at station 4 and station 13)of the flowshop with sinusoidal arrivals. It is observed that the ONBTM approximation generally tends to overestimate WIP levels for all workstations. The ONBTM approximation however, is sensitive to the differences in the five configurations and accordingly adjusts its WIP estimates at the bottleneck workstations. This is evident by the increase in WIP for station 4 and 13 in configuration 5.



(a) Configuration 1

(b) Configuration 5

Figure 2.5. Flowshop with Sinusoidal Poisson Arrivals

Table 2.2 summarizes the relative error for throughput estimates obtained from the CNBTM approximation. It is seen that the CNBTM approximation generally does well for throughput estimation under dynamic conditions and the relative errors are below 5% in most of the cases. The relative errors for the steady state estimate are significantly larger. For instance values in the first row of Table 2.2 are in the range of 5% to 8%.

Arrivals	Configuration	Product 1	Product 2	Product 3	Product 4
Sinusoidal	1	7.52	6.74	4.68	4.78
Sinusoidal	2	7.21	6.39	4.93	5.60
Sinusoidal	3	7.51	6.66	4.79	5.44
Sinusoidal	4	7.61	6.76	4.60	5.36
Sinusoidal	5	8.37	6.95	5.09	5.32
Triangular	1	2.69	5.40	8.05	10.39
Triangular	2	2.57	5.18	7.93	10.32
Triangular	3	2.97	5.18	8.46	10.47
Triangular	4	3.51	5.08	8.04	10.21
Triangular	5	3.02	5.04	8.86	10.43

Table 2.2. CNBTM Relative Error for Throughput (%)

Table 2.3 summarizes the relative error for WIP estimates obtained from the CNBTM approximation. The CNBTM does a reasonably good job of capturing the WIP trend but tends to significantly underestimate overall WIP levels for both sinusoidal and triangular arrival patterns.

Arrivals	Configuration	Product 1	Product 2	Product 3	Product 4
Sinusoidal	1	16.38	16.11	16.07	16.71
Sinusoidal	2	17.20	17.02	16.65	16.73
Sinusoidal	3	16.08	15.95	16.06	16.10
Sinusoidal	4	16.16	15.77	15.65	16.20
Sinusoidal	5	16.79	16.61	16.96	16.69
Triangular	1	18.32	19.71	22.52	23.82
Triangular	2	17.57	19.00	21.53	23.33
Triangular	3	17.26	18.69	21.48	23.20
Triangular	4	16.96	19.27	21.94	23.74
Triangular	5	16.64	18.31	20.96	23.01

Table 2.3. CNBTM Relative Error for Product Level WIP (%)

Table 2.4 summarizes the relative error for throughput estimates obtained from the ONBTM approximation. The ONBTM seems to perform pretty well for throughput estimation with most of the relative errors being below 3%.

Arrivals	Configuration	Product 1	Product 2	Product 3	Product 4
Sinusoidal	1	2.30	2.35	2.53	3.38
Sinusoidal	2	2.93	3.31	2.75	2.52
Sinusoidal	3	2.72	3.08	2.77	2.80
Sinusoidal	4	2.58	2.76	3.11	2.87
Sinusoidal	5	2.37	3.13	2.78	3.28
Triangular	1	2.42	2.00	2.13	2.36
Triangular	2	2.58	2.52	2.55	2.57
Triangular	3	1.88	2.07	2.41	2.38
Triangular	4	1.27	2.41	2.64	3.31
Triangular	5	1.76	2.53	1.94	2.89

Table 2.4. ONBTM Relative Error for Throughput (%)

Table 2.5 reports the summary statistics of normalized relative errors by product type for the ONBTM approximation sampled at a subset of five workstations $(1, 4, 8, 13, 16)$ for all configurations. The approximation does a reasonably good job of estimating WIP levels by product and workstation with all errors in the 10% to 20% range.

Arrivals		Product 1	Product 2	Product 3	Product 4
Sinusoidal	Average	9.41	10.34	10.37	9.93
	Min.	4.93	5.16	5.88	4.65
	Max.	17.71	18.89	19.19	18.13
Triangular	Average	8.77	8.99	8.8	7.77
	Min.	3.48	3.99	4.13	4.43
	Max.	22.19	22.14	20.2	17.10

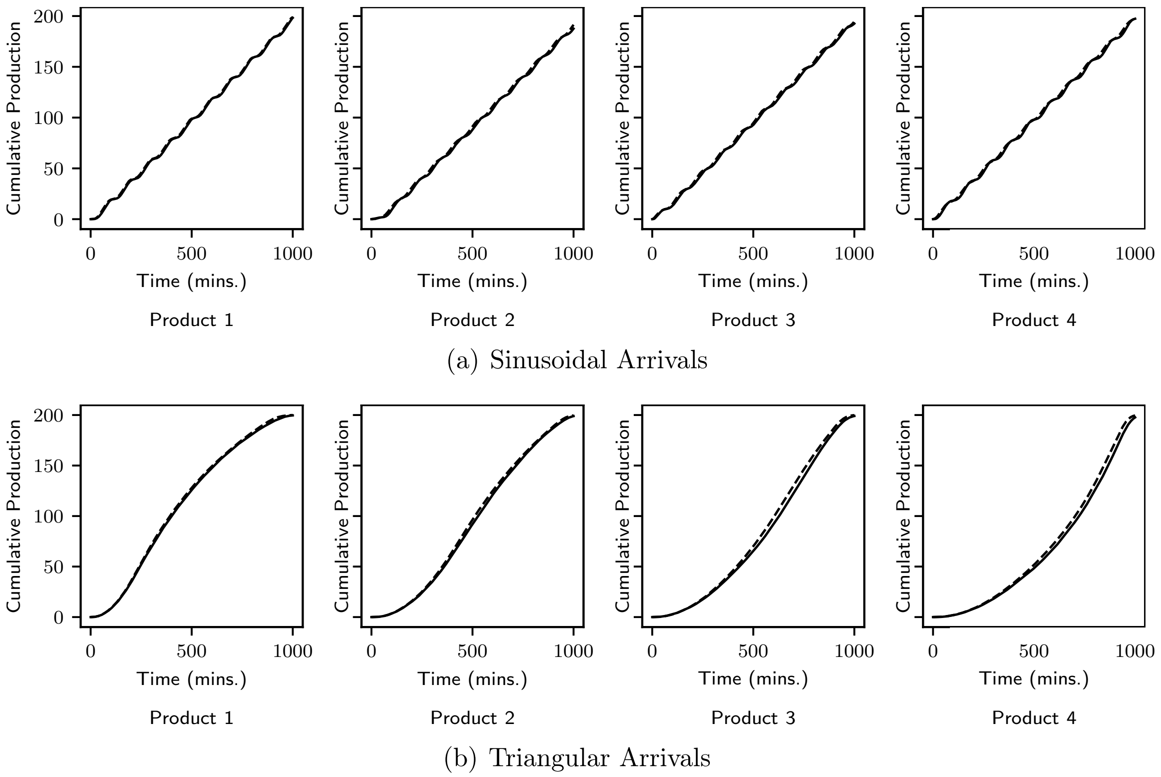
Table 2.5. ONBTM Relative Error for WIP Levels (%)

2.6.2 Experiment 2: A Large Jobshop

As a second experiment, a jobshop with four product types ($\{1, \ldots, 4\}$) and sixteen unit rate workstations ($\{1, \ldots, 16\}$) are considered. Jobs are assumed to arrive at the system according to a nonhomogeneous Poisson process with interarrival times averaging 1.25 minutes and follow a deterministic route as shown in Table 2.6.

Product	Routing
1	$4 \rightarrow 12 \rightarrow 2 \rightarrow 5 \rightarrow 13 \rightarrow 8$
2	$2 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 5 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 16$
3	$1 \rightarrow 8 \rightarrow 10 \rightarrow 9 \rightarrow 11 \rightarrow 13 \rightarrow 14 \rightarrow 4 \rightarrow 16$
4	$4 \rightarrow 5 \rightarrow 3 \rightarrow 8 \rightarrow 10 \rightarrow 6 \rightarrow 9 \rightarrow 12 \rightarrow 1 \rightarrow 15 \rightarrow 16$

Table 2.6. Jobshop Routing

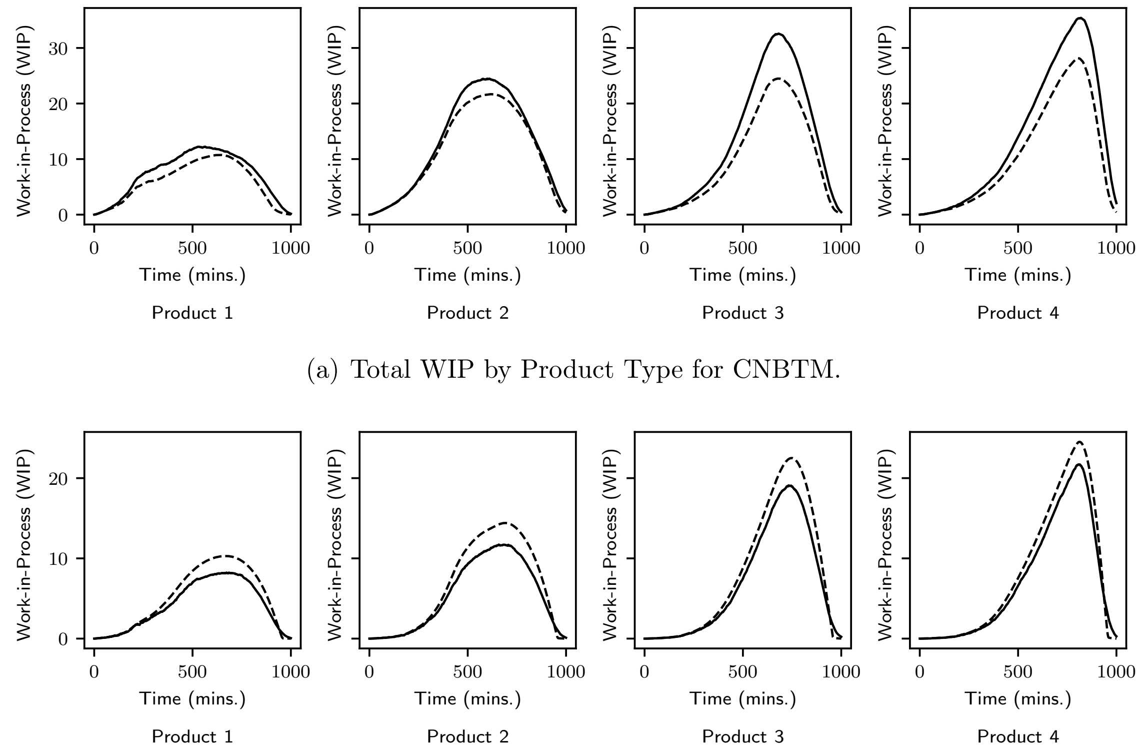


(a) Sinusoidal Arrivals

(b) Triangular Arrivals

Figure 2.6. Throughput Plots for CNBTM

Figure 2.6 shows CNBTM closely tracks throughput. The solid line indicates simulation and the dashed lines analytic model results. The corresponding plots for ONBTM track throughput pretty closely but are conservative overall. Figure 2.7(a) displays a comparison of total WIP in system by product type for CNBTM against simulation estimates. Figure 2.7(b) plots WIP estimates for ONBTM by product type at the bottleneck workstation. Dashed lines indicate approximation values and solid lines are average values across the simulation runs



(a) Total WIP by Product Type for CNBTM.

(b) WIP by Product Type at bottleneck workstation for ONBTM.

Figure 2.7. WIP for Triangular Arrivals

Figure 2.8 plots total WIP in system by product type for the CNBTM approximation with sinusoidal arrivals.

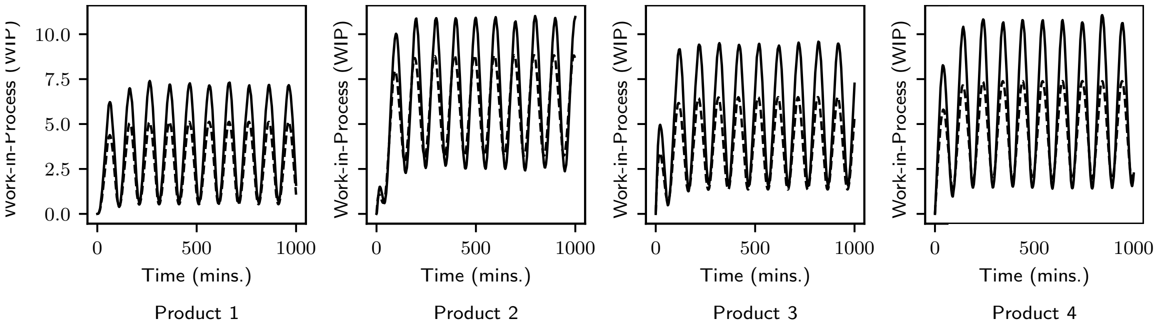
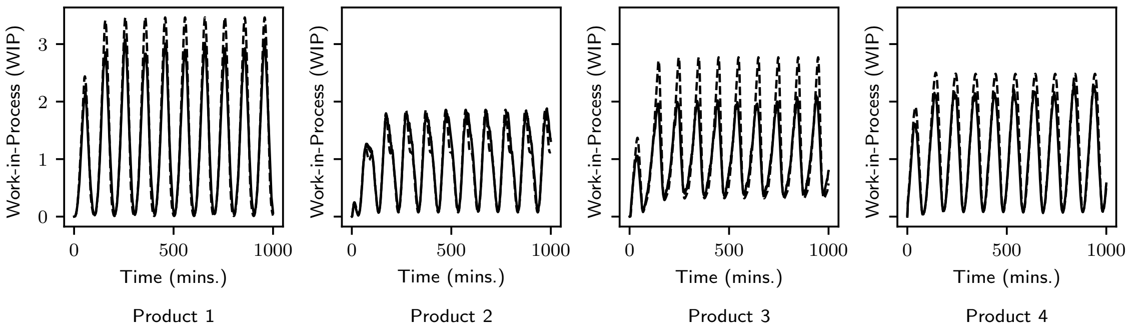


Figure 2.8. Total WIP by Product Type for CNBTM: Sinusoidal Arrivals

Figure 2.9 compares ONBTM estimates for WIP at the bottleneck workstation by product type against simulation estimates. In general, it is seen that both models capture the pattern but tend to overestimate WIP in the ONBTM model and underestimate WIP in the CNBTM.



Product 1

Product 2

Product 3

Product 4

Figure 2.9. WIP by Product Type at Bottleneck Workstation for ONBTM: Sinusoidal Arrivals

Table 2.7 reports the normalized relative errors of the ONBTM approximation for the Jobshop experiment while Table 2.8 reports the relative errors for WIP levels over time. Similar to the results found with two moment queuing approximations for stationary systems, the throughput estimate errors are typically in the range of 1% to 2%, but WIP estimates at any point in time are less accurate with relative errors occasionally exceeding 20%. Note however that the pattern clearly follows the simulation and thus estimates are much more meaningful that constant values that would be obtained by a stationary model.

Arrival Pattern	Product 1	Product 2	Product 3	Product 4
Sinusoidal Arrivals	0.96	2.03	1.73	1.72
Triangular Arrivals	1.02	1.96	1.81	1.85

Table 2.7. ONBTM Relative Errors for Jobshop Throughput (%)

Arrival Pattern	Station	Product 1	Product 2	Product 3	Product 4
Sinusoidal	1		18.17	13.38	14.96
	4	20.22	11.24	20.81	15.87
	8	17.84		16.16	13.53
	13	23.08	17.89	20.52	
	16		20.47	18.74	18.10
Triangular	1		7.74	7.77	9.99
	4	23.44	23.05	20.02	18.81
	8	9.45		6.71	10.90
	13	9.35	9.23	6.68	
	16		9.94	10.54	9.94

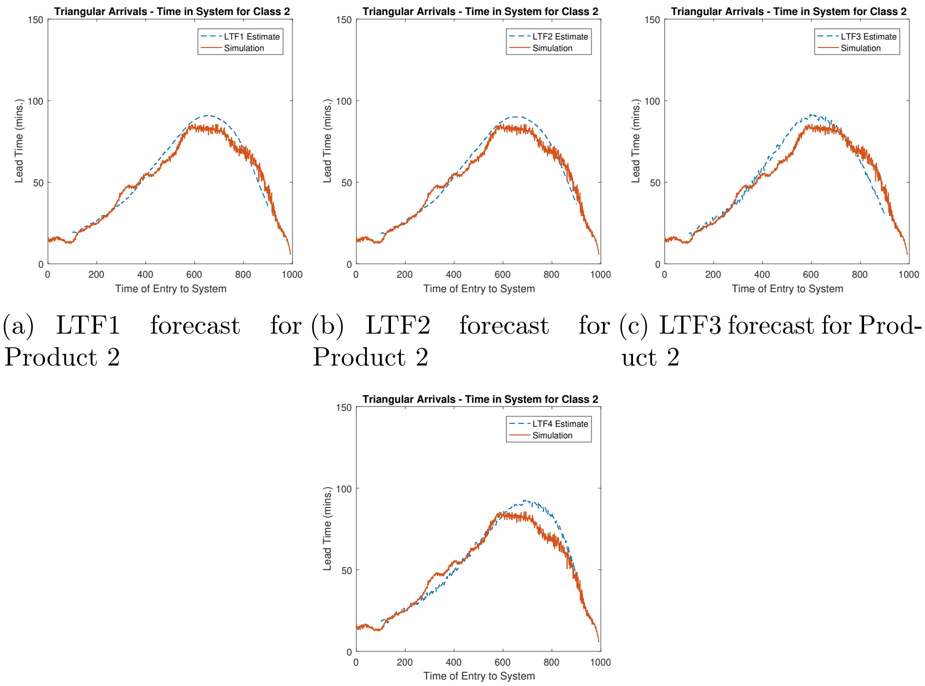
Table 2.8. ONBTM Relative Errors for Jobshop WIP Levels (%)

2.6.3 Lead Time Estimation Results

Lead time estimation is one potential use of the proposed models. The performance of the CNBTM and ONBTM models for projecting flow times is considered next.

2.6.3.1 CNBTM Results

Figs. 2.10(a) through 2.10(d) compare throughput time estimates of LTF1 through LTF4 for Product 2 jobs against simulation estimates for the CNBTM. All methods work well for triangular arrivals with LTF1 and LTF2 the best (results are similar for other Products). For the sinusoidal demand the lead time forecasts underestimated the amplitude of the seasonal patterns.



(a) LTF1 forecast for Product 2

(b) LTF2 forecast for Product 2

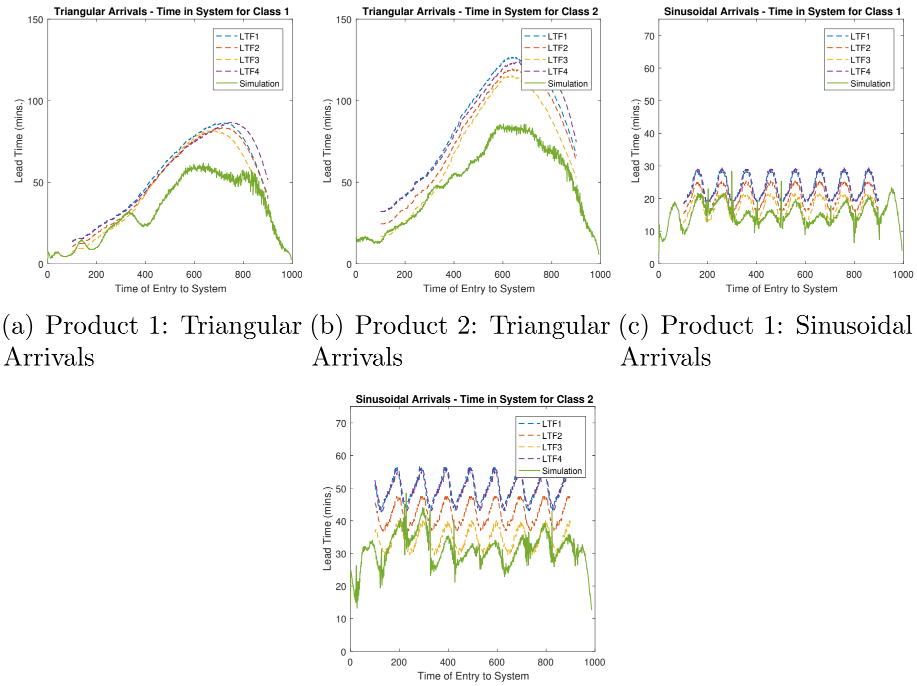
(c) LTF3 forecast for Product 2

(d) LTF4 forecast for Product 2

Figure 2.10. Lead Time Forecast for Product 2

2.6.3.2 ONBTM Results

The ONBTM model tends to overestimate WIP levels (Askin and Jampani Hanumantha (2017)) and this is reflected in the results shown in Figs. 2.11(a) through 2.11(d) for job Products 1 and 2 with both demand patterns.

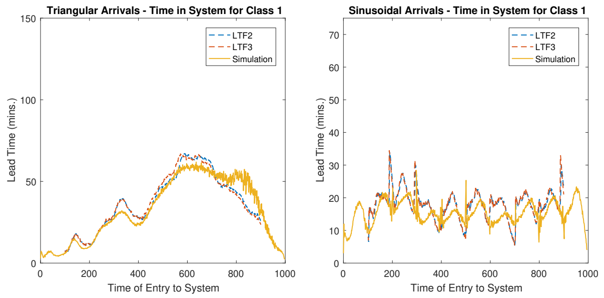


(a) Product 1: Triangular Arrivals (b) Product 2: Triangular Arrivals (c) Product 1: Sinusoidal Arrivals

(d) Product 2: Sinusoidal Arrivals

Figure 2.11. Lead Time Forecasts for Product 1 and Product 2

Results are similar in pattern for Products 3 and 4. All lead time forecast methods show similar patterns but LTF3 consistently performs best with ONBTM. To investigate whether simulation and model estimate differences are due to the format of the LTF equations or the quality of the performance estimates used, Figs. 2.12(a) and 2.12(b) use the simulation WIP estimates for Product 1 with the LTF2 and LTF3 forecasting models. Comparison to Figs. 2.11(a) and 2.11(c) indicate Step 6 of the LTF2 approximation and Step 6 of the LTF3 approximation provide reasonably accurate estimates when reliable WIP level estimates are available.



(a) Triangular Arrivals (b) Sinusoidal Arrivals

Figure 2.12. Lead Time Forecast Using Simulation WIP Estimates

Table 2.9 summarizes WIP and throughput computation times for the chosen problem instance on a desktop computer with a 3.4GHz Intel i7-6700 processor. Lead time forecast estimates require negligible time. The offline pre-computation phase of CNBTM takes a significant amount of runtime, but the online computation runtime is relatively small as compared to ONBTM computation time. However, the CNBTM does provide an advantage over ONBTM for applications where repetitive performance evaluation computations might be required. Overall, both the CNBTM and ONBTM are computationally very efficient as it only takes a couple of milliseconds to compute estimates for a large jobshop with four product types and sixteen workstation as observed in 2.9.

	CNBTM		ONBTM
	Offline Pre-computation	Online Computation	
Sinusoidal Arrivals	102.073	0.004	0.15
Triangular Arrivals	105.195	0.003	0.16

Table 2.9. Computation Time Summary (Seconds)

2.7 Computational Complexity

The computational complexity of the CNBTM and ONBTM algorithms is analyzed for a manufacturing system with L workstations or stages and R job types (products). The CNBTM model utilizes pre-computed performance estimates from the MVA algorithm. The complexity of the MVA algorithm for a closed queueing network with a fixed K jobs of each Product in the system is $O(LRK^R)$ (see Reiser and Lavenberg (1980)). The complexity of CNBTM given MVA estimates is $O(R)$ for each time step. The computational complexity of the CNBTM algorithm for the entire time horizon T under consideration is $O\Big(R\lceil \frac{T}{t_s} \rceil\Big)$. Thus, the CNBTM algorithm's complexity is linear in the number of job types and does not depend on the number of stages in the system. However, it should be noted that the complexity of the MVA algorithm grows exponentially as the number of job types are increased and linearly as the number of stages are increased.

The ONBTM algorithm involves $O(LR)$ operations for each time step as the throughput rate and WIP updates are performed by job type independently for each workstation. Thus, the complexity for the entire time horizon, T under consideration is given by $O\Big(LR\lceil \frac{T}{t_s} \rceil\Big)$. Hence, the ONBTM algorithm's complexity is linear in both the number of job types as well as the number of stages.

2.8 Effect of Approximation Step Size

The impact of the choice of time step size for the CNBTM and ONBTM approximations is studied in this section. Intuitively, one would expect the performance of the approximations to improve with diminishing step size, although, the trade-off being that this improvement would be accompanied by an increase in computation time. An experiment is performed to study the impact of step-size for the ONBTM approximation applied to a single server workstation which services two different product types. A single server workstation is chosen for the purpose of this study over a network of workstations as the ONBTM is a decomposition based approach. The reasoning behind choosing to examine a single workstation is that in a decomposition based approach the choice of step size would have a similar impact regardless of network structure and size. The accuracy of the approximation would however depend on its ability to accurately characterize the departure processes from each individual workstation. The results of the experiment for a two-product single workstation with sinusoidal arrival patterns is discussed in the rest of this section. The parameter assumptions for the experiment are detailed in Table 2.10.

Parameter	Value(s)
Service Rate, μ	1
Product 1 Arrivals, $\lambda_1(t)$	$0.275(1+\sin(\frac{2\pi t}{100}))$
Product 2 Arrivals, $\lambda_2(t)$	$0.275(1+\cos(\frac{2\pi t}{100}))$
Time step size, t_s	$\frac{\mu}{2^k}, k \in \{-2,-1,\ldots,6\}$

Table 2.10. Parameter Assumptions for Time Step Size Experiment

The experiment focuses on measuring impact of step size on WIP level estimates since WIP levels are usually harder to estimate accurately in general. The range of time step sizes considered are given by $t_s = \frac{\mu}{2^k}, k \in \{-2,-1,\ldots,6\}$.

	Normalized Relative Error	
k	Product 1	Product 2
-2	48.38	40.60
-1	27.09	20.81
0	21.95	18.62
1	19.47	17.74
2	18.27	17.38
3	17.69	17.21
4	17.41	17.13
5	17.27	17.09
6	17.20	17.07

Table 2.11. Impact of Step Size on ONBTM Accuracy

Table 2.11 reports the normalized relative errors (as defined in equation 2.1) for performance estimates of a single server workstation serving two products over the chosen range of time step sizes. The normalized relative errors are computed by comparing the approximation estimates against 5000 replications of a discrete-event simulation. The results in Table 2.11 show that although the ONBTM accuracy improves with smaller time step size a diminishing marginal improvement in accuracy is observed as the time step size shrinks. Empirically it is observed that the performance of both the CNBTM and ONBTM algorithms are relatively insensitive to relatively small time steps ($\frac{t_s}{\mu_l} \leq 0.5$).

Chapter 3

DYNAMIC MANUFACTURING SYSTEMS WITH FINITE BUFFERS AND PRODUCT PRIORITIES

In this chapter approximations are presented for dynamic multi-product manufacturing systems with finite buffers and product priorities. The study of such systems is of particular interest as these models capture the dynamics of modern manufacturing systems accurately. Specifically, multi-class finite queueing networks with non-stationary arrival processes and a priority discipline are studied for medium term capacity planning. Priority service discipline allows planners to incorporate service level differences between different product families while finite buffers model the network effects of blocking and starvation. Numerical approximations applied incrementally are explored for the performance analysis of the general class of open queueing networks described above. The objective is to develop computationally tractable semi-rapid models that can be useful primarily for performance evaluation and scenario evaluation. Other potential uses include lead time estimation/quotation, labor scheduling, equipment acquisition and maintenance planning.

3.1 Problem Definition

Consider a manufacturing facility with multiple product types (or product families), each with their own pre-defined process plans. Demands for the products are random with parameter values that vary over time. Workstations have a finite buffer capacity and process parts based on a static priority assigned to their product family. The primary performance measures of interest for this manufacturing facility are the average product level work-in-process (WIP) at each workstation and the average throughput by product as a function of time. Other aspects of interest are potential time-varying bottlenecks which might cause starvation of downstream workstations and blocking of upstream workstations. Note that the finite buffers and product specific routing permits modelling of various forms of hybrid and open networks.

3.1.1 Model Formulation

The manufacturing facility described above is modelled as a multi-class queueing network with non-homogeneous Poisson arrivals, part priorities, and finite buffers. The queueing network consists of L single-server workstations and R product types which may represent priority classes (product families). The priority classes are indexed by $r = 1, \ldots, R$ with a lower priority designation indicating higher preference. The topology of the queueing network is arbitrary and is dictated by the process plans for each priority class. The routing of jobs in the queueing network from workstation i to j is described by a routing matrix $P_r = [p_{ij}^r], i, j \in \{1, \ldots, R\}$ for product type r. The service discipline at each workstation is described by either a first-come first-serve (FCFS) discipline or a priority discipline with non-preemptive service. The blocking mechanism in the case of finite buffer capacity is assumed to be Type-I blocking (blocking-after-service). Workstations are assumed to be reliable with no breakdowns. Unreliable workstations can be incorporated by appropriately modifying the exponential service rate.

The following notation is adopted for the rest of this chapter. The set of workstations visited by product type r is described by $S(r)$. The set of product types which visit station l is denoted by $R(l)$. The time-varying Poisson arrival rate for product type r is given by $\lambda_r(t)$. The deterministic routing of parts through the network is governed by the matrix P_r. The exponential service rate of product type r at station l is denoted by μ_l. The buffer capacity at workstation l is denoted by b_l. The time-varying utilization of a station l is denoted as $\sigma_l(t)$ while product type level cumulative utilizations up to that product type are denoted by $\sigma_r(t)$. The individual product type utilizations as a function of time are denoted by $\rho_{rl}(t)$. The time-dependent throughput rates for product type r at station l are denoted by $X_{rl}(t)$. The time-varying WIP levels for product type r at station l are denoted by $n_{rl}(t)$.

3.2 Queueing Models with Priorities and Finite Buffers

In this section prior work on developing analytical approximations is reviewed. First product type priorities are considered and, then, finite buffers. The proposed approximations for a nonhomogeneous network and the performance of those approximations will be addressed in later sections.

3.2.1 Queueing Models with Priorities

Priority queueing disciplines in multi-class queueing networks provide a convenient way of modeling processing preferences between different product types. The priority queueing discipline can either be preemptive resume (PR) or non-preemptive (also called Head of Line (HOL) discipline) based on the service interruption behavior when a higher priority job arrives. Priority queues can further be classified into homogeneous and non-homogeneous based on whether or not service requirements are identical for different classes. The reduced occupancy approximation (ROA) by Morris (1981) was one of the first attempts at modeling preemptive priority disciplines in queueing networks. They used Markov Chain modeling to provide an exact analysis of a two stage closed queueing network with an exponential server at the first stage and an infinite server queue at the second stage. The ROA technique substitutes two dedicated servers with adjusted service rates for every single server priority queue in the network. Improvements to the adjusted service rate used in the ROA technique were suggested by Kaufman (1984) for the lower priority classes. The shadow approximation proposed by Sevcik (1977) is an extension of the reduced work-rate approximation for a more general class of queueing networks. A modification to the exact Mean Value Analysis (MVA) algorithm was presented by Bryant *et al.* (1984) for computing expected wait times for mixed (combination of open and closed routing chains) queueing networks with preemptive or non-preemptive part priorities. A simultaneous system of equations inspired by MVA to analyze steady state performance measures of closed manufacturing networks with part priorities was developed by Shalev-Oren *et al.* (1985). The use of approximate MVA algorithms as an enhancement to Bryant *et al.* (1984) for computational gains was suggested by Eager and Lipscomb (1988).

Later efforts have focused on dynamic assignment of priorities in multi-class queueing networks for scheduling and input control purposes. Brownian control problems have been studied extensively under heavy traffic assumptions for the dynamic scheduling of multi-class closed queueing networks which results in a priority ranking of the classes (Harrison and Wein (1990), Wein (1990), Lee and Sengupta

(1993), Kumar and Kumar (1994)). The stability of multi-class queueing networks under dynamic priority disciplines were studied in Chang (1994), Stolyar (1995), Kumar and Meyn (1995), Dai (1995),Dai and Meyn (1995), Chen (1995), and Chen and Zhang (2000). A fluid model criterion for studying the instability of multi-class queueing networks with dynamic priorities was developed by Dai (1996).

3.2.2 Finite Buffer Queueing Models

The scope of this section is restricted to work on finite open queueing networks with Type I blocking (blocking after service completion). Problem structure was exploited by Gershwin and Schick (1983) to solve the steady state equations associated with the Markov chain model of a three stage transfer line with finite buffers and unreliable machines. Suri and Diehl (1984) presented a *variable buffer* building block for parametric decomposition of finite queueing networks where performance measures for a stage with buffer capacity N are computed as a sum of associated measures for systems with capacity $\{1, 2.., N\}$ weighed by proportion of time the stage buffer remains in state of occupancy $\{1, 2..N\}$. A parametric decomposition method was employed by Altiok and Perros (1987) and Takahashi *et al.* (1980) in combination with the $M/M/1/K$ loss model with pseudo arrival and service rates to analyze finite open networks with Type I blocking. The expansion method for finite open queueing networks was proposed by Kerbache and Smith (1988), where each stage with a finite buffer is *expanded* by adding a virtual stage upstream with a feedback routing arc to replicate the blocking mechanism of the original network. A two server subsystem based decomposition method for the analysis of single class finite open tandem queueing networks resulting in a system of linear equations was proposed by Dallery and Frein (1993). Two moment approximations for series parallel configurations of $M/M/k$ servers based on the expansion method were presented by Jain and Smith (1994). Bounds on performance for reentrant manufacturing lines with finite buffers under different scheduling policy (buffer priority discipline) assumptions were provided by Kumar and Kumar (1994). The queue length process for a single class finite open queueing network with general service and interarrival distributions was shown to converge to a reflected Brownian motion under heavy traffic assumptions by Dai and Dai (1999). A solution to the buffer allocation problem for single class finite open queueing network based on closed form approximations for throughput and work-in-process (WIP) in terms of buffer size was proposed by MacGregor Smith and Cruz (2005). Multistage assembly/disassembly systems were analyzed by Manitz (2008) through a two-moment approximation based on a two-station subsystem decomposition of the original network. The blocking phase was modeled explicitly by Osorio and Bierlaire (2009) and the steady state balance equations for a Markov chain model were solved to develop a single station decomposition approximation of finite open networks with Type I blocking.

The performance measures of queueing networks with finite buffers, Poisson arrivals and general service distributions were studied by Smith (2014) through a two-moment approximation. A rate iterative method based on the generalized expansion method was proposed by Zhang *et al.* (2017a) to estimate performance measures of open queueing networks with type I blocking and general topologies. The correlation between interdeparture times for a two station production line with Poisson arrivals, finite buffer space and phase-type service distributions was studied by Tan and Lagershausen (2017). An iterative optimization algorithm based on sequential quadratic programming was presented by Smith (2018) for simultaneous service rate and buffer size optimization of open queueing networks with exponential servers, Type I blocking and Poisson arrivals. Tandem queues with unreliable servers and type I blocking were analyzed by Shin and Moon (2018) through a decomposition approach.

The performance evaluation of multi-class queueing networks in manufacturing applications have focused on steady state results under Type I blocking assumptions. The study of part priorities in manufacturing contexts have been to a larger extent concentrated on an optimal control framework resulting in a dynamic priority assignment based on system state. Dynamic priority assignment based on current system state is of significance for operational decisions but does not provide a basis for capacity planning for the medium-term to long-term decision making. Priorities based on customer job or product type importance are envisioned. The methods described above address different problems than the one being considered in this chapter and/or require prohibitive computational effort for the problem being addressed by this chapter. The primary interest is in computationally efficient approaches that can quickly provide approximate estimates to identify potential problem areas and allow investigation of various control decisions. The major contribution of this chapter is a computationally efficient numerical approximation that tracks dynamic system status for use in capacity planning in a multiproduct manufacturing setting with possible priorities and finite buffers. In addition to providing descriptive performance evaluation of candidate resource allocation plans in a dynamic environment, potential applications for the proposed approximations include use as an embedded performance evaluator in optimization or surrogate based simulation optimization frameworks.

The rest of this chapter is structured as follows. Incremental approximations which leverage exact results from queueing theory are presented in Section 3.3. The approximations are tested on large scale test instances in Section 3.4 and Section 3.5.

3.3 Performance Evaluation Methods

Heuristic approximations are presented for nonstationary queues and queueing networks with part priorities and finite buffers in this section.

3.3.1 Single Stage Multi-Class Nonstationary Queue with Priorities

There exist known results for the exact steady-state analysis of single server queues with non-homogeneous priorities. Standard results also exist for the exact steady-state results of multi-server queues with homogeneous priorities. The exact analysis of queues with nonstationary arrivals are mathematically intractable as they involve infinite sums of Bessel functions. The notion of steady state does not generally apply to nonstationary queues and queueing networks. However, periodic steady state could exist in special cases. The queue of interest at this point is a multi-class infinite capacity queue with a homogeneous priority scheme and nonstationary Poisson arrivals. The performance of such a queue is approximated by a state model which matches average system state with known steady state results in a incremental fashion over time.

The validity of incrementally matching system state to an equivalent queue with a known exact solution for nonstationary queues has been demonstrated by Tipper and Sundareshan (1990), the Pointwise Stationary Approximation (PSA) (Green and Kolesar (1991)) and the stationary backlog carryover approximation (SBC) (Stolletz (2008)).

The time-step system approximation is detailed below as Approximation 1 (the subscript l is dropped as this algorithm concerns itself with a single stage queue). The operational equations are an extension of the equations developed in Askin and Hanumantha (2018) for FCFS queueing. The model outputs throughput levels (rates) and queue lengths at each point in time. Note that the approximation relies on estimates of the effective service rate at each point in time. Effective service rate is given by the service rate capacity (μ_l) and the presence of one or more jobs with proportional distribution across classes by the number of jobs of each class in the queue.

```
Algorithm: Single stage multi-class queue with homogeneous priority
while $t \leq T$ do
    for $r \in \{1, \ldots, R\}$ do
        $\sigma(t) \leftarrow \frac{\sum_{p=1}^{R} n_p(t)}{\sum_{p=1}^{R} n_p(t)+1}$
        $\sigma_r(t) \leftarrow solution\ to\ \left[\sigma_r^2(t) - \left(\sum_{p=1}^{r} n_p(t) + \sigma(t) + 1\right)\sigma_r(t) + \sum_{p=1}^{r} n_p(t) = 0\right]$
        $\rho_r(t) \leftarrow \sigma_r(t) - \sigma_{(r-1)}(t),\ (\sigma_0(t) = 0)$
        $X_r(t) \leftarrow min\left[\rho_r\mu, \left(\frac{n_r(t)+t_s\lambda_0^r(t)}{t_s}\right)\right]$
        $n_r(t + t_s) \leftarrow max\left[n_r(t) + t_s\left(X_r(t) - \lambda_0^r(t)\right), 0\right]$
    end
    $t \leftarrow t + t_s$
end
```

The overall utilization of the single stage queue is determined in step 4 by inverting the relationship between utilization and WIP for a $M/M/1$ queue. The approximation made here is that the merging of nonstationary Poisson input streams for the different job types results in an overall Poisson input stream. The relation between class level utilization and average WIP of a single stage exponential queue with homogeneous priorities is described through a quadratic equation (see Appendix for derivation). This quadratic equation is solved to estimate class level cumulative utilization in step 5. The existence of an unique and meaningful root of the quadratic equation that corresponds to class level cumulative utilization are shown in the Appendix. The cumulative class level utilization's are then disaggregated into individual class utilization's in step 6. The throughput rate for each product type at time t is obtained in step 7. The first term in step 7 represents station capacity available to a given class, while, the second term accounts for the fact that throughput rate is limited by the availability of jobs during that time step. Available jobs are those present at the start of the time step period and those that arrive during this time step either from external arrivals or internal routing. Finally step 8 updates current queue length as the sum of starting queue and arriving jobs minus the processed and departed jobs. Note at this point infinite buffers are assumed and hence blocking does not occur.

3.3.2 Multi-Class Nonstationary Queueing Networks with Priorities

The numerical approximation presented in section 3.3.1 is extended to an open network of queues with an universal homogeneous priority scheme. This approximation combines a linked period-by-period approach with parametric decomposition to obtain performance measures of interest for dynamic manufacturing systems. The approximation is detailed below. The departure processes at each workstation are assumed to be a renewal process for each priority class, thus allowing merging of these flows to determine effective arrival rates at downstream workstations.

The relation between WIP and utilization for a $M/M/1$ queue with a homogeneous priority service discipline can be expressed as a quadratic equation as shown in step 6 of Approximation 9. The derivation of this expression is presented in the Appendix. This relation is embedded into an incremental scheme to analyze multi-class open networks with priority service. Proposition 1 shows that there exists a determinable solution to the quadratic equation that provides valid results in the queueing network context.

```
Algorithm: Multi-class open network with homogeneous priority
while $t \leq T$ do
    for $l \in \{1 \dots L\}$ do
        $\sigma_l(t) \leftarrow \frac{\sum_{p \in R(l)} n_{pl}(t)}{\sum_{p \in R(l)} n_{pl}(t)+1}$
        for $r \in R(l)$ do
            $\sigma_{rl}(t) \leftarrow$
              $\textit{solution to} \left[\sigma_{rl}^2(t) - \left(\sum_{p=1}^{r} n_{pl}(t) + \sigma_l(t) + 1\right)\sigma_r(t) + \sum_{p=1}^{r} n_{pl}(t) = 0\right]$
            $\rho_{rl}(t) \leftarrow \sigma_{rl}(t) - \sigma_{(r-1)l}(t),\ (\sigma_{0l}(t) = 0)$
            $X_{rl}(t) \leftarrow min\left[\rho_{rl}\mu_l, \left(\frac{n_{rl}(t)+t_s\lambda_{0l}^r(t)}{t_s}\right)\right]$
            for $k \in S(r) \setminus \{0\}$ do
                $\lambda_{kl}^r(t) \leftarrow p_{kl}^r X_{rk}(t)$
            end
            $n_{rl}(t+t_s) \leftarrow max\left[n_{rl}(t) + t_s\left(\sum_{j=0}^{L} \lambda_{jl}^r(t) - \sum_{j=0}^{L} \lambda_{lj}^r(t)\right), 0\right]$
        end
    end
    $t \leftarrow t + t_s$
end
```

Steps 4 through 8 are an extension of the single stage algorithm presented in Section 3.3.1 to each station in the queueing network. The merging of output flows from each station to determine effective arrival rates by product type to downstream workstations is accomplished through step 10. Flow balance equations applied to the average condition of the system are used to update WIP from time step to the next as shown in step 12. Queue length is obtained by a basic material balance equation with a lower limit of 0 for physical stock.

3.3.3 Multi-Class Nonstationary Queueing Networks with Finite Buffers

In this section numerical approximations are presented for finite open queueing networks with nonstationary arrivals and multiple classes. An additional assumption is made that the buffers of stations where products are introduced into the network have infinite capacity, i.e., workstations with external arrivals have infinite buffer capacity. An approximation is proposed based on a linked period-by-period approach combined with parametric decomposition for performance evaluation of such networks. The proposed approximation estimates availability of downstream stations by product type at each station. It combines this information with the probability of each of the product type's $r \in R(l)$ being in service at station l and their process plans to arrive at an estimate of effective availability for the station l.

The basic unit employed for parametric decomposition of the open network is a $M/M/1/K$ loss system with finite system capacity, K. Note that for an input buffer limit of b_l at workstation l, K corresponds to $b_l + 1$ in the proposed model. The average utilization of the loss system is deduced from the average queue length by approximately solving the polynomial equation (3.1) based on the steady-state relation between average utilization ($\rho = \frac{\lambda}{\mu}$) and average number in system (L) for a $M/M/1/K$ loss system. The derivation of equation (3.1) is presented in the Appendix.

$$(L - K)\rho^{K+2} + (1 + K - L)\rho^{K+1} - (L + 1)\rho + L = 0 \tag{3.1}$$

Some key characteristics of $M/M/1/K$ loss queues are exploited to obtain estimates of system utilization associated with various levels of WIP, say, L, for a given system capacity K. Denote this relation as $\rho(L, K)$. Note that the overall utilization of a $M/M/1/K$ loss system can exceed 1 with values ≥ 1 meaning the system is rejecting arrivals. (Note, however, that in the system of interest, such jobs are not "lost" but instead result in blocking and the impact of that blocking may percolate back upstream). This fact is used to replicate the blocking mechanism of the multi-class finite open networks with time-varying arrivals. Denote the idle probability of the $M/M/1/K$ queue computed as a function of $\rho(L, K)$ as $p_0(L, K)$. The function *computeRho* is used to numerically approximate $\rho(L, K)$ over $[0, K]$ in ΔL increments. Figure 3.1 plots the results of the function *computeRho* for a system capacity of 3 computed in increments of $\Delta L = 0.001$. Note that the utilization obtained is a monotonically increasing function of average number in system (L).

$$p_0(L, K) = \frac{1 - \rho(L, K)}{1 - \rho(L, K)^{K+1}} \tag{3.2}$$

$$p_b(L, K) = \rho^K p_0(L, K) \tag{3.3}$$

```
Function computeRho($K$, $\Delta L$):
    for $l \in \{0, 1, \ldots \lceil \frac{K}{\Delta L} \rceil\}$ do
        $\rho(l\Delta L,\ K) \leftarrow$
        smallest real root of eqn. (1): $\rho(l\Delta L,\ K) > \rho((l-1)\Delta L,\ K)$
    end
    return
```

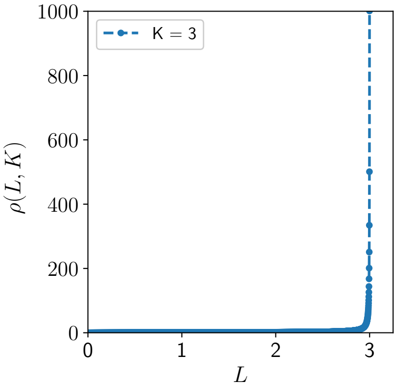


Figure 3.1. $\rho(L, K)$ vs. L for $K = 3$

Denote the downstream workstation for product type r at workstation l as $d(r, l)$. The availability of downstream workstation for product type r at station l is denoted by p_{rl}^{avl}. The algorithm for open networks with finite buffers and nonstationary Poisson arrivals is described below:

```
1  Algorithm: Time-varying multi-class open network with finite buffers
2  while $t \leq T$ do
3      for $l \in \{1 \ldots L\}$ do
4          $\sigma_l(t) \leftarrow 1 - p_0\Big( \sum_{p \in R(l)} n_{pl}(t),\ b_l + 1 \Big)$
5          for $r \in R(l)$ do
6              $\rho_{rl}(t) \leftarrow \frac{n_{rl}(t)\sigma_l}{\sum_{p \in R(l)} n_{pl}(t)}$
7              $p^{avl}_{rl}(t) \leftarrow 1 - p_b\Big( \sum_{p \in R(d(r,l))} n_{pd(r,l)}(t) + \rho_{rl}(t),\ b_{d(r,l)} + 2 \Big)$
8              $X_{rl}(t) \leftarrow min \left[ \left[ \frac{\sum_{i \in R(l)} n_{il}(t)(p^{avl}_{il}(t))}{\sum_{p \in R(l)} n_{pl}(t)} \right] \rho_{rl}\mu_l, \left( \frac{n_{rl}(t)+t_s\lambda^r_{0l}(t)}{t_s} \right) \right]$
9              for $k \in S(r) \setminus \{0\}$ do
10                 $\lambda^r_{kl}(t) \leftarrow p^r_{kl} X_{rk}(t)$
11             end
12             $n_{rl}(t + t_s) \leftarrow max \left[ n_{rl}(t) + t_s \left( \sum_{j=0}^{L} \lambda^r_{jl}(t) - \sum_{j=0}^{L} \lambda^r_{lj}(t) \right), 0 \right]$
13         end
14     end
15     $t \leftarrow t + t_s$
16 end
```

The station level utilization, $\sigma_l(t)$ is computed in step 4 by solving polynomial equation (3.1) for station l which relates utilization to system capacity $(b_l + 1)$ and WIP levels $(n_{rl}(t), r \in R(l))$. The station level utilization is then disaggregated into product type level utilization by using a processor sharing analogy in step 6. The availability of downstream workstation $d(r, l)$ for product type r at station l is calculated in step 7 by solving a polynomial equation for station $d(r, l)$. Step 7 considers a finite queue with capacity $b_{d(r,l)} + 2$ to include the additional server slot of station l. The polynomial equation (3.1) is solved for $d(r,l)$ for WIP level of $\sum\limits_{p \in R(d(r,l))} n_{pd(r,l)}(t) + \rho_{rl}(t)$ which combines WIP level at $d(r,l)$ at time t with average occupancy of the server slot at station l accounted for by product type r. Thus, step 7 effectively introduces an interdependence between station l and each of its downstream workstations $\{d(r,l) : r \in R(l)\}$. The throughput rate of product type r at station l is computed in step 8. The expression $\frac{\sum\limits_{i=1}^{R} n_{il}(t)(p_{il}^{avl}(t))}{\sum\limits_{p=1}^{R} n_{pl}(t)}$ computes overall availability of downstream workstations for station l by using a processor sharing paradigm to estimate the probability of product type r being in service. Flow balance equations are used to update time-varying WIP levels in step 12.

3.4 Results for Networks with Product Priorities

In this section the results of empirical tests are provided on the computational time and accuracy of the proposed numerical approximations for multi-class systems with priorities. Results are shown for both flowshop and jobshop environments.

Experiment 1: A Large Flowshop

Consider a flowshop with four product types ($\{1, \ldots, 4\}$) and sixteen single server unit rate workstations ($\{1, \ldots, 16\}$). Jobs arrive at the system according to a nonhomogeneous Poisson process with interarrival times averaging 1.25 minutes. Consider two nonstationary arrival patterns representing cyclical and short product life cycle (technical innovation) patterns:

1. Sinusoidal with frequency $2\pi = 100$ $mins.$ and phase shift between product types.
2. Triangular over $T = 1000$ $mins.$ with peaks offset by 200 $mins.$

The following five workload configurations are considered for the flowshop described above as shown in Table 3.1. The service rate of the bottleneck workstations is set to 0.9 jobs/min. for the imbalanced configurations as opposed to all unit rate servers in the balanced case.

Configuration	Workload	Bottleneck Station
1	Balanced	-
2	Unbalanced	Workstation 4
3	Unbalanced	Workstation 8
4	Unbalanced	Workstation 13
5	Unbalanced	Workstation 4 and Workstation 13

Table 3.1. Flowshop Configurations

Note that the chosen combination of arrival patterns and service rates result in 80% overall system utilization over the entire time horizon in the balanced configurations and 89% utilization at the bottleneck workstations in the unbalanced configurations.

Experiment 2: A Large Jobshop

As a second experiment, a jobshop with four product types ($\{1, \ldots, 4\}$) and sixteen unit rate workstations ($\{1, \ldots, 16\}$) is considered. Jobs are assumed to arrive at the system according to a nonhomogeneous Poisson process with interarrival times averaging 1.25 minutes and follow a deterministic route as shown in Table 3.2.

Product	Routing
1	$4 \rightarrow 12 \rightarrow 2 \rightarrow 5 \rightarrow 13 \rightarrow 8$
2	$2 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 5 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 16$
3	$1 \rightarrow 8 \rightarrow 10 \rightarrow 9 \rightarrow 11 \rightarrow 13 \rightarrow 14 \rightarrow 4 \rightarrow 16$
4	$4 \rightarrow 5 \rightarrow 3 \rightarrow 8 \rightarrow 10 \rightarrow 6 \rightarrow 9 \rightarrow 12 \rightarrow 1 \rightarrow 15 \rightarrow 16$

Table 3.2. Routing Matrix for Jobshop

The rest of this section presents results of the multi-product priority approximation applied to Experiment 1 and Experiment 2 under the assumptions of priority service discipline. The product types are indexed by priority $\in \{1, 2, 3, 4\}$ where lower index indicates higher processing priority. The performance measures of interest are the expected total throughout and the expected work-in-process (WIP) levels as a function of time. The results of the approximations are compared to 1000 independent replications of reference discrete-event simulations. The nonhomogeneous Poisson processes used to describe product demand are simulated by the *thinning* algorithm applied to homogeneous Poisson processes (Banks *et al.* (2005)). The relative errors presented in the rest of this section are calculated as shown below. Reported values are average normalized relative error equally weighted over the $\frac{1000}{t_s}$ time intervals. Normalized relative error (NRE) at any time t is given by

$$\frac{max(0, |\theta_{APPROX}(t) - \theta_{EXT}(t)| - \epsilon)}{\theta_{EXT}(t) + \epsilon} * 100 \tag{3.4}$$

where $\theta_{EXT}(t)$ is the exact value at time t (based on the average of the simulation runs), and $\theta_{APPROX}(t)$ is the approximate estimate at time t. The user specified constant ϵ is used to guard against the possibility of very small values in the denominator from skewing the overall relative error. The value of the user specified constant used in this chapter is $\epsilon = 10^{-3}$ as it is reasonable to assume that deviations at average WIP levels or average total throughput less than 0.001 jobs are not of practical significance.

3.4.1 Total Throughput

Figures 3.2 and 3.3 display the total throughput for each product type for the balanced configurations of the flowshop with sinusoidal and triangular demand patterns respectively. The plots are very similar for all other experiments and hence have been omitted to avoid redundancy. The solid line represents the simulation estimates obtained from 1000 independent replications and the dashed line represents the corresponding approximation estimate. The figures clearly show that the approximation works well for evaluation of expected total throughput over time. The approximation captures the magnitude and the trend of total throughput with reasonable accuracy. The relative errors displayed in Table 3.3 are within 10% of simulation estimates for bulk of the experiments performed. A progressive reduction in accuracy is observed as one moves from the highest priority class to the lowest. The errors for the lowest priority class of the flowshop experiments with triangular arrivals are particularly large (> 10%). Errors for the top priority product, presumably the most important, are 1.2% at most and frequently less than 1%.

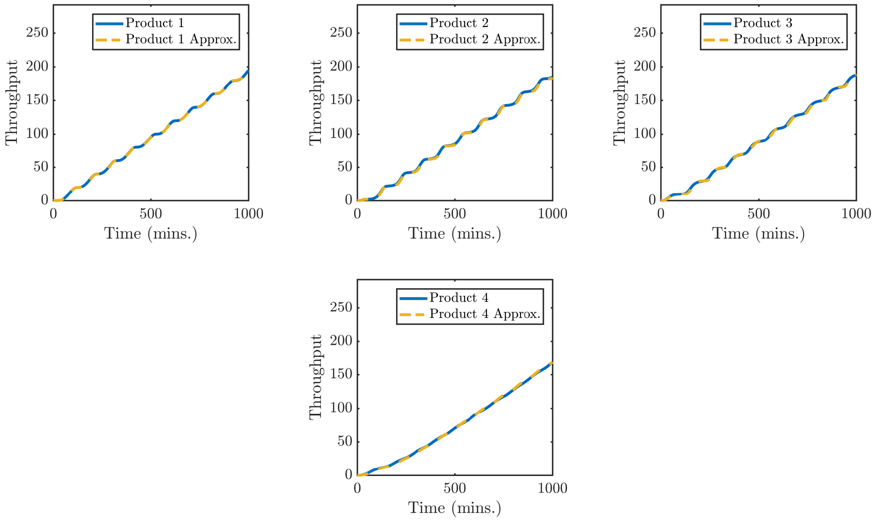


Figure 3.2. Total Throughput by Product Type: Sinusoidal Arrivals

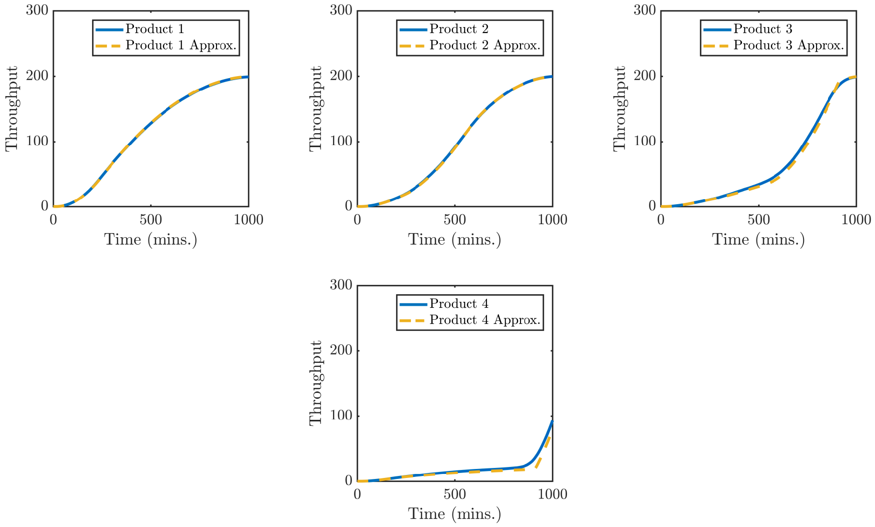


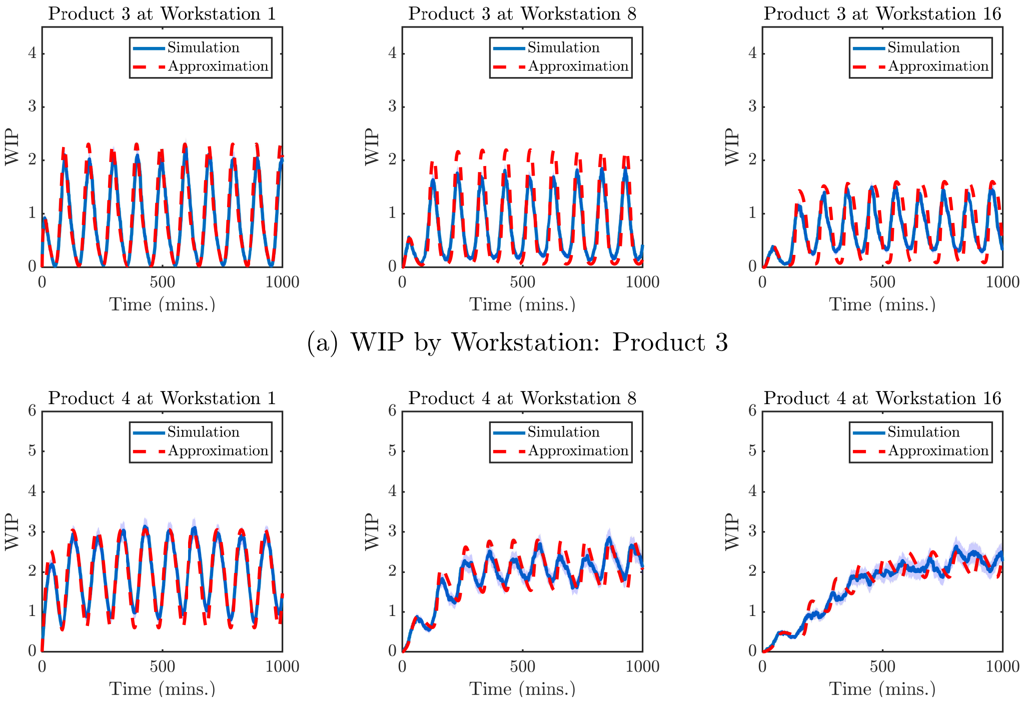
Figure 3.3. Total Throughput by Product Type: Triangular Arrivals

Experiment	Arrivals	Configuration	Product			
			1	2	3	4
1	Sinusoidal	1	1.14	2.74	2.71	5.29
1	Sinusoidal	2	1.07	1.81	2.89	6.49
1	Sinusoidal	3	1.21	1.99	2.65	6.30
1	Sinusoidal	4	0.96	2.36	2.96	5.84
1	Sinusoidal	5	1.09	1.97	2.91	6.90
Average			**1.09**	**2.17**	**2.82**	**6.16**
1	Triangular	1	0.84	1.38	5.36	11.70
1	Triangular	2	0.73	1.63	5.87	11.71
1	Triangular	3	1.01	1.22	6.13	9.61
1	Triangular	4	0.59	1.48	6.85	10.93
1	Triangular	5	0.62	1.91	6.53	11.46
Average			**0.76**	**1.52**	**6.15**	**11.08**
2	Sinusoidal	-	0.63	0.83	1.09	4.66
2	Triangular	-	0.62	0.76	1.68	8.24

Table 3.3. NRE for Total Throughput (%)

3.4.2 Work-In-Process Levels

Figure 3.4 displays the expected WIP levels by product type for the balanced flowshop configurations with sinusoidal demand patterns. The solid line represents simulation estimates from 1000 independent replications while the dashed lines represent the approximation estimates. The approximation performs extremely well for the highest priority class, though, a progressive reduction in accuracy observed as one moves from the highest priority class to the lowest. In the sinusoidal demand pattern case it is observed that the approximation captures the frequency of WIP variation over time but overestimates the amplitude of the WIP levels. In the triangular case similar patterns are observed with the degree of overestimation progressively increasing for product types with lower priority.

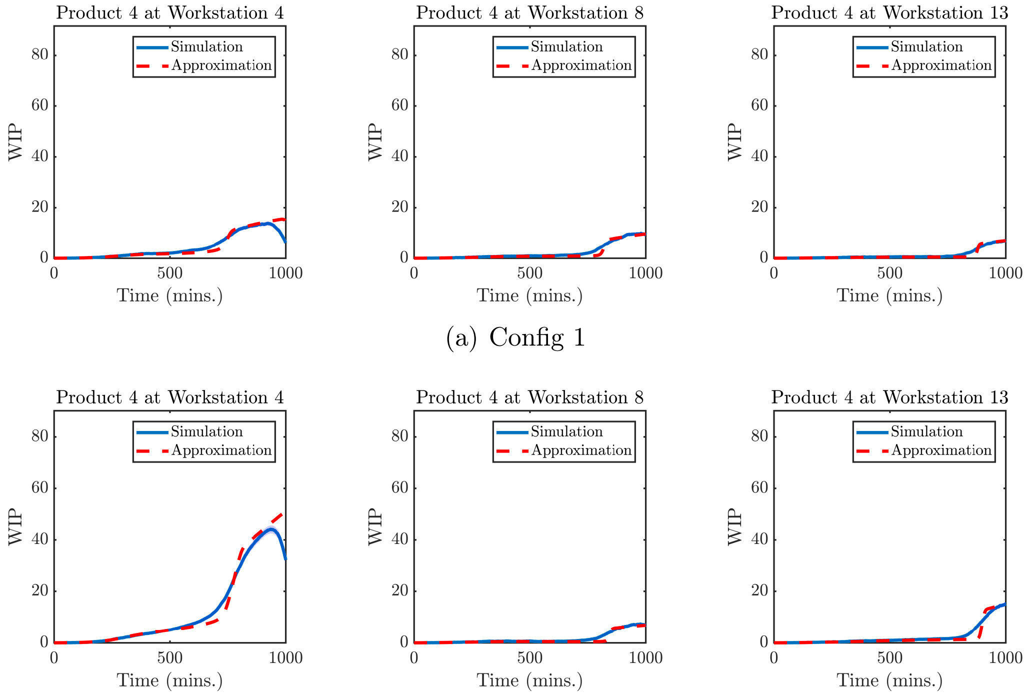


(a) WIP by Workstation: Product 3

(b) WIP by Workstation: Product 4

Figure 3.4. WIP Levels by Workstation: Flowshop with Sinusoidal Arrivals

Figure 3.5 plots the WIP levels for product type 4 at workstations 4, 8 and 13 for configurations 1 and 5 of the flowshop with triangular demand patterns. The approximation adequately captures the effects of introducing imbalanced workloads in the flowshop although the same pattern of overestimation of peak WIP levels persists.

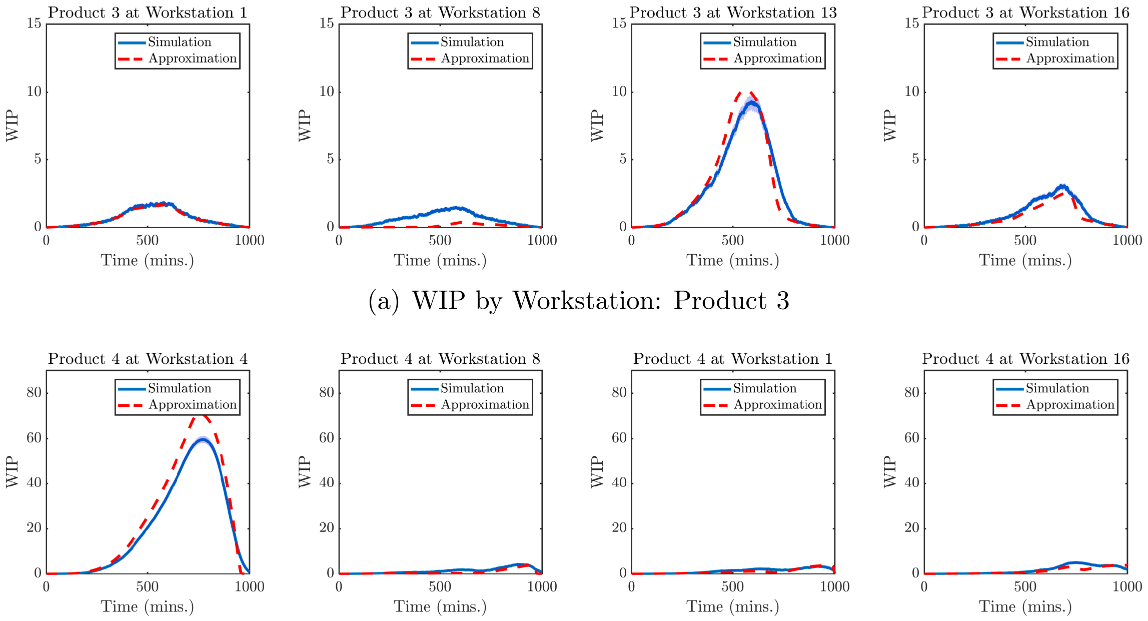


(a) Config 1

(b) Config 5

Figure 3.5. Product 4 WIP Levels by Flowshop Configuration: Triangular Arrivals

Figure 3.6 plots WIP levels at workstations $1, 4, 8, 13$ and 16 for the jobshop in Experiment 2 with sinusoidal and triangular demand patterns respectively . The plots are arranged in the order of processing sequence for each product type. For the jobshop the priority discipline is assumed to be fixed across all workstations but a workstation is visited by a subset of all product types. Apart from the patterns observed for Experiment 1 it is noticed that the approximation underestimates WIP levels at workstation 8 in figure 3.6(a).



(a) WIP by Workstation: Product 3

(b) WIP by Workstation: Product 4

Figure 3.6. Work-In-Process by Workstation: Jobshop with Triangular Arrivals

The relative errors for WIP levels for all twelve experiments (10 flowshop configurations and 2 jobshop configurations) are summarized in Table 3.4. Specifically, Table 3.4 reports the average, minimum and maximum NRE for workstations $1, 4, 8, 13$ and 16 by product calculated across the twelve experiments.

		Workstation				
		1	4	8	13	16
Product 1	Min	5.36	5.41	5.45	5.32	6.00
	Avg.	6.36	7.98	9.36	7.97	8.07
	Max	7.71	11.23	13.96	11.27	10.46
Product 2	Min	5.58	6.15	6.72	6.32	6.41
	Avg.	7.96	9.76	9.80	15.57	15.90
	Max	11.41	13.70	17.65	24.40	27.46
Product 3	Min	7.30	11.15	12.01	13.59	13.90
	Avg.	13.32	22.90	34.58	28.05	27.04
	Max	15.55	32.09	77.73	43.79	43.77
Product 4	Min	10.42	10.33	11.34	13.90	13.84
	Avg.	19.89	14.85	20.98	19.09	21.70
	Max	33.48	24.73	51.46	27.00	34.47

Table 3.4. Relative Error for WIP by Product and Workstation(%)

3.4.3 Computation Times

Table 3.5 displays the computation times for a time step of $t_s = 0.125$ min. The running time for all experiments performed were under one second on a laptop with 8GB of RAM. In contrast, 1000 independent simulation replications take about 90 minutes when run in parallel on a laptop with 2 cores and 16 GB of RAM. Thus, the objective of computational feasibility for rough-cut estimation and scenario analysis is achieved.

Experiment	Arrivals	Configuration	Time (s)
1	Sinusoidal	1	0.754
1		2	0.813
1		3	0.711
1		4	0.731
1		5	0.689
1	Triangular	1	0.721
1		2	0.682
1		3	0.783
1		4	0.697
1		5	0.808
2	Sinusoidal	-	0.723
2	Triangular	-	0.675

Table 3.5. Computation Times (Seconds)

3.5 Results for Finite Buffer Networks

In this section, the focus is limited to WIP levels and probability of blocking estimates while presenting the results of the finite buffer approximation. Empirical experience indicates that the performance of the approximations for throughput estimation are very accurate. Normalized relative errors are consistently below 10% deviation for all product types.

3.5.1 Experiment Design

A designed experiment is performed to study the accuracy of the finite buffer approximation under a wide range of parameter and configuration assumptions. Consider a four-factor factorial experiment with two or three levels of each factor to study the impact of buffer size, overall utilization, queueing network topology and configuration, and, demand patterns. The factorial experiment is described in Table

3.6. Utilizations represent light and moderate to heavy workload. Buffer sizes vary from being unconstrained to allowing evaluation with relevant blocking probabilities. Arrival pattern is chosen to be cyclical with offsets that either reinforce or partially cancel arrivals and triangular for modeling new product introductions and demand expiration (product life cycles). The configuration defines whether the shop is a balanced flowshop (no specific bottleneck), a flowshop with two bottlenecks, or a general job shop.

Factor	Level 1 (L1)	Level 2 (L2)	Level 3 (L3)
Average Utilization, u	0.4	0.8	–
Buffer Size, b	Infinite	3	5
Arrival Pattern	Sinusoid, offset $= \pi/4$	Sinusoid, offset $= \pi/8$	Triangular, offset $= T/5$
Configuration	Balanced Flowshop	Unbalanced Flowshop	Jobshop

Table 3.6. Factorial Experiment

The jobshop configuration in the designed experiment is assumed to have a routing matrix as displayed in Table 3.7. The routing matrix is chosen carefully to avoid the deadlock phenomenon commonly experienced in discrete-event simulation of finite open networks with loops.

Product	Routing
1	$3 \rightarrow 1 \rightarrow 6 \rightarrow 9 \rightarrow 10 \rightarrow 11$
2	$2 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 5 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 16$
3	$12 \rightarrow 1 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 10 \rightarrow 14 \rightarrow 8$
4	$3 \rightarrow 1 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 14 \rightarrow 16$

Table 3.7. Jobshop Routing

A total of 54 experimental runs are performed as part of the designed experiment. For each experimental run in the designed experiment simulation estimates of WIP levels averaged across 500 independent discrete-event simulation replications are compared against approximation results. In the study of probability of blocking, simulation estimates averaged across 2000 independent discrete-event simulation replications are compared against approximation results.

The factorial experiment is used to study the performance of the proposed approximation at four different stages of each product type's process plan - at the entry workstation (designated *Entry*), the first finite station (designated *Start*), a station in the middle of its route (designated *Middle*) and the last station (designated *End*). Note that the *Entry* station for each product has infinite buffer capacity to preserve the non-homogeneous Poisson arrival process. The relative errors are averaged across all product types to understand how well the approximations perform at the chosen four stages. The primary interest is in main effects and two-way interactions and thus higher degree interactions are dropped for the purpose of this designed experiment. Table 3.8 reports the average and the standard deviation of NRE across all product types by processing stage for each of the designed experiment factors at each of their levels. For instance, the average NRE over all cases for the Balanced Flowshop, i.e. Configuration Level 1, was 19.17%. Note that WIP level errors are less than 20% in all but a few cases. In addition, there is no evidence of deterioration in the quality of the estimates as products progress further in their route.

Stage	Factor	L1		L2		L3	
-	-	Avg.	Stddev.	Avg.	Stddev.	Avg.	Stddev.
Entry	config	19.17	7.92	17.89	5.95	24.59	18.21
	buffer	17.57	5.43	23.67	17.55	20.40	9.96
	utilization	15.49	2.74	25.61	15.42		-
	arrivals	21.31	9.37	22.42	6.97	17.91	17.56
Start	config	15.73	3.83	14.89	4.01	20.14	8.40
	buffer	16.18	4.55	18.12	7.43	16.46	6.33
	utilization	15.58	2.90	18.26	8.08		-
	arrivals	19.02	7.22	19.06	5.59	12.69	2.55
Middle	config	15.01	5.33	14.95	4.71	17.92	5.70
	buffer	13.68	4.66	17.90	5.90	16.30	4.79
	utilization	16.41	3.42	15.51	6.79		-
	arrivals	18.32	5.69	17.82	4.75	11.74	2.41
End	config	15.59	5.60	15.54	5.37	17.92	6.79
	buffer	13.76	4.75	18.79	6.69	16.30	5.65
	utilization	16.68	3.67	16.59	7.79		-
	arrivals	19.73	6.51	18.33	5.25	11.74	2.48

Table 3.8. NRE of WIP Levels for Designed Experiment

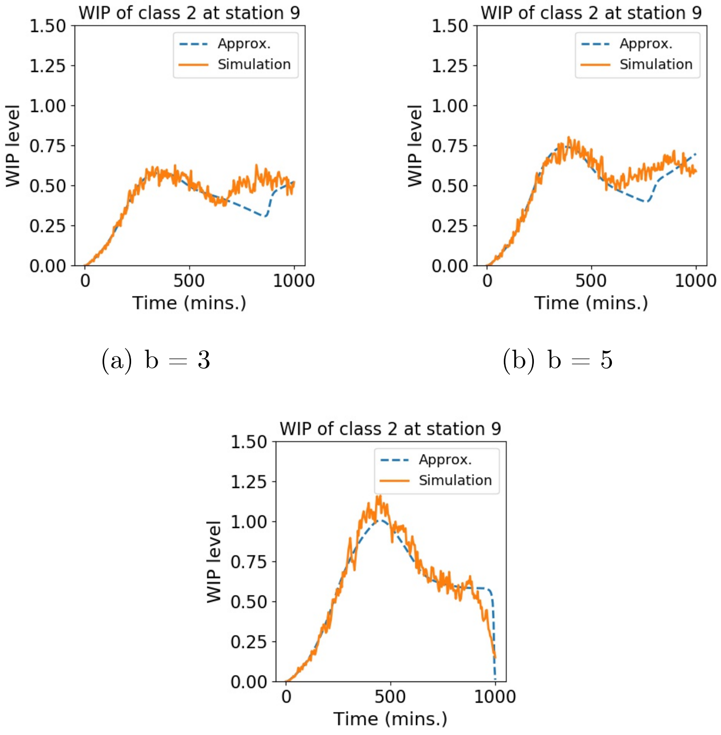
The Analysis of Variance (ANOVA) and the effects test for the factorial experiment are documented in Appendix A.3. The factorial experiment provides statistical evidence to support the fact that for the stage *Entry* where jobs enter the network, the accuracy of the approximation is driven by the configuration, utilization and their two-way interaction. The two-way interaction between arrival pattern and configuration also has a statistically significant effect on the accuracy of the approximation for the stage *Entry*. For the stages, *Start, Middle, End* the factorial experiments show that all single factor effects and two-way interactions are statistically significant with the exception of the two-way interactions between configuration and buffer size, configuration and arrival patterns, and between buffer size and arrival patterns.

3.5.2 Discussion of Results

The following observations are made with regards to various aspects of the application of the numerical approximation to nonstationary queueing networks with finite buffers. These observations are discussed through the use of appropriate numerical instances.

Observation 1: Buffer size

The impact of buffer size on the performance of the Finite Buffer Approximation is studied in the subsequent discussion. The *Middle* station for product type 2 in the jobshop configuration of the factorial experiment is employed to study the impact of buffer size on the accuracy of the approximation. Figure 3.7 presents the accuracy as a function of buffer size for three different cases. Figure 3.7 demonstrates that the approximation tracks the time-varying pattern of WIP accurately across all buffer sizes. The approximation is also able to capture the magnitude of WIP reasonably accurately for all buffer sizes. The finite buffer approximation also does a reasonably good job of capturing the increase in WIP from buffer size, $b = 3$ to the infinite buffer case. The approximation generally seems to lag the simulation estimates for all buffer capacities with the lag in estimates improving as the buffer size increases.



(a) b = 3

(b) b = 5

(c) b = ∞

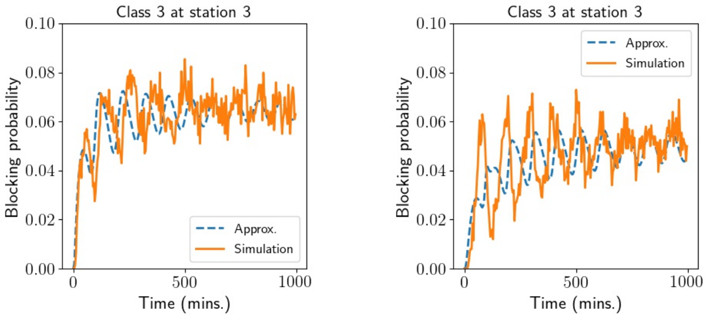
Figure 3.7. Impact of Buffer Size on Finite Buffer Approximation

Observation 2: Probability of blocking

In this discussion, the focus is on the ability of the approximation to accurately estimate the probability of being blocked for a given workstation in the network. Blocking can occur when one of the downstream workstations has a full buffer. The probability of a blocked downstream workstation by product type r at workstation l at time t, $Pr^b_{rl}(t)$, is given by the below expression.

$$Pr^b_{rl}(t) = \frac{n_{rl}(t)(1 - p^{avl}_{rl}(t))}{\sum\limits_{i \in R(l)} n_{il}(t)} = \frac{n_{rl}(t)p_b\Big(b_{d(r,l)} + 2, \sum\limits_{p \in R(d(r,l))} n_{pd(r,l)}(t) + \rho_{rl}(t)\Big)}{\sum\limits_{i \in R(l)} n_{il}(t)} \quad (3.5)$$

The expression in equation 3.5 combines the probability of product type r being in service based on a processor sharing paradigm with the probability that the downstream workstation for product type r has a full buffer. These probabilities are examined for the unbalanced flowshop configuration in the factorial experiment. The upstream workstation adjacent to the bottleneck workstation is studied. Consider product type 3 at workstation 3 to illustrate the results. Figure 3.8 plots the blocking probability approximation (L.H.S of equation 3.5) against simulation estimates. Simulation estimates are computed by tracking the product type and downstream workstation state (available or blocked) at every service completion event for 2000 independent replications. The proportion of replications where the downstream workstation is blocked across 2000 replications computed by product type is used to obtain simulation estimates. Figure 3.8 indicates that the finite buffer approximation is able to capture the time-varying congestion trends reasonably accurately at various buffer capacities. The approximation is also able to capture the increase in congestion as the buffer capacity is reduced from $b = 5$ to $b = 3$.

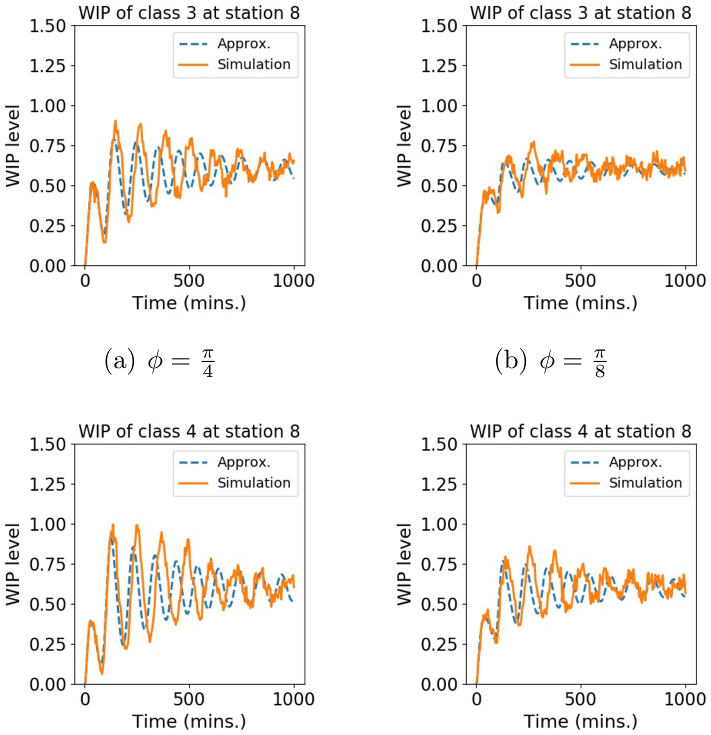


(a) b = 3 (b) b = 5

Figure 3.8. Probability of Blocking for Finite Buffer Approximation

Observation 3: Product mix

The effect of the relation between the time-varying arrival rate functions for different product types on the accuracy of the proposed approximations are explored in the following discussion. The balanced flowshop configuration is examined for the sinusoidal arrival patterns. The arrivals with a phase offset of $\frac{\pi}{4}$ are compared against arrivals with a phase offset of $\frac{\pi}{8}$. The *Middle* workstation of product types 3 and 4 of the balanced flowshop configuration is studied for this discussion. Figure 3.9 demonstrates that the finite buffer approximation is able to adapt well to different frequencies of time-varying patterns. The approximation is also able to perform well for all classes. However, observe the approximation leads the simulation estimates in both cases with the extent of lead increasing from offset of $\frac{\pi}{8}$ to offset of $\frac{\pi}{4}$.



(a) $\phi = \frac{\pi}{4}$

(b) $\phi = \frac{\pi}{8}$

(c) $\phi = \frac{\pi}{4}$

(d) $\phi = \frac{\pi}{8}$

Figure 3.9. Impact of Product Mix on Finite Buffer Approximation

Observation 4: Computational complexity and choice of time step

The computational complexity of all the approximations presented are linear in the number of product types R, number of stages L and the number of time steps chosen for the approximation $\lceil \frac{T}{t_s} \rceil$. The overall worst-case computational complexity of the finite buffer approximation is $O\big(RL\lceil \frac{T}{t_s} \rceil\big) + O\big(\frac{B^4}{\delta_L}\big)$. The first term accounts for the computational cost of the numerical approximation while the second term accounts for the cost of solving the polynomial equation in increments of δ_L for the M/M/1/K loss system assuming the largest finite buffer capacity in the network is B. The complexity of approximately solving the polynomial equation

3.1 is $O(B^3)$ for a fixed WIP level since it involves identifying the eigenvalues of the companion matrix of the polynomial equation. The polynomial equation and its companion matrix are discussed in detail in Appendix A.2. The polynomial equation is solved at $\frac{B}{\delta_l}$ WIP levels thus resulting in $O\left(\frac{B^4}{\delta_L}\right)$ operations. Empirically, the approximations are efficient with runtime under one minute for the mid-sized systems studied for time-step $t_s = 0.125min$ and a 1000 minute problem horizon. The accuracy of the finite buffer approximation shows decaying marginal improvement as the time step size is decreased with a trade-off between time-step size, t_s and runtime.

3.6 Case Study: An Airline Security Screening Checkpoint

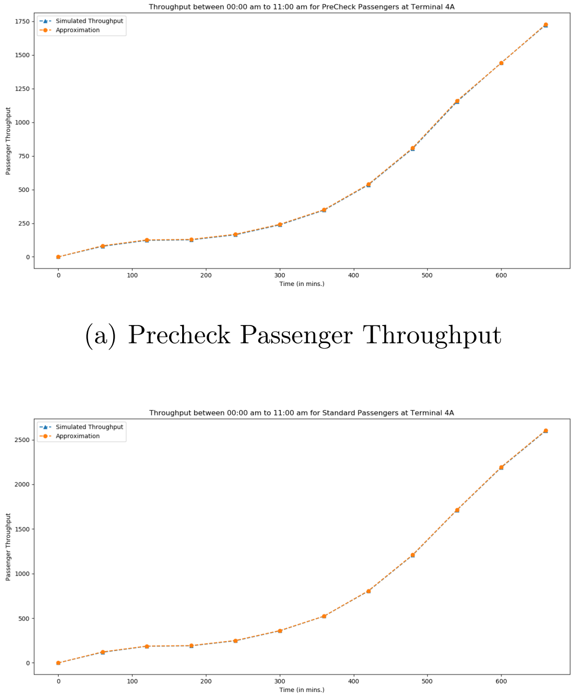
As an additional test, the finite buffer approximation presented in Section 3.3.3 is applied to an airport security screening checkpoint to estimate passenger throughput and queue lengths. The Phoenix Sky Harbor Airport handled about 45 Million passengers in 2018 with over 453000 take-off and landing operations. It has three major terminals with six security screening checkpoints operating across these three terminals. The security checkpoint at terminal 4A, primarily in use for American Airlines flights, is one of the largest checkpoint operations at Sky Harbor Airport in terms of passenger volumes experienced. Each Transportation Security Administration (TSA) security screening checkpoint (SSCP) conforms to a standard layout as prescribed in the Checkpoint Design Guide. Passengers enter the checkpoint into a queue for travel document verification where a Transportation Security Officer (TSO) compares the traveler's ID card with their boarding pass and a visual passenger check. Passengers are then routed to the X-ray screening lanes where separate screening is conducted for the passenger and their belongings. The X-ray stage could possibly be followed by a secondary screening stage as dictated by TSA standard operating procedures where additional screening such as explosive trace detection, bottled liquids scanning or alternate viewing stations might be employed.

The finite buffer approximation presented in Section 3.3.3 is applied to Terminal 4A to understand the impact of finite queueing room between the travel document checkers (TDCs) and the X-ray lanes. Terminal 4a has 5 TDC stations and 8 Baggage Screening X-ray lanes. (Note that each pair of X-ray lanes for baggage checking is accompanied by a Walk-Through Metal Detector or Advanced Imaging Technology (WTMD/AIT) station for passenger screening). The major classes of passengers are Precheck passengers and standard passengers. The throughput and queue buildup at the checkpoint is studied for 11th January 2017 between the hours of 00:00 am to 11:00 am as this captures the peak period of the day. A piecewise linear arrival function is fitted to historical throughput data to estimate the time-varying Poisson arrival rate function for passenger arrivals to the security checkpoint. Passengers are assumed to choose the shortest open X-ray lane after they pass through the TDC stage. The finite buffer approximation is modified to fit the single class fork topology of the SSCP network. The throughput of the TDC stage is first calculated without accounting for blocking by the X-ray stage assuming there is ample queueing room at the X-ray stage. The inflow to the X-ray stage is then determined by combining availability (probability of having queueing room) with the probability that a given X-ray lane has the shortest queue length.

Three experiments are performed to study the impact of various decisions at the SSCP.

1. Base configuration with infinite queueing room at X-ray lanes.
2. Finite queueing room at X-ray lanes. X-ray queue capacity set to 4 customers.
3. Impact of closing an X-ray lane.

The results of the three experiments are presented below. Figures 3.10 plot the throughput estimates for Standard and Precheck passengers at the SSCP. The dashed lines represent the approximation estimates while the solid lines represent the estimates obtained from 1000 replications of a discrete event simulation. The plots indicate that the approximation tracks the SSCP throughput very accurately.

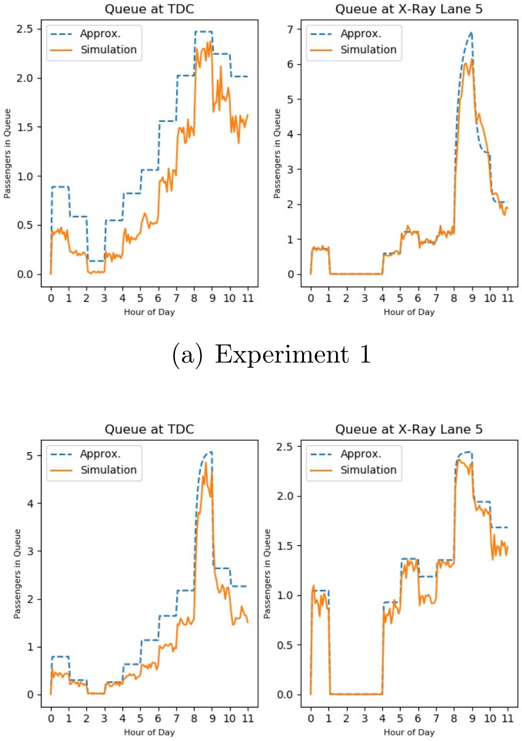


(a) Precheck Passenger Throughput

(b) Standard Passenger Throughput

Figure 3.10. Throughput for Precheck and Standard Passengers

Figure 3.11 displays the TDC queue lengths and the queue lengths at at X-ray lane 5 (chosen for this discussion as it is open most of the time). The plots indicate that the finite buffer approximation is able to accurately capture the queue buildup at the TDC stage under the assumption of finite queueing room at the X-ray lanes.



(a) Experiment 1

(b) Experiment 2

Figure 3.11. Impact of Finite Buffers and Lane Open/Close Policies

Figure 3.12 displays the impact of closing an X-ray lane for the entire duration between 0:00 am to 11:00 am. The figure shows that the approximation is adequately able to capture the queue build-up at the TDC stage and the corresponding increase in average queue length at X-ray lane 5 as a result of the X-ray lane closure.

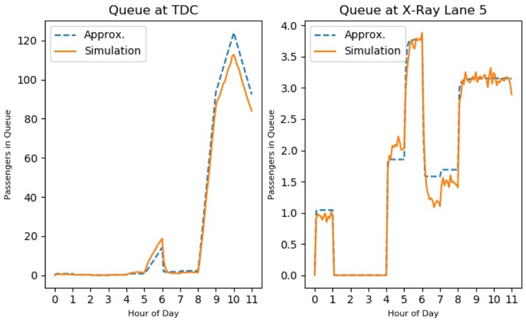


Figure 3.12. Impact of Finite Buffers and Lane Open/Close Policies

Chapter 4

DYNAMIC SERVER ALLOCATION FOR MULTI-PRODUCT MANUFACTURING SYSTEMS WITH NONSTATIONARY DEMAND

In this chapter, an optimization model is presented for dynamic server allocation between workstations in multi-product manufacturing systems where workstations may have more than one server. A multi-server extension of the open network algorithm for dynamic manufacturing systems with infinite buffers is proposed. The multi-server approximation is reformulated as a mixed integer non-linear optimization problem for dynamic server allocation. Heuristic approaches are presented to solve the optimization problem for practically-sized manufacturing systems.

4.1 Background on Optimal Server Allocation

The existing work on allocation of servers in manufacturing systems and queueing networks in general is reviewed in this section. Much of the existing work is based on one of two approaches. The first approach treats the server allocation problem as an optimization problem. The second approach models the server allocation problem as a Markov Decision Process and optimal policies are examined.

Shanthikumar and Yao (1987) modelled the optimal server allocation problem for closed queueing networks with state dependent Poisson arrivals and exponential service as an optimization problem. The exploited the concavity property of throughput for closed queueing networks. Dallery and Frein (1988) proposed a heuristic and exact approach for the server allocation problem in multi-server closed queueing networks.

Shanthikumar and Yao (1988) formulated a mixed-integer nonlinear program for maximizing throughput in multi-server closed queueing networks. They employed bounding techniques to inform a search procedure for solving the proposed optimization problem. Boxma *et al.* (1990) and Frenk *et al.* (1994) studied the server allocation problem in a multi-product manufacturing system by analyzing the aggregated product under product-form assumptions for the associated queueing network. They proposed an iterative algorithm for solving the server allocation problem. Tassiulas and Ephremides (1993) studied a queueing system with N parallel queues competing for the attention of a single server. They examined conditions for stability and proposed optimal policies for maximizing throughput. Hillier and So (1996) examined the joint effect of allocation of servers and load balancing in production lines by modelling them as tandem multi-server queues with finite buffers in between stations. Palmer and Mitrani (2005) characterized the optimal policy for allocating M job classes each waiting in a dedicated queue to a fixed number of processors in a computing grid. Kittipiyakul and Javidi (2009) studied server allocation policies for N statistically identical queues connected to K servers with time-varying and stochastic arrivals and connectivity. They studied the interaction between competing objectives of load-balancing and maximizing throughput in time-slotted framework. Smith *et al.* (2010) employed two moment approximations for finite open queueing networks with general service distributions at workstations. Smith and Barnes (2015) proposed a decomposition based approximation for the performance analysis of closed queueing networks with multiple servers at each service station. The approximation was embedded in an optimization problem to suggest an optimal allocation of servers in the network.

The majority of the work on optimal server allocation in manufacturing systems have made stationary arrival assumptions and very little attention has been devoted to optimal server allocation in multi-product manufacturing systems with time-varying demands. Framing the server allocation problem as an optimal control problem has been limited to parallel queues feeding into a common pool of servers. The applicability of optimal control approaches to server allocation problems in time-varying queueing networks is very limited due to the complexity of characterizing the departure processes from workstations. Furthermore, it is difficult to reasonably rationalize the Markov property for state transitions in time-varying queueing networks with multiple customer classes.

In the rest of this chapter, the server allocation problem in multi-product manufacturing systems with time-varying demands is viewed as an optimization problem in a time-slotted framework. The configuration changes caused by the reallocation of servers between workstations are assumed to be allowed at specific points in time and these points in time are assumed to be known beforehand.

4.2 Problem Definition and Mathematical Formulation

The server allocation problem in a multi-product manufacturing system is defined in this section. A nonlinear optimization model is presented and heuristic approaches to efficiently solve the optimization problem are presented.

4.2.1 Multi-Sever Open Network Based Throughput Model (ONBTM)

The open network based approximation for dynamic manufacturing systems with infinite waiting room is extended to workstations with multiple servers.

```
1 Algorithm: Multi-Server Open Network Based Throughput Model (ONBTM)
2 Initialize $n_{rl}$.
3 while $t \leq T$ do
4     $X_{rl}(t) \leftarrow min\left[\left(\frac{n_{rl}(t)}{\sum\limits_{p\in R(l)} n_{pl}(t)+1}\right) min(c_l, \lceil \sum\limits_{p\in R(l)} n_{pl}(t) \rceil)\mu_l, \left(\frac{n_{rl}(t)+t_s\lambda^r_{0l}(t)}{t_s}\right)\right]$
5     $\lambda^r_{kl}(t) = p^r_{kl}X_{rk}(t),\ k \in S(r) \setminus \{0\}$
6     $n_{rl}(t+t_s) \leftarrow max\left[n_{rl}(t) + t_s\left(\sum\limits_{j=0}^{L}\lambda^r_{jl}(t) - \sum\limits_{j=0}^{L}\lambda^r_{lj}(t)\right), 0\right]$
7     $t \leftarrow t + t_s$
8 end
```

The multi-server ONBTM algorithm introduces a correction term for the throughput rate computation in Step 4. The expression $min(c_l, \lceil \sum\limits_{p\in R(l)} n_{pl}(t) \rceil)$ accounts for the fact that when there are fewer jobs in the system than the available number of servers the service rate is dictated by the number of jobs in system. The other aspects of the multi-server ONBTM algorithm such as WIP updates and effective arrival rate computations are similar to the single server version of the ONBTM algorithm. The multi-server ONBTM algorithm is employed in subsequent parts of this chapter to determine the time-varying allocation of servers at workstations.

4.2.2 Server Allocation in Dynamic Manufacturing Systems with Fixed Number of Servers

The multi-server approximation presented in section 4.2.1 is reformulated as a mixed integer non-linear optimization problem. The primary decisions of interest are the number of servers employed at each multi-server workstation, $c_l(t)$ with an objective of minimizing the maximum holding cost of work-in-process jobs incurred at any workstation over the entire time horizon. The optimization formulation is presented below. The set of workstations is denoted by L. The set of product types is denoted by R. The set of workstations visited by a product of type $r \in R$ is given by $S(r)$, while, the set of products visiting a workstation $l \in L$ is denoted by $R(l)$. The set of time periods when configuration changes occur is denoted by T. Let $q_{rl}(t)$ denote the holding cost incurred at station $l \in L$ for a product type $r \in R(l)$ in time period $t \in T$. The fixed total number of servers to allocated in each time period is denoted by S.

$$\min_{r\in R,\ l\in S(r),\ t\in T} \max \quad q_{rl}(t)n_{rl}(t) \tag{4.1}$$

$$\text{subject to} \tag{4.2}$$

$$X_{rl}(t) \leq \Big(\frac{n_{rl}}{\sum\limits_{r\in R(l)} n_{rl}+1}\Big)c_l(t)\mu_{rl} \quad \forall\, l \in L,\ r \in R(l),\ t \in T \tag{4.3}$$

$$X_{rl}(t) \leq \Big(\frac{n_{rl}}{\sum\limits_{r\in R(l)} n_{rl}+1}\Big)\lceil \sum_{p\in R(l)} n_{pl}(t)\rceil\mu_{rl} \quad \forall\, l \in L,\ r \in R(l),\ t \in T \tag{4.4}$$

$$X_{rl}(t) \leq \frac{n_{rl}(t)+t_s\lambda^r_{0l}(t)}{t_s} \quad \forall\, l \in L,\ r \in R(l)\ t \in T \tag{4.5}$$

$$n_{rl}(t+t_s) \geq n_{rl}(t) + t_s\left(\sum_{j=0}^{L} \lambda_{jl}^{r}(t) - \sum_{j=0}^{L} \lambda_{lj}^{r}(t)\right) \quad \forall\, l \in L,\ r \in R(l)\ t \in T \tag{4.6}$$

$$\lambda_{kl}^{r}(t) = p_{kl}^{r} X_{rk}(t)\ \forall\ r \in R,\ k \in S(r) \setminus \{0\},\ t \in T \tag{4.7}$$

$$\sum_{l \in L} c_l(t) \leq S,\ \forall t \in T \tag{4.8}$$

$$c_l(t) \in \mathbb{Z}^{+}, \forall l \in L, t \in T \tag{4.9}$$

$$X_{rl}(t) \geq 0,\ \forall l \in L, r \in R(l), t \in T \tag{4.10}$$

$$n_{rl}(t) \geq 0,\ \forall l \in L, r \in R(l), t \in T \tag{4.11}$$

The objective function in equation 4.1 minimizes the maximum holding cost incurred over the entire planning horizon. Clearly setting the $q_{rl}(t) = 1$ will minimize the maximum queue length experienced at any workstation for any product type at any epoch. Alternative objectives could readily be substituted. For instance Equation 4.12 would minimize total holding cost.

$$\min \sum_{r \in R, l \in S(r), t \in T} q_{rl}(t) n_{rl}(t) \tag{4.12}$$

The constraint 4.3 limits throughput rate by server capacity. Constraint4.4 limits throughput rate by the number of jobs in system times the capacity of each server. This constraint is relevant when there are fewer jobs in system than the total number of available servers. Constraint 4.5 limits the throughput rate by the expected number of jobs in a given time window. The expected number of jobs values are computed by combining expected arrivals (governed by a Non-homogeneous Poisson Process) with the WIP level at the start of the time period. Thus, constraints 4.3 through 4.5 capture the fact that throughput rate is limited by available capacity, number of jobs in system and expected number of arrivals in a given time period. The flow balance equations are captured by 4.6. The total number of servers available to be allocated at any time period is captured in constraint 4.8.

An alternative formulation for the allocation of servers in dynamic manufacturing systems could incorporate time varying costs of allocating a server to a workstation under a total budget for the entire planning horizon. Denote the time varying cost of allocation as $r_l(t)$ and the available total budget for server allocation over the entire time horizon as C. Then, constraint 4.8 may be modified appropriately as shown below in constraint 4.13

$$\sum_{l \in L} \sum_{t \in T} r_l(t) c_l(t) \leq C \tag{4.13}$$

4.2.3 Simulated Annealing Algorithm

The server allocation formulation is nonlinear with products of continuous $X_{rl}(t)$ and n_{rl} variables with integer $c_l(t)$ variables. As such, a simulated annealing based heuristic is presented for solving the mixed integer non-linear optimization model presented in section 4.2.2. The simulated annealing algorithm is usually appropriate for discrete optimization problems having a nonlinear objective function with multiple local minima. The simulated annealing algorithm is employed for dynamic server allocation by embedding the multi-server ONBTM as a function evaluator to analyze the quality of a candidate solution. Simulated annealing is a meta-heuristic algorithm inspired by the behavior of materials while being subject to a gradual lowering of temperature. The simulated annealing algorithm has been shown to converge to the set of optimal solutions by modeling the cooling process as a sequence of Markov Chains or as a single in-homogeneous Markov Chain. Henderson *et al.* (2003) and Yang (2014) provide an in-depth discussion of the various aspects of the Simulated Annealing algorithm. The various elements of the simulated annealing algorithm as applied to the server allocation problem are discussed in the rest of this section.

4.2.3.1 Algorithm Description

```
Algorithm: Simulated Annealing Algorithm for Server Allocation
while stopping criterion not attained do
    Generate an Initial Solution
    while repetition count not attained do
        Generate a neighborhood around current solution using one-swaps
        Randomly choose a candidate solution from the generated
          neighborhood
        Evaluate the solution and compute $\Delta$
        if $\Delta < 0$ then
            Accept the candidate solution
        end
        else
            Accept the solution with probability $exp\left\{\frac{-\Delta}{T}\right\}$
        end
    end
    $k = k + 1$
    $T = T_0(\alpha^k)$
end
```

4.2.3.2 Initial Solution Generation

It is assumed that all workstations have at least one server at all times since arrivals could (and do) occur at each workstation in each period. Then, a naive initial solution is generated by allocating all excess $(S - L)$ servers to a single workstation. Intuitively, one would expect the initial solution to have little effect on the allocation proposed by the simulated annealing algorithm if the algorithm is run for a substantial number of iterations. This is particularly true since the neighborhood definition for modifying candidate solutions between iterations selects from the set of workstations at any period with excess servers. Thus, one excess server will necessarily be redistributed from that workstation on the first iteration, and, will have high likelihood of having additional servers moved in subsequent iterations if the workstation is not the bottleneck queue.

4.2.3.3 Cooling Schedule

A cooling schedule is completely specified by a choice of initial temperature and a cooling schedule. The initial temperature is heuristically chosen such that the acceptance probability at the start of the simulated annealing process is p_0. The initial temperature is calculated by analyzing the temperature at which the acceptance probability is p_0 for the largest possible deviation in the objective function, Δ_{max}. The expression for acceptance probability is as given below.

$$\begin{aligned} p_0 &= exp\left\{\frac{-\Delta_{max}}{T_0}\right\} \\ ln(p_0) &= \frac{-\Delta_{max}}{T_0} \\ T_0 &= \frac{-\Delta_{max}}{ln(p_0)} \end{aligned}$$

The largest possible deviation in the objective function, Δ_{max} is computed by computing the largest queue over all of time for the initial solution generated. The reasoning behind such a computation is that the best possible solution attainable in theory is empty queues for all stations over all of time while the worst possible solution is the one provided by the initial solution for the simulated annealing algorithm.

A geometric cooling schedule of the form shown below is chosen for the simulated annealing process.

$$T = T_0(\alpha^k) \tag{4.14}$$

The value of α is chosen such that the cooling process is slow enough to allow the system to stabilize easily. The stopping criterion for the annealing process is usually set to be when the temperature, T reaches the 10^{-5} to 10^{-10} range.

4.2.3.4 Neighborhood Generation

At each iteration of the simulated annealing algorithm a candidate solution is chosen randomly from a neighborhood of the current solution. The definition of neighborhood is subjective for the simulated annealing algorithm. For the dynamic server allocation problem a neighborhood is defined by a swap of one server for one time period between two workstations. First, the workstation with the largest queue over all of time is identified. Then one of the time periods up to and including that period is selected at random. For that period, one of the other workstations with more than one server is randomly selected and that server is moved to the workstation that was identified as having the largest queue. The strategy essentially is to attack the current bottleneck, measured by queue length, and provide it some extra resource at some period that can potentially contribute to reducing the current queue. In the absence of any such one-server swaps to the workstation with the largest queue over all of time a completely random swap is performed, where, a time period is chosen at random and one server is moved from a workstation with more than one server to a randomly chosen workstation in the chosen time period. The pseudocode for the neighborhood generation procedure is detailed below.

```
Algorithm: Neighborhood Generation
Identify max-queue and associated time window, $t_{max-queue}$ for current solution.
Make a feasible one-swap to max-queue for a randomly chosen time period in $0 \ldots, t_{max-queue}$
```

4.2.3.5 Solution Quality Evaluation

The multi-server approximation presented in Section 4.2.1 is embedded into the simulated annealing process as a function evaluator to assess the quality of a candidate solution. The computationally efficient multi-server ONBTM algorithm is used to assess the maximum queue length experienced over all of time in the system given the non-homogeneous Poisson arrival process for all the product types.

4.3 Results and Discussion

The performance of the Simulated Annealing algorithm for dynamic server allocation is compared against a greedy static allocation policy for a sixteen workstation, four product jobshop. The greedy static allocation policy assigns servers in proportion to the aggregated effective arrival rate for each workstation based on the set of products, their arrival rates and processing times. The SA algorithm with a geometric cooling schedule is applied to the jobshop instance for six different levels of servers to be allocated. A factorial experiment is performed with three factors - namely, non-homogeneous arrival pattern, choice of initial solution, and, choice of neighborhood each with two levels. The arrival pattern factor captures the effect of the time-varying pattern for products on the performance of the Simulated Annealing algorithm. The two arrival patterns investigated in the designed experiment are sinusoidal and triangular arrival rate functions. The choice of initial solution and the choice of neighborhood are subjective aspects of the SA algorithm which are included in the designed experiment. The two levels of choice of initial solution are nominated as *Naive-First* and *Naive-Last* indicating the index of the station to which the excess servers are allocated initially. The two levels of the choice of neighborhood are similar in the sense that a feasible one server swaps are performed to generate neighborhoods of the current solution in both cases. However, the two levels differ in the how these swaps are performed. In the random swap level the neighborhood is defined to be all feasible one server swaps with respect to the current solution. In the Max-Queue swap the neighborhood is defined to be the universe of one-servers swaps where servers are added to the workstation with the largest queue length over all of time in epochs preceding the occurrence of the largest queue. The Max-Queue swap has been described previously in the neighborhood generation discussion in Section 4.2.3. The following parameter assumptions for the SA algorithm experimental runs as shown in Table 4.1.

Parameter	Parameter Value/Range
Product Types, $\|R\|$	4
Workstations, $\|L\|$	16
Non-Homogeneous Arrival Pattern	Sinusoidal/Triangular
Initial acceptance probability, p_0	0.99
Initial Temperature, T_0	$\frac{-\Delta_{max}}{ln(p_0)}$
Cooling Schedule	Geometric, $T_k = T_0\alpha^k$
α	0.99
Maximum Iterations, K	2000 (or) 20000
Servers to allocated, S	17 ($\|L\| + 1$) to 22 ($\|L\| + 5$)
Initial Solution	Naive-First (or) Naive-Last
Choice of neighborhood	Random (or) Max-Queue

Table 4.1. Parameters for SA Designed Experiment

The performance of the SA algorithm for each experimental run is assessed by the percentage improvement (or degradation) over a greedy static allocation scheme, where the server configuration is kept fixed for all time.

```
Algorithm: Greedy Static Allocation Scheme
Compute time-average arrival rate for each workstation by product
    $\lambda_l^r = \frac{\int\limits_0^T \lambda_{0l}^r(t)}{T}$    (4.15)
Solve traffic equations for queueing network with aggregated arrival rates to compute effective arrival rates
    $\lambda_l^{\prime r} = \lambda_l^r + \sum_{k \neq l} p_{kl}^r \lambda_k^{\prime r}$    (4.16)
Compute aggregated arrival rates
    $\lambda_l = \sum_{r \in R(l)} \lambda_l^{\prime r}$    (4.17)
Create an ordering of workstations $o : L \to L$
    $\lambda_{o(1)} \geq \lambda_{o(2)} \geq \ldots \geq \lambda_{o(|L|)}$    (4.18)
Allocate servers in descending order of aggregated arrival rates
    $c_l(t) = c_l = \min\Big\{ \Big\lceil \Big( \frac{\lambda_l}{\sum\limits_{l \in L} \lambda_l} \Big) S \Big\rceil,\ S - \sum_{\{p:\ o(p)<o(l)\}} c_p \Big\}$    (4.19)
```

The average and standard deviation of the percentage improvement (or degradation) of the Simulated Annealing experimental runs aggregated across all available server levels is presented in Table 4.2. The large variance around the average improvement is explained by significant difference in the performance of the SA algorithm for different available server levels. The designed experiment is analyzed as a 2^3 factorial experiment with the available servers treated as a blocking factor. Two replications were performed for each design point of the factorial experiment for each of the six levels of the blocking factor. Thus, a total of 96 experiment runs were performed with

Arrival Pattern	Initial Allocation	Neighborhood	Avg. Improvement	Std. Dev.
Sinusoidal	Naive - First	Random	-62.38	33.39
Sinusoidal	Naive - First	Max - Queue	7.47	21.45
Sinusoidal	Naive - Last	Random	-69.64	40.68
Sinusoidal	Naive - Last	Max - Queue	9.69	21.78
Triangular	Naive - First	Random	-64.23	55.42
Triangular	Naive - First	Max - Queue	46.63	12.21
Triangular	Naive - Last	Random	-70.80	64.72
Triangular	Naive - Last	Max - Queue	48.45	11.64

Table 4.2. Average Improvement/Degradation of SA over Greedy Allocation

runs being randomized within each block. The results of the statistical analysis of the factorial experiment and the impact of iterations on convergence are discussed in the rest of this section.

4.3.1 Statistical Analysis of Factorial Experiment

The p-values of the effects test for the SA factorial experiment is presented in Table 4.3. The detailed Analysis of Variance and effect tests are reported in Appendix B.1. The hypothesis for the effect test is that the effect of all levels for a factor under consideration are equal to one another, while, the alternative hypothesis postulates that the effect of at least one pair of factor levels are not equal. The large p-value for the initial solution factor indicates that the various levels of factor has no impact on the performance of the SA algorithm. This makes intuitive sense as one would expect the initial solution to have no bearing on a randomized iterative search process such as simulated annealing if the process is run for adequate number of iterations. The effects test also suggests that there is statistical evidence to show that the neighborhood definition and the number of servers available to be allocated have a significant effect on how the SA algorithm fares as compared to the greedy static allocation heuristic.

The Arrival Pattern factor has a significant effect at the $\alpha = 0.05$ significance level but the main effect is not significant at the $\alpha = 0.01$ significance level. The only two-way interaction effect that is significant is the interaction between arrival pattern and neighborhood. All interaction effects involving the initial solution factor have no statistically significant impact.

Factor	Effect Type	p-value
Arrival Pattern	Main	0.0139
Initial Solution	Main	0.6024
Neighborhood	Main	< 0.0001
Servers Available	Blocking	0.0009
(Arrival Pattern)*(Initial Solution)	Interaction	0.8712
(Arrival Pattern)*(Neighborhood)	Interaction	0.0026
(Initial Solution)*(Neighborhood)	Interaction	0.4007
(Arrival Pattern)*(Initial Solution)*(Neighborhood)	Interaction	0.8692

Table 4.3. Effect Test for SA Designed Experiment

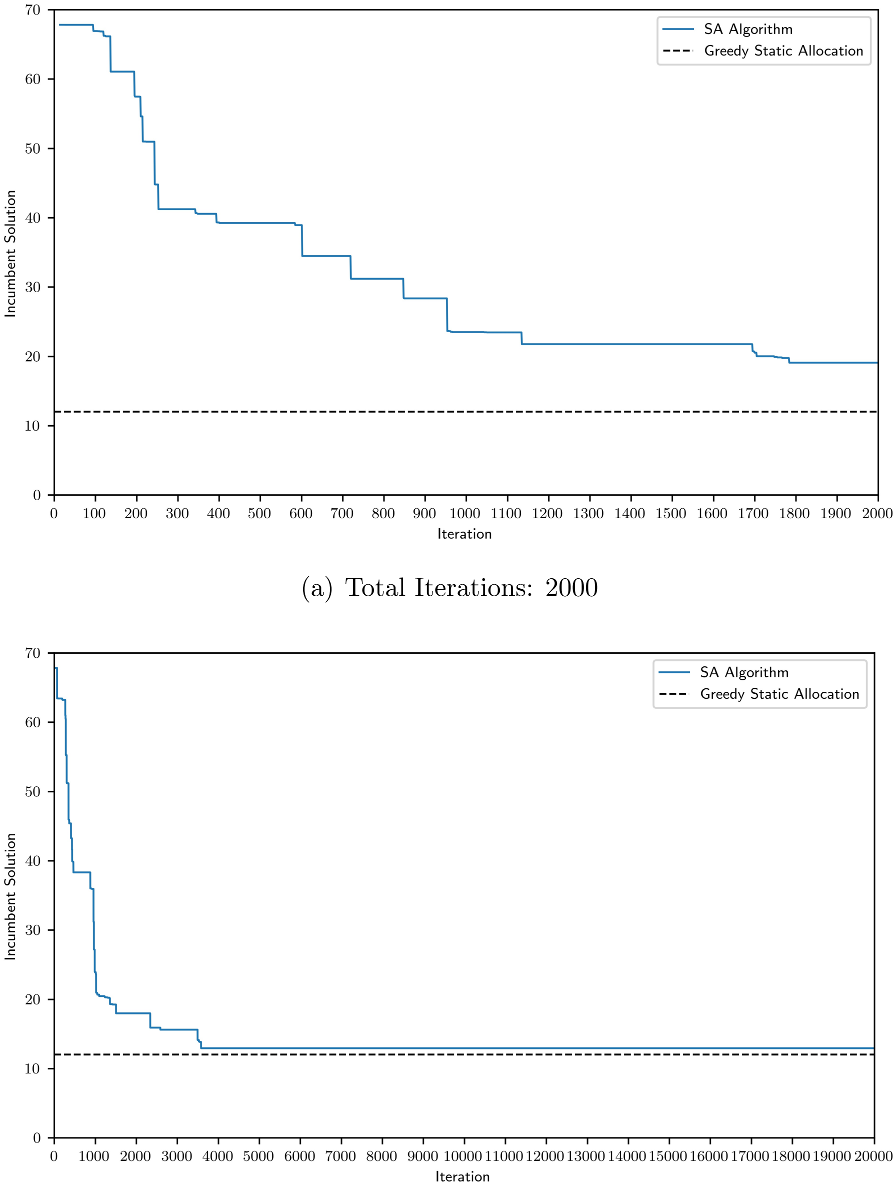
4.3.2 Impact of Run Length and Trajectory of Solution Improvement

The SA algorithm is an iterative process where inferior solutions may be accepted probabilistically to avoid being trapped in local optima. The number of annealing iterations performed could have a significant impact on solution quality. The impact of annealing run length is investigated and the trajectory of improvement achieved is studied in the context of the factorial experiment. The performance of the SA algorithm applied to the jobshop instance for 2000 iterations is compared against its performance for 20000 iterations. The run with 2000 iterations results in a final temperature of approximately 10^{-5}. The performance of the SA algorithm is examined for the Sinusoidal arrival pattern for 19 servers to be allocated with a focus on studying the impact of run length and neighborhood definition. One important aspect of any iterative process is speed at which it arrives a quality solution. The last iteration where the incumbent solution changed is tracked for the two different neighborhood definitions for the jobshop instance. Table 4.4 reports the last incumbent update iteration for the Random neighborhood and the Max-Queue neighborhood for runs of length 2000 and 20000 respectively. The results in Table 4.4 demonstrate that while the Random neighborhood experiences fewer incumbent updates relatively slowly, the Max-Queue neighborhood experiences a large number of incumbent updates in a small number of iterations. Thus, the strategy of augmenting the workstation with the largest queue with one additional server by going back into time seems to offer superior results over randomly swapping a server between two workstations.

Neighborhood	Total Iterations	Last Incumbent Update Iteration	Total Incumbent Updates
Random	2000	1784	66
Random	20000	3581	89
Max-Queue	2000	999	93
Max-Queue	20000	764	94

Table 4.4. Last Incumbent Improvement Iteration

Figure 4.1 plots the incumbent solution against the iteration number for the random neighborhood. The plots for the random neighborhood indicate that the random neighborhood is slow to converge to a quality solution which outperforms the greedy static allocation. This observation is corroborated by the last incumbent update iteration values of 1784 and 3581 for 2000 and 20000 total iterations respectively. It is also observed that the random neighborhood fails to perform better than the greedy static allocation even when the SA algorithm is run for a significantly long time (20000 iterations).

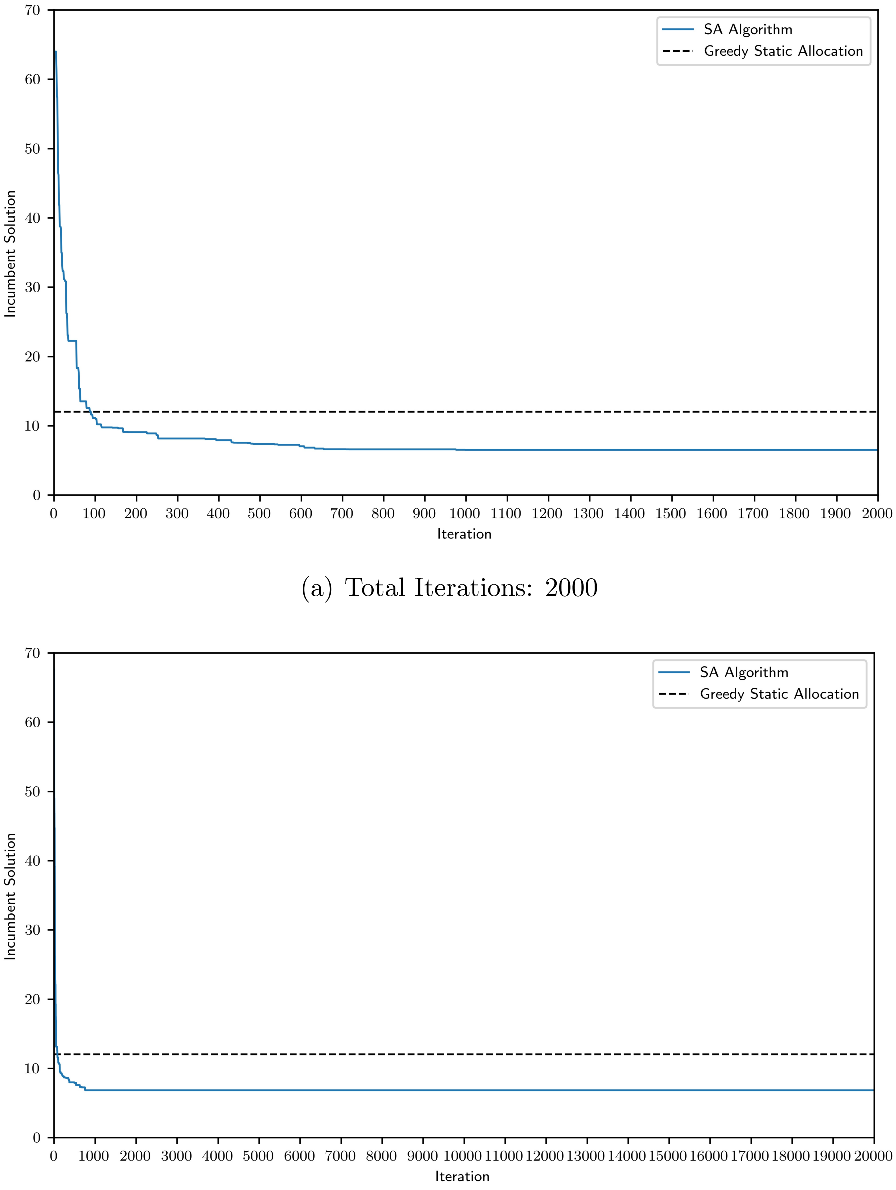


(a) Total Iterations: 2000

(b) Total Iterations: 20000

Figure 4.1. Improvement Trajectory: Random Neighborhood

Figure 4.2 plots the incumbent solution against the iteration number for the Max-Queue neighborhood. In contrast to the Random neighborhood, the Max-Queue neighborhood quickly outperforms the greedy static allocation and the trajectory of solution improvement is steep. This provides empirical evidence that the Max-Queue neighborhood yields a quality solution with relatively low computational effort.



(a) Total Iterations: 2000

(b) Total Iterations: 20000

Figure 4.2. Improvement Trajectory: Max-Queue Neighborhood

The Max-Queue neighborhood outperforms the Random neighborhood on solution quality as well as time to obtain a quality solution. The Max-Queue neighborhood outperforms the Greedy Static Allocation in most of the cases in the factorial experiment as observed in Table 4.2.

Chapter 5

CONCLUSIONS AND FUTURE RESEARCH

In this dissertation, multi-product manufacturing systems with dynamic product demands are studied. The research presented in this dissertation addresses the gap in literature for the performance analysis of multi-product manufacturing systems with time-varying product demands. The key mathematical modeling framework employed in this dissertation is multi-class queueing networks with time-varying arrivals. The exact queueing analysis of time-varying queues is intractable for very simple cases and thus heuristic approximations are explored as part of this dissertation. Specifically, heuristic approximations are presented for the performance analysis of such systems under assumptions of unlimited and limited queueing space at workstations. Approximations for incorporating priorities to reflect due-date priorities are presented as well. The final part of this dissertation demonstrates the application of presented approximations to resource and capacity allocation decisions in modern manufacturing systems. In particular, a simulated annealing based heuristic is presented for the allocation of servers in a dynamic multi-product manufacturing system with time-varying product demands.

Chapter 2 presented incremental approximations for the analysis of dynamic multi-product manufacturing systems with no waiting room restrictions. Two different approximation approached were presented - one based on equivalences between closed and open queueing networks and the other based on an open network decomposition. The exactness of the approximation approaches were mathematically established for stationary queueing systems. The estimates of WIP levels and throughput rates were combined to derive lead time forecasts by product type. The performance of these approximations were numerically tested on large flowshop and jobshop instances.

Chapter 3 explored the aspects of product priorities and finite waiting room in dynamic multi-product manufacturing systems. Approximations based on the $M/M/1$ priority queue with a homogeneous priority scheme were presented for modeling priorities. The $M/M/1/K$ loss queue was leveraged to arrive at performance estimates in the finite buffer case. The efficacy of the proposed approximations was statistically analyzed through a designed experiment considering relevant factors of interest.

Chapter 4 considered the issue of employing performance estimates to drive resource allocation decisions in dynamic multi-product manufacturing systems. A multi-server extension to the ONBTM algorithm was presented. The multi-server approximation is reformulated as a mixed-integer non-linear optimization problem for allocating servers in a time-varying fashion. A simulated annealing based heuristic was presented to solve the server allocation optimization problem. The performance of the simulated annealing heuristic was compared against a greedy static allocation scheme through a designed experiment.

The work in this dissertation opens up avenues for a wide of range of problems with regards to dynamic manufacturing systems and time-varying queueing networks in general. The research presented in this dissertation can be extended and enriched in several different directions. A few possible directions for future research are discussed in the rest of this chapter.

Simultaneous Buffer and Workload Allocation in Dynamic Multi-Product Manufacturing Systems

The simultaneous optimization of buffer space (waiting room) and workloads (service rates) have been gaining a lot of attention in recent literature (for instance see Smith (2018), Zhang *et al.* (2017b)). However, this problem is yet to be addressed in the context of manufacturing systems with dynamic time-varying demands. The identification of time-varying bottlenecks and time-varying buffer occupancy levels and appropriately allocating workloads and service resources in a time-varying fashion are some aspects that could be examined. Addressing the simultaneous buffer and service rate optimization problem for multi-product manufacturing systems with time-varying product demands would be immensely useful to modern manufacturing organizations. This is particularly true if the model could be transformed into one of a multi-class closed network formulation such that buffer limits were set by product type over the time horizon reflecting their dynamic demand. This could possibly lead to a release control policy similar to a product-dependent dynamic CONWIP philosophy.

Staffing and Real-Time Rostering in Dynamic Queueing Systems

Another possible future research area is the staffing and real-time rostering of queueing systems with time-varying arrivals. A very relevant application of this future research direction would be for staffing and real-time rostering of security screening checkpoints. Consider an airport with D terminals each with N_d security checkpoints. Daily passenger volume forecasts are generated for each passenger class (Standard or PreCheck) and appropriate non-homogeneous Poisson arrival rate functions are fitted to these forecasts. The staffing problem would involve assigning transportation security officer's (TSO's) to pre-determined shifts with an objective of minimizing the maximum wait-time that passengers experience over the entire day at any checkpoint. The real-time rostering problem involves determining when to move TSO's between checkpoints to reponsively deal with unexpected surges in passenger volumes at a given checkpoint.

Dynamic Manufacturing Systems with General Service and General Non-Homogeneous Demands

This dissertation focuses on the performance evaluation of multi-product manufacturing systems with non-homogeneous Poisson demands and exponential service. However, the assumptions of exponential service and nonstationary Poisson arrivals might not be appropriate for some systems. The analysis of systems with general service and arrival patterns is another important future research direction. One possible solution for the performance evaluation of such systems is the application of two-moment approximations for $G/G/1$ or $G/G/c$ queues in an incremental framework similar to the approach employed in this dissertation.

Optimally Solving the MINLP for Server Allocation in Dynamic Manufacturing Systems

A simulated annealing heuristic is presented in Chapter 4 for the MINLP formulation presented in Section 4.2.2. However, the simulated annealing heuristic does not provide any quantifiable guarantees on solution quality for the server allocation problem. Thus, a natural future research direction is exploring ways to solve the MINLP optimally. One approach for solving the MINLP optimally would be a divide-and-conquer strategy such as Dynamic Programming. However, a potential obstacle for the successful application of such a strategy is that in a forward pass over time an optimal decision at time t is not necessarily optimal over the entire time horizon beyond time t. The successful application of a backward pass over time in a DP approach starting at the end of the time horizon, T is hindered by the need to condition on the starting WIP levels at the beginning of the time epoch, which, is a continuous variable in the formulation presented in Section 4.2.2.

REFERENCES

Alnowibet, K. A. and H. Perros, “Nonstationary analysis of the loss queue and of queueing networks of loss queues”, European Journal of Operational Research **196**, 3, 1015–1030 (2009).

Altiok, T. and H. G. Perros, “Approximate analysis of arbitrary configurations of open queueing networks with blocking”, Annals of Operations Research **9**, 1, 481–509 (1987).

Askin, R. and J. Goldberg, “Design and operation of lean production systems”, John Wiley& Sons: New York (2002).

Askin, R. G. and G. J. Hanumantha, “Queueing network models for analysis of nonstationary manufacturing systems”, International Journal of Production Research **56**, 1-2, 22–42 (2018).

Askin, R. G. and G. Jampani Hanumantha, “Analysis of performance approximations for queueing networks with non-homogeneous arrival processes”, in “11th Conference on Stochastic Models of Manufacturing and Service Operations (SMMSO 2017)”, pp. 123–130 (2017).

Baker, K. R., “Sequencing rules and due-date assignments in a job shop”, Management science **30**, 9, 1093–1104 (1984).

Banks, J., I. Carson, B. L. Nelson, D. M. Nicol *et al.*, *Discrete-event system simulation* (Pearson, 2005).

Boxma, O. J., A. H. G. R. Kan and M. van Vliet, “Machine allocation problems in manufacturing networks”, European Journal of Operational Research **45**, 1, 47–54 (1990).

Bryant, R. M., A. E. Krzesinski, M. S. Lakshmi and K. M. Chandy, “The mva priority approximation”, ACM Transactions on Computer Systems (TOCS) **2**, 4, 335–359 (1984).

Chang, C.-S., “Stability, queue length, and delay of deterministic and stochastic queueing networks”, IEEE Transactions on Automatic Control **39**, 5, 913–931 (1994).

Chen, H., “Fluid approximations and stability of multiclass queueing networks: work-conserving disciplines”, The Annals of Applied Probability pp. 637–665 (1995).

Chen, H. and H. Zhang, “Stability of multiclass queueing networks under priority service disciplines”, Operations Research **48**, 1, 26–37 (2000).

Cheng, T. and M. Gupta, "Survey of scheduling research involving due date determination decisions", European Journal of Operational Research **38**, 2, 156–166 (1989).

Clark, G. M., "Use of polya distributions in approximate solutions to nonstationary m/m/s queues", Communications of the ACM **24**, 4, 206–217 (1981).

Dai, J., "A fluid limit model criterion for instability of multiclass queueing networks", The Annals of Applied Probability pp. 751–757 (1996).

Dai, J. and W. Dai, "A heavy traffic limit theorem for a class of open queueing networks with finite buffers", Queueing Systems **32**, 1-3, 5–40 (1999).

Dai, J. G., "On positive harris recurrence of multiclass queueing networks: a unified approach via fluid limit models", The Annals of Applied Probability pp. 49–77 (1995).

Dai, J. G. and S. P. Meyn, "Stability and convergence of moments for multiclass queueing networks via fluid limit models", IEEE Transactions on Automatic Control **40**, 11, 1889–1904 (1995).

Dallery, Y. and Y. Frein, "An efficient method to determine the optimal configuration of a flexible manufacturing system", Annals of Operations Research **15**, 1, 207–225 (1988).

Dallery, Y. and Y. Frein, "On decomposition methods for tandem queueing networks with blocking", Operations research **41**, 2, 386–399 (1993).

Di Crescenzo, A. and A. G. Nobile, "Diffusion approximation to a queueing system with time-dependent arrival and service rates", Queueing systems **19**, 1, 41–62 (1995).

Duda, A., "Diffusion approximations for time-dependent queueing systems", IEEE Journal on Selected Areas in Communications **4**, 6, 905–918 (1986).

Duenyas, I., "Single facility due date setting with multiple customer classes", Management Science **41**, 4, 608–619 (1995).

Duenyas, I. and W. J. Hopp, "Quoting customer lead times", Management Science **41**, 1, 43–57 (1995).

Eager, D. L. and J. N. Lipscomb, "The amva priority approximation", Performance Evaluation **8**, 3, 173–193 (1988).

Edelman, A. and H. Murakami, "Polynomial roots from companion matrix eigenvalues", Mathematics of Computation **64**, 210, 763–776 (1995).

Frenk, H., M. Labbé, M. Van Vliet and S. Zhang, “Improved algorithms for machine allocation in manufacturing systems”, Operations Research **42**, 3, 523–530 (1994).

Gershwin, S. B. and I. C. Schick, “Modeling and analysis of three-stage transfer lines with unreliable machines and finite buffers”, Operations Research **31**, 2, 354–380 (1983).

Grassmann, W. K., “Transient solutions in markovian queueing systems”, Computers & Operations Research **4**, 1, 47–53 (1977).

Green, L. and P. Kolesar, “The pointwise stationary approximation for queues with nonstationary arrivals”, Management Science **37**, 1, 84–97 (1991).

Green, L. V., P. J. Kolesar and J. Soares, “Improving the sipp approach for staffing service systems that have cyclic demands”, Operations Research **49**, 4, 549–564 (2001).

Green, L. V., P. J. Kolesar and J. Soares, “An improved heuristic for staffing telephone call centers with limited operating hours”, Production and Operations Management **12**, 1, 46–61 (2003).

Gross, D., *Fundamentals of queueing theory* (John Wiley & Sons, 2008).

Harrison, J. M. and L. M. Wein, “Scheduling networks of queues: Heavy traffic analysis of a two-station closed network”, Operations research **38**, 6, 1052–1064 (1990).

Hasan, C. and M. Spearman, “Optimal material release times in stochastic production environments”, International Journal of Production Research **37**, 6, 1201–1216 (1999).

Henderson, D., S. H. Jacobson and A. W. Johnson, “The theory and practice of simulated annealing”, in “Handbook of metaheuristics”, pp. 287–319 (Springer, 2003).

Hillier, F. S. and K. C. So, “On the simultaneous optimization of server and work allocations in production line systems with variable processing times”, Operations Research **44**, 3, 435–443 (1996).

Ioannou, G. and S. Dimitriou, “Lead time estimation in mrp/erp for make-to-order manufacturing systems”, International Journal of Production Economics **139**, 2, 551–563 (2012).

Izady, N. and D. Worthington, “Approximate analysis of non-stationary loss queues and networks of loss queues with general service time distributions”, European Journal of Operational Research **213**, 3, 498–508 (2011).

Izady, N. and D. Worthington, “Setting staffing requirements for time dependent queueing networks: The case of accident and emergency departments”, European Journal of Operational Research **219**, 3, 531–540 (2012).

Jain, S. and J. M. Smith, “Open finite queueing networks with m/m/c/k parallel servers”, Computers & operations research **21**, 3, 297–317 (1994).

Kaufman, J., “Approximation methods for networks of queues with priorities”, Performance Evaluation **4**, 3, 183–198 (1984).

Kerbache, L. and J. M. Smith, “Asymptotic behavior of the expansion method for open finite queueing networks”, Computers & Operations Research **15**, 2, 157–169 (1988).

Kittipiyakul, S. and T. Javidi, “Delay-optimal server allocation in multiqueue multiserver systems with time-varying connectivities”, IEEE Transactions on Information Theory **55**, 5, 2319–2333 (2009).

Kumar, P. and S. P. Meyn, “Stability of queueing networks and scheduling policies”, IEEE Transactions on Automatic Control **40**, 2, 251–260 (1995).

Kumar, S. and P. Kumar, “Performance bounds for queueing networks and scheduling policies”, IEEE Transactions on Automatic Control **39**, 8, 1600–1611 (1994).

Lee, D.-S. and B. Sengupta, “Queueing analysis of a threshold based priority scheme for atm networks”, IEEE/ACM Transactions on Networking (TON) **1**, 6, 709–717 (1993).

Lee, G.-C., “Estimating order lead times in hybrid flowshops with different scheduling rules”, Computers & Industrial Engineering **56**, 4, 1668–1674 (2009).

Liu, Y. and W. Whitt, “Large-time asymptotics for the g t/m t/s t+ gi t many-server fluid queue with abandonment”, Queueing systems **67**, 2, 145–182 (2011).

Liu, Y. and W. Whitt, “Algorithms for time-varying networks of many-server fluid queues”, INFORMS Journal on Computing **26**, 1, 59–73 (2013).

MacGregor Smith, J. and F. Cruz, “The buffer allocation problem for general finite buffer queueing networks”, Iie Transactions **37**, 4, 343–365 (2005).

Malone, K. M., *Dynamic queueing systems: behavior and approximations for individual queues and for networks*, Ph.D. thesis, Massachusetts Institute of Technology (1995).

Mandelbaum, A. and W. A. Massey, “Strong approximations for time-dependent queues”, Mathematics of Operations Research **20**, 1, 33–64 (1995).

Manitz, M., “Queueing-model based analysis of assembly lines with finite buffers and general service times”, Computers & Operations Research **35**, 8, 2520–2536 (2008).

Massey, W. A., “The analysis of queues with time-varying rates for telecommunication models”, Telecommunication Systems **21**, 2, 173–204 (2002).

Massey, W. A. and W. Whitt, “Networks of infinite-server queues with nonstationary poisson input”, Queueing Systems **13**, 1, 183–250 (1993).

Massey, W. A. and W. Whitt, “Uniform acceleration expansions for markov chains with time-varying rates”, Annals of Applied Probability pp. 1130–1155 (1998).

Morris, R., “Priority queuing networks”, The Bell System Technical Journal **60**, 8, 1745–1769 (1981).

Nelson, B. L. and M. R. Taaffe, “The [pht/pht/] k queueing system: Part ii—the multiclass network”, INFORMS Journal on Computing **16**, 3, 275–283 (2004).

Osorio, C. and M. Bierlaire, “An analytic finite capacity queueing network model capturing the propagation of congestion and blocking”, European Journal of Operational Research **196**, 3, 996–1007 (2009).

Öztürk, A., S. Kayalıgil and N. E. Özdemirel, “Manufacturing lead time estimation using data mining”, European Journal of Operational Research **173**, 2, 683–700 (2006).

Palmer, J. and I. Mitrani, “Optimal and heuristic policies for dynamic server allocation”, Journal of Parallel and Distributed Computing **65**, 10, 1204–1211 (2005).

Pender, J., “A poisson–charlier approximation for nonstationary queues”, Operations Research Letters **42**, 4, 293–298 (2014).

Pender, J., “An analysis of nonstationary coupled queues”, Telecommunication Systems **61**, 4, 823–838 (2016a).

Pender, J., “Sampling the functional kolmogorov forward equations for nonstationary queueing networks”, INFORMS Journal on Computing **29**, 1, 1–17 (2016b).

Reiser, M. and S. S. Lavenberg, “Mean-value analysis of closed multichain queuing networks”, Journal of the ACM (JACM) **27**, 2, 313–322 (1980).

Rothkopf, M. H. and S. S. Oren, “A closure approximation for the nonstationary m/m/s queue”, Management Science **25**, 6, 522–534 (1979).

Schwarz, J. A., G. Selinka and R. Stolletz, “Performance analysis of time-dependent queueing systems: survey and classification”, Omega **63**, 170–189 (2016).

Sevcik, K. C., "Priority scheduling disciplines in queuing network models of computer systems", in "IFIP Congress", (1977).

Shalev-Oren, S., A. Seidmann and P. J. Schweitzer, "Analysis of flexible manufacturing systems with priority scheduling: Pmva", Annals of Operations Research **3**, 3, 113–139 (1985).

Shanthikumar, J. G. and D. D. Yao, "Optimal server allocation in a system of multi-server stations", Management Science **33**, 9, 1173–1180 (1987).

Shanthikumar, J. G. and D. D. Yao, "On server allocation in multiple center manufacturing systems", Operations Research **36**, 2, 333–342 (1988).

Shin, Y. W. and D. H. Moon, "Approximation of discrete time tandem queueing networks with unreliable servers and blocking", Performance Evaluation **120**, 49–74 (2018).

Smith, J. M., "System capacity and performance modelling of finite buffer queueing networks", International Journal of Production Research **52**, 11, 3125–3163 (2014).

Smith, J. M., "Simultaneous buffer and service rate allocation in open finite queueing networks", IISE Transactions **50**, 3, 203–216 (2018).

Smith, J. M. and R. Barnes, "Optimal server allocation in closed finite queueing networks", Flexible Services and Manufacturing Journal **27**, 1, 58–85 (2015).

Smith, J. M., F. R. B. Cruz and T. van Woensel, "Optimal server allocation in general, finite, multi-server queueing networks", Applied Stochastic Models in Business and Industry **26**, 6, 705–736 (2010).

Spearman, M. L. and R. Q. Zhang, "Optimal lead time policies", Management Science **45**, 2, 290–295 (1999).

Stolletz, R., "Approximation of the non-stationary m (t)/m (t)/c (t)-queue using stationary queueing models: The stationary backlog-carryover approach", European Journal of operational research **190**, 2, 478–493 (2008).

Stolyar, A. L., "On the stability of multiclass queueing networks: a relaxed sufficient condition via limiting fluid processes", Markov Processes and Related Fields **1**, 4, 491–512 (1995).

Suri, R. and G. W. Diehl, "A new'building block'for performance evaluation of queueing networks with finite buffers", in "ACM SIGMETRICS Performance Evaluation Review", vol. 12, pp. 134–142 (ACM, 1984).

Taaffe, M. R. and K. L. Ong, "Approximating nonstationaryph (t)/m (t)/s/c queueing systems", Annals of Operations Research **8**, 1, 103–116 (1987).

Takahashi, Y., H. Miyahara and T. Hasegawa, “An approximation method for open restricted queueing networks”, Operations research **28**, 3-part-i, 594–602 (1980).

Tan, B. and S. Lagershausen, “On the output dynamics of production systems subject to blocking”, IISE Transactions **49**, 3, 268–284 (2017).

Tassiulas, L. and A. Ephremides, “Dynamic server allocation to parallel queues with randomly varying connectivity”, IEEE Transactions on Information Theory **39**, 2, 466–478 (1993).

Thompson, G. M., “Accounting for the multi-period impact of service when determining employee requirements for labor scheduling”, Journal of Operations Management **11**, 3, 269–287 (1993).

Tipper, D. and M. K. Sundareshan, “Numerical methods for modeling computer networks under nonstationary conditions”, IEEE Journal on Selected Areas in Communications **8**, 9, 1682–1695 (1990).

Vandaele, N., L. De Boeck and D. Callewier, “An open queueing network for lead time analysis”, IIE transactions **34**, 1, 1–9 (2002).

Vig, M. M. and K. J. Dooley, “Mixing static and dynamic flowtime estimates for due-date assignment”, Journal of Operations Management **11**, 1, 67–79 (1993).

Wang, W.-P., D. Tipper and S. Banerjee, “A simple approximation for modeling nonstationary queues”, in “INFOCOM’96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE”, vol. 1, pp. 255–262 (IEEE, 1996).

Webster, S., “Dynamic pricing and lead-time policies for make-to-order systems”, Decision Sciences **33**, 4, 579–600 (2002).

Wein, L. M., “Scheduling networks of queues: heavy traffic analysis of a two-station network with controllable inputs”, Operations Research **38**, 6, 1065–1078 (1990).

Weng, Z. K., “Manufacturing lead times, system utilization rates and lead-time-related demand”, European Journal of Operational Research **89**, 2, 259–268 (1996).

Weng, Z. K., “Strategies for integrating lead time and customer-order decisions”, IIE transactions **31**, 2, 161–171 (1999).

Whitt, W., “The pointwise stationary approximation for mt/mt/s queues is asymptotically correct as the rates increase”, Management Science **37**, 3, 307–314 (1991).

Whitt, W., “Decomposition approximations for time-dependent markovian queueing networks”, Operations Research Letters **24**, 3, 97–103 (1999).

Whitt, W., “Heavy-traffic limits for queues with periodic arrival processes”, Operations Research Letters **42**, 6, 458–461 (2014).

Yang, X.-S., *Nature-inspired optimization algorithms* (Elsevier, 2014).

Zhang, H.-Y., Q.-X. Chen, J. M. Smith, N. Mao, A.-L. Yu and Z.-T. Li, “Performance analysis of open general queuing networks with blocking and feedback”, International Journal of Production Research **55**, 19, 5760–5781 (2017a).

Zhang, M., A. Matta, A. Alfieri and G. Pedrielli, “A simulation-based benders’ cuts generation for the joint workstation, workload and buffer allocation problem”, in “2017 13th IEEE Conference on Automation Science and Engineering (CASE)”, pp. 1067–1072 (IEEE, 2017b).

APPENDIX A

APPENDIX FOR CHAPTER 3

A.1 Derivation of Step 6 of Priority Approximation for Multi-class Networks

For a multi-class $M/M/1$ queue with homogeneous priority discipline, average wait time in queue is given by

$$W_r^q = \frac{\frac{\rho}{\mu}}{(1-\sigma_{r-1})(1-\sigma_r)}, \; (\rho = \sum_{p=1}^{R} \rho_p = \sigma_R)$$

By Little's Law, average length of queue is

$$L_r^q = \lambda W_r^q = \frac{\rho \rho_r}{(1-\sigma_{r-1})(1-\sigma_r)}$$

Number in System is given by

$$L_r = L_r^q + \rho_r = \frac{\rho \rho_r}{(1-\sigma_{r-1})(1-\sigma_r)} + \rho_r$$

Consider queue lengths of jobs of priority class r or higher

$$\begin{aligned}
\sum_{p=1}^{r} L_p &= \frac{\rho\rho_1}{(1-\sigma_1)} + \rho_1 + \frac{\rho\rho_2}{(1-\sigma_1)(1-\sigma_2)} + \rho_2 + ... + \frac{\rho\rho_r}{(1-\sigma_{r-1})(1-\sigma_r)} + \rho_r \\
&= \sigma_r + \left[\frac{\rho\rho_1}{(1-\sigma_1)} + \frac{\rho\rho_2}{(1-\sigma_1)(1-\sigma_2)} + .. + \frac{\rho\rho_r}{(1-\sigma_{r-1})(1-\sigma_r)}\right] \\
&= \sigma_r + \rho\left[\frac{1}{(1-\sigma_1)}\left(\frac{\rho_1(1-\sigma_2)+\rho_2}{(1-\sigma_2)}\right) + .. + \frac{\rho_r}{(1-\sigma_{r-1})(1-\sigma_r)}\right] \\
&= \sigma_r + \rho\left[\frac{1}{(1-\sigma_1)}\left(\frac{\rho_1(1-(\rho_1+\rho_2))+\rho_2}{(1-\sigma_2)}\right) + .. + \frac{\rho_r}{(1-\sigma_{r-1})(1-\sigma_r)}\right] \\
&= \sigma_r + \rho\left[\frac{1}{(1-\sigma_1)}\left(\frac{(\rho_1+\rho_2)(1-\rho_1)}{(1-\sigma_2)}\right) + .. + \frac{\rho_r}{(1-\sigma_{r-1})(1-\sigma_r)}\right] \\
&= \sigma_r + \rho\left[\left(\frac{\sigma_2}{1-\sigma_2}\right) + .. + \frac{\rho_r}{(1-\sigma_{r-1})(1-\sigma_r)}\right] \\
&(since\ \rho_1 = \sigma_1\ and\ \sigma_2 = \rho_1 + \rho_2)
\end{aligned}$$

Similarly combining successive terms results in

$$\sum_{p=1}^{r} L_p = \sigma_r + \frac{\rho\sigma_r}{1-\sigma_r}$$

Expressing the above relation as a quadratic equation in σ_r gives

$$\sigma_r^2 - \left(1 + \sum_{p=1}^{r} L_p + \rho\right)\sigma_r + \sum_{p=1}^{r} L_p = 0 \tag{A.1}$$

Matching $\sum_{p=1}^{r} L_p$ with $\sum_{p=1}^{r} n_{rl}(t)$ gives an estimate of $\sigma_{rl}(t)$ and $\rho_{rl}(t)$ for the algorithm.

A.1.0.0.1 **Proposition 3**

The solution to the quadratic equation in Step 6 of the proposed multi-class priority approximation has a lower bound and upper bound.

Proof Consider equation A.1

$$\sigma_r^2 - \left(1 + \sum_{p=1}^{r} L_p + \rho\right)\sigma_r + \sum_{p=1}^{r} L_p = 0$$

Dividing by $(\sum_{p=1}^{r} L_p + 1)$

$$\left(\frac{1}{\sum_{p=1}^{r} L_p + 1}\right)\sigma_r^2 - \left(1 + \frac{\rho}{\sum_{p=1}^{r} L_p + 1}\right)\sigma_r + \left(\frac{\sum_{p=1}^{r} L_p}{\sum_{p=1}^{r} L_p + 1}\right) = 0$$

Denoting $x = \sigma_r$ and $n = \sum_{p=1}^{r} L_p$ roots of the equation are given by

$$x = \frac{\left[\left(1 + \frac{\rho}{n+1}\right) \pm \sqrt{(1 + \frac{\rho}{n+1})^2 - (\frac{4n}{(n+1)^2})}\right]}{\frac{2}{n+1}}$$

The root of interest is

$$x = \frac{\left[\left(1 + \frac{\rho}{n+1}\right) - \sqrt{(1 + \frac{\rho}{n+1})^2 - (\frac{4n}{(n+1)^2})}\right]}{\frac{2}{n+1}}$$

as the other root is at least as large as $\frac{n+1+\rho}{2} = \frac{n+1}{2} + \frac{\rho}{2} \geq \rho$ (since $\rho \leq 1$ and $n \geq 0$).

Notice that

$$\sqrt{(1 + \frac{\rho}{n+1})^2 - (\frac{4n}{(n+1)^2})} \leq 1 + \frac{\rho}{n+1}$$

Thus 0 is a lower bound.

Also notice that since $\rho \geq \frac{n}{n+1}$

$$\sqrt{(1+\frac{\rho}{n+1})^2-(\frac{4n}{(n+1)^2})} \geq \sqrt{(1+\frac{n}{(n+1)^2})^2-(\frac{4n}{(n+1)^2})} = 1-\left(\frac{n}{(n+1)^2}\right)$$

Thus, $\frac{1}{2}(\rho+\frac{n}{n+1})$ is an upper bound.

A.2 Derivation of Equation 3.1 of the Finite Buffer Approximation for Multi-class Networks

The average queue length of a $M/M/1/K$ loss queue is given by

$$L_q = \frac{\rho}{1-\rho} - \frac{\rho(K\rho^K+1)}{1-\rho^{K+1}}$$

The average number in system for an $M/M/1/K$ queue can be expressed as

$$L = L_q + (1-p_0)$$

where

$$p_0 = \frac{1-\rho}{1-\rho^{K+1}}$$

Thus the relation between ρ and L can be expressed as

$$\begin{aligned} L &= \Big(\frac{\rho}{1-\rho} - \frac{\rho(K\rho^K+1)}{1-\rho^{K+1}}\Big) + \Big(1 - \frac{1-\rho}{1-\rho^{K+1}}\Big) \\ &= \Big(\frac{\rho}{1-\rho}+1\Big) - \Big(\frac{\rho(K\rho^K+1)}{1-\rho^{K+1}} + \frac{1-\rho}{1-\rho^{K+1}}\Big) \\ &= \Big(\frac{1}{1-\rho}\Big) - \Big(\frac{K\rho^{K+1}+1}{1-\rho^{K+1}}\Big) \end{aligned}$$

The above relation can be expressed as the polynomial equation (in ρ) as shown below.

$$(L-K)\rho^{K+2} + (1+K-L)\rho^{K+1} - (L+1)\rho + L = 0$$

The polynomial equation can be reduced to

$$(L\rho^{K+2} - L\rho^{K+1}) - K\rho^{K+1}(\rho-1) + (\rho^{K+1}-\rho) + L(1-\rho) = 0$$

$$
\begin{aligned}
(K-L)\rho^{K+1}(1-\rho) - (1-\rho)(\rho^{K} + \rho^{K-1} + \ldots + \rho) + L(1-\rho) &= 0 \\
(1-\rho)\Big((K-L)\rho^{K+1} - (\rho^{K} + \rho^{K-1} + \ldots + \rho) + L\Big) &= 0
\end{aligned}
$$

The above equation can further be reduced to

$$
\begin{aligned}
(1-\rho)\Big((\rho^{K+1} - \rho^{K}) + (\rho^{K+1} - \rho^{K-1}) + \ldots + (\rho^{K+1} - \rho)) + L(1-\rho^{K+1})\Big) &= 0 \\
(1-\rho)^2\Big((L-K)\rho^{K} + (L-K+1)\rho^{K-1} + \ldots + (L-1)\rho + L\Big) &= 0
\end{aligned}
$$

The companion matrix (Edelman and Murakami (1995)) for the polynomial equation of degree K is given by

$$
A = \begin{bmatrix}
\frac{L-K+1}{K-L} & \frac{L-K+2}{K-L} & \frac{L-K+3}{K-L} & \frac{L-K+4}{K-L} & \ldots & \frac{L-1}{K-L} & \frac{L}{K-L} \\
1 & 0 & 0 & 0 & \ldots & 0 & 0 \\
0 & 1 & 0 & 0 & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \ldots & 1 & 0
\end{bmatrix}_{(K)\times(K)}
$$

The polynomial equation is approximately solved (in MATLAB/Python) by identifying the eigenvalues of the companion matrix A as originally proposed in Edelman and Murakami (1995). The eigen value that relates to valid utilization for $K = 1$ (no waiting room) and $K = 2$ (one waiting spot) are shown below.

$$
\begin{aligned}
\rho(L,K) &= \frac{L}{1-L},\ K = 1 \\
\rho(L,K) &= \frac{L + \sqrt{-3L^2 + 6L + 1} - 1}{2(2-L)},\ K = 2
\end{aligned}
$$

Note that for both $K = 1$ and $K = 2$ the utilizations are monotonically increasing functions of L over $[0, K]$. For larger values of K the appropriate eigen value which corresponds to valid utilization take the form of complicated non-linear functions of L. However, empirically it is observed that the monotoneity property over $[0, K]$ still holds for larger values of system capacity as well.

A.3 Analysis of Variance and Effects Tests for Factorial Experiment

The results of the factorial experiment in section 3.5 of Chapter 3 are described in detail below. Four distinct single response factorial experiments were performed in JMP, one for each of the stations designated as *Entry*, *Start*, *Middle*, *End*. Tables A.1 through A.4 summarize the ANOVA and effect tests results for the four factorial experiments performed. The assumptions of normality, independence, and, constant variance of the residuals were checked and verified. Box-Cox transformation was performed on the response to resolve deviations from the normality and constant variance assumptions. All four factorial experiments provide statistical evidence of atleast one significant effect among all factors considered supported by the p-values obtained.

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	25	2358.4939	94.3398	2.4183
Error	28	1092.3138	39.0112	Prob >F
Total	53	3450.8078		0.0126*

Effect Tests

Source	Nparm	DF	Sum of Squares	F Ratio	Prob >F
config	2	2	103.78365	1.3302	0.2806
buffer	2	2	70.44976	0.9029	0.4169
util	1	1	451.15933	11.5649	0.0020*
arr	2	2	652.37362	8.3614	0.0014*
config*buffer	4	4	372.93399	2.3899	0.0747
config*util	2	2	142.49126	1.8263	0.1797
config*arr	4	4	156.99943	1.0061	0.4210
buffer*util	2	2	20.82601	0.2669	0.7677
buffer*arr	4	4	136.79149	0.8766	0.4904
util*arr	2	2	250.68539	3.2130	0.0554

Table A.1. ANOVA for *Entry* Workstation

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	25	1131.8671	45.2747	9.1092
Error	28	139.1653	4.9702	Prob >F
Total	53	1271.0325		<.0001*

Effect Tests

Source	Nparm	DF	Sum of Squares	F Ratio	Prob >F
config	2	2	148.71809	14.9610	<.0001*
buffer	2	2	22.12674	2.2259	0.1267
util	1	1	8.43128	1.6964	0.2034
arr	2	2	463.63112	46.6412	<.0001*
config*buffer	4	4	7.43745	0.3741	0.8251
config*util	2	2	234.89647	23.6305	<.0001*
config*arr	4	4	11.26236	0.5665	0.6890
buffer*util	2	2	0.73441	0.0739	0.9290
buffer*arr	4	4	50.57993	2.5442	0.0617
util*arr	2	2	184.04927	18.5153	<.0001*

Table A.2. ANOVA for *Start* Workstation

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	25	1375.2755	55.0110	20.0226
Error	28	76.9286	2.7474	Prob >F
Total	53	1452.2040		<.0001*

Effect Tests

Source	Nparm	DF	Sum of Squares	F Ratio	Prob >F
config	2	2	115.27638	20.9788	<.0001*
buffer	2	2	191.35337	34.8238	<.0001*
util	1	1	77.12316	28.0708	<.0001*
arr	2	2	491.94901	89.5283	<.0001*
config*buffer	4	4	22.12130	2.0129	0.1199
config*util	2	2	155.36688	28.2747	<.0001*
config*arr	4	4	6.93721	0.6312	0.6443
buffer*util	2	2	137.59565	25.0406	<.0001*
buffer*arr	4	4	15.10781	1.3747	0.2680
util*arr	2	2	162.44471	29.5628	<.0001*

Table A.3. ANOVA for *Middle* Workstation

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	25	1659.6593	66.3864	25.1329
Error	28	73.9596	2.6414	Prob >F
Total	53	1733.6190		<.0001*

Effect Tests

Source	Nparm	DF	Sum of Squares	F Ratio	Prob >F
config	2	2	126.92278	24.0255	<.0001*
buffer	2	2	273.69617	51.8086	<.0001*
util	1	1	51.53019	19.5085	0.0001*
arr	2	2	621.34914	117.6167	<.0001*
config*buffer	4	4	4.07052	0.3853	0.8173
config*util	2	2	158.27800	29.9608	<.0001*
config*arr	4	4	26.13954	2.4740	0.0673
buffer*util	2	2	244.79562	46.3380	<.0001*
buffer*arr	4	4	9.39585	0.8893	0.4832
util*arr	2	2	143.48153	27.1600	<.0001*

Table A.4. ANOVA for *End* Workstation

APPENDIX B

APPENDIX FOR CHAPTER 4

B.1 Analysis of Variance and Effect Tests for SA Algorithm Designed Experiment

The ANOVA and the Effects test of the designed experiment performed in Section 4.3 of Chapter 4 are presented in Table B.1.

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	12	268994.56	22416.2	19.2012
Error	83	96897.22	1167.4	Prob >F
Total	95	365891.79		<.0001*

Effect Tests

Source	Nparm	DF	Sum of Squares	F Ratio	Prob >F
arr	1	1	7372.77	6.3153	0.0139*
init alloc	1	1	319.30	0.2735	0.6024
neigh	1	1	222325.95	190.4394	<.0001*
avl servers	5	5	26822.51	4.5951	0.0009*
arr*init alloc	1	1	30.89	0.0265	0.8712
arr*neigh	1	1	11258.43	9.6437	0.0026*
init alloc*neigh	1	1	832.85	0.7134	0.4007
arr*init alloc*neigh	1	1	31.86	0.0273	0.8692
buffer*arr	4	4	9.39585	0.8893	0.4832
util*arr	2	2	143.48153	27.1600	<.0001*

Table B.1. ANOVA for SA Algorithm for Dynamic Server Allocation