Taylor & Francis
Taylor & Francis Group

International Journal of Production Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

Supervised-learning-based approximation method for multi-server queueing networks under different service disciplines with correlated interarrival and service times

Siamak Khayyati & Barış Tan

To cite this article: Siamak Khayyati & Barış Tan (2022) Supervised-learning-based approximation method for multi-server queueing networks under different service disciplines with correlated interarrival and service times, International Journal of Production Research, 60:17, 5176-5200, DOI: 10.1080/00207543.2021.1951448

To link to this article: https://doi.org/10.1080/00207543.2021.1951448

Published online: 19 Jul 2021.

INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH
2022, VOL. 60, NO. 17, 5176–5200
https://doi.org/10.1080/00207543.2021.1951448

Taylor & Francis
Taylor & Francis Group

Supervised-learning-based approximation method for multi-server queueing networks under different service disciplines with correlated interarrival and service times

Siamak Khayyati [a] and Barış Tan [b]

[a]College of Engineering, Koç University, Istanbul, Turkey ; [b]College of Administrative Sciences and Economics, College of Engineering, Koç University, Istanbul, Turkey

ABSTRACT

Developing efficient performance evaluation methods is important to design and control complex production systems effectively. We present an approximation method (SLQNA) to predict the performance measures of queueing networks composed of multi-server stations operating under different service disciplines with correlated interarrival and service times with merge, split, and batching blocks separated with infinite capacity buffers. SLQNA yields the mean, coefficient of variation, and first-lag autocorrelation of the inter-departure times and the distribution of the time spent in the block, referred as the cycle time at each block. The method generates the training data by simulating different blocks for different parameters and uses Gaussian Process Regression to predict the inter-departure time and the cycle time distribution characteristics of each block in isolation. The predictions obtained for one block are fed into the next block in the network. The cycle time distributions of the blocks are used to approximate the distribution of the total time spent in the network (total cycle time). This approach eliminates the need to generate new data and train new models for each given network. We present SLQNA as a versatile, accurate, and efficient method to evaluate the cycle time distribution and other performance measures in queueing networks.

ARTICLE HISTORY
Received 24 February 2021
Accepted 28 June 2021

KEYWORDS
Queueing networks; manufacturing systems; machine learning; simulation; stochastic models; sequence dependent systems

1. Introduction

A complex manufacturing system such as a semiconductor manufacturing plant includes hundreds of machines, tangled flow of materials and is affected by variability and dependence in part arrivals, order arrivals, and processing times. In addition, production and flow of materials at the machine levels and also at the system level are controlled in order to improve the number of parts produced per unit time, the inventory levels, and the time spent in a given part of the system referred as the *cycle time*. In this environment, developing efficient performance evaluation methods that take these characteristics into account is important to design and control production systems effectively (Fowler and Rose 2004).

Analytical methods developed to analyze the performance of queueing network models of manufacturing systems often impose many simplifying assumptions due to analytical tractability and achieving a high computational efficiency. Most of these methods assume independent and identically distributed (i.i.d) interarrival and service times, first-come first-serve discipline (FCFS) and consider a single server per block (Kuehn 1979; Dallery and Gershwin 1992; Buzacott and Shanthikumar 1993; Hopp and Spearman 2011). Empirical and analytical studies have indicated that these assumptions do not hold in many production systems, especially in complex systems such as semiconductor manufacturing plants (Schömig and Mittler 1995; Tan and Lagershausen 2017; Manafzadeh Dizbin 2020).

Although, many queueing network analysis algorithms consider queues that are managed with the First-Come-First-Served (FCFS) principle, often more complex dispatching rules are employed to improve the system performance. These rules can use more than one attribute of the jobs or might incorporate information about the other queues in the system (Mönch, Fowler, and Mason 2012). The application of various dispatching rules affects the distribution of all the performance measures such as the cycle time and also the characteristics of the flows in the system. For this reason, analysis of queueing networks with the aforementioned characteristics is often performed using simulation. Simulating a complex system with these characteristics can be computationally demanding. The computational inefficiency

CONTACT Barış Tan btan@ku.edu.tr College of Administrative Sciences and Economics, College of Engineering, Koç University, Rumeli Feneri Yolu, Istanbul 34450, Turkey

of simulation is especially pronounced when the simulation model is embedded in an optimization problem (Liu et al. 2021).

With the drastic increase in the availability of data about manufacturing systems in the industry 4.0 era, machine learning methods that provide data-driven solutions have started being used (Kuo and Kusiak 2019; Angelopoulos et al. 2020). Machine learning methods can be deployed for evaluating and controlling various aspects of manufacturing systems including planning, controlling operations and processes, and estimating the behavior of systems for a given structure (Priore et al. 2001; Can and Heavey 2012; Cadavid et al. 2020).

The advancement of machine learning tools allows combining the generality of simulation with the computational efficiency of analytical methods. Recently, Tan and Khayyati (2021) presented a supervised-learning-based approximation method (SLQNA) to analyze open queueing networks with single servers that operate according to First-Come-First-Serve (FCFS) service discipline with correlated interarrival and service times. SLQNA uses local simulations to generate training data for building supervised learning blocks that approximate the behavior of the various blocks in the queueing network. The predicted output characteristics of these blocks are then fed into the following block in order to analyze a network. The results obtained with SLQNA are significantly more accurate and computationally efficient compared to the other existing methods.

In this work, we extend SLQNA algorithm to analyze the performance of open queueing networks with multiple servers that operate with different sequencing rules including FCFS, Last-Come-First-Served (LCFS), Service-in-Random-Order (SIRO), Shortest Processing Time (SPT), Longest Processing Time (LPT) separated with infinite capacity buffers, and with batching, merge, and split processes. A given network is decomposed into delay, merge, split, and batching blocks. We generate the training data by simulating different blocks under these service disciplines for a wide range of input variables in a computer cluster. We then use this data together with the available analytical approximations to train 188 different Gaussian Process Regression (GPR) models for each of the output variables for each block. The predicted output variables for each block are then fed into the next block in the network to obtain the network performance measures. Using already-trained models for different blocks allows obtaining fast and accurate predictions. Therefore, the algorithm we present eliminates the need for generating new data and training different models based on the specific structure of a given network.

Our extension also yields an accurate approximation for the total cycle time distribution in addition to the characteristics of the flow at different points of a given network. The obtained cycle time distribution allows determining a lead time that can be promised for the completion of a given order with the desired service level accurately. For example, the probability that the cycle time is less than or equal to the quoted lead time can be used as a service level measure. By using the accurate approximation method presented in this study, we also analyze the effects of sequencing rules on the cycle time distribution and the departure process in a queueing network and discuss how the proposed approximation method can be used to determine the solution of a resource allocation problem in a manufacturing system.

The main contributions of this study are three fold. First, we present an approximation algorithm to analyze open queueing networks with multiple servers that operate under different service disciplines and with correlated interarrival and service times, batching, merge, and split blocks separated with infinite capacity buffers. The method we presented is the first approximation method developed to analyze such queueing networks. Second, we present an approximation method for the total cycle time distribution. Third, through a large set of numerical experiments, we show the effects of sequencing rules on the cycle time distribution and departure processes in an open queueing network and show how a resource allocation and service discipline determination problem can be solved by using the proposed approximation method.

The remainder of this paper is organized as follows. Section 2 reviews the pertinent literature. Section 3 describes the building blocks of the SLQNA algorithm and the method for generating the training data for supervised learning. Section 4 discusses the supervised-learning-based prediction model for the multi-server delay block under different service disciplines. Section 5 gives the SLQNA algorithm for analyzing the network and predicting its cycle time distribution. Section 6 gives the numerical experiments conducted to assess the performance of the SLQNA algorithm, analyze the effects of the service disciplines on the cycle time distribution, and a case study. Finally, Section 7 concludes the paper.

2. Literature review

Queueing networks have been used in modeling a wide range of systems including manufacturing systems, transportation networks, health care facilities and biochemical processes (Jiang and Giachetti 2008; Zhang and Pavone 2016; Fan, Ma, and Li 2020; Nitz et al. 2021).

Queueing networks have been extensively studied by using simulation and approximation methods. The approximation methods are based on decomposition and

diffusion approaches (Govil and Fu 1999). Decomposition methods involve using the available approximation formulas for single queues for analyzing the blocks of a network one by one and feeding the output of one block to the next one (Kuehn 1979; Buzacott and Shanthikumar 1993; Hopp and Spearman 2011). Diffusion methods are used to analyze queueing networks in heavy traffic (Harrison and Nguyen 1990).

For a multi-server queue with i.i.d interarrival and service times operating according to FCFS discipline, the analytical results are limited to systems with service, failure, and repair times modeled as phase-type distributions (Yang and Alfa 2009; Shin and Moon 2021). For general distributions, there are several approximations for the average performance measures, e.g. Wu and Chan (1989) and Kimura (1994).

Decomposition methods have been used to analyze serial lines with parallel stations at each stage (Diamantidis et al. 2020) and assembly/disassembly systems (Jeong and Kim 1999; Patchong and Willaeys 2001). The approximation methods for queueing networks have been extended to various settings, e.g. transient analysis of multi-product serial lines (Chen et al. 2020; Wang, Huang, and Li 2021), systems with blocking and feedback and multi-class systems (Zhang et al. 2017; Shin et al. 2019).

Applying queueing network analysis methods to analyze complex manufacturing systems such as semiconductor manufacturing plants requires extension of the existing methods in order to capture the characteristics that are common in manufacturing. Shanthikumar, Ding, and Zhang (2007) characterize the features that are needed in queueing network models of semiconductor manufacturing systems as including batching, multi-server stations, and correlated arrival and service times. They attribute the inaccuracies of queueing network analysis methods in describing semiconductor manufacturing systems partially to ignoring autocorrelation in arrival processes and service times. Kumar and Kumar (2001) discuss the importance of studying scheduling in semiconductor manufacturing and more specifically the usage of sequencing rules. Additionally, they highlight FCFS, SPT and LPT as commonly used sequencing rules. In order to address the need to meet customers' short cycle time requests with a high service level, the queueing network analysis methods should also yield accurate distribution of the cycle time. Most of the existing methods only yield the average cycle time and the cycle time distribution can be determined only for some restricted cases, e.g. Harrison (1984) and Alkaff, Qomarudin, and Wiratno (2020).

Recently, Markov arrival processes (MAP) have been used to describe the flows between the blocks in a queueing network with the arrival and the service time autocorrelation (Manafzadeh Dizbin and Tan 2019; Horváth, Horváth, and Telek 2010). Horváth, Horváth, and Telek (2010) use MAPs to analyze queueing networks with single servers operating under FCFS. Manafzadeh Dizbin and Tan (2019) analyze the effects of correlation on interarrival and service times on the performance of a production/inventory system controlled with a base-stock policy by using MAP models. MAPs have been also used in analysis of the conditional sojourn time of a randomly selected customer in a single-server system with the SIRO discipline (Ghosh and Banik 2018). However, there is no MAP-based analysis of networks with multi-server stations that operate under SPT and LPT sequencing rules available. Likewise, MAP-based multi-sever queueing systems under FCFS discipline are computationally prohibitive due to the exponential increase of the state space size with adding more servers.

Many of the aspects of the sequencing rules considered in this work have not been addressed in the context of queueing network analysis. The number of studies on the analysis of single queues under different service disciplines with i.i.d interarrival and service times is also limited. Schrage and Miller (1966) study the waiting time for the M/G/1 queue with preemptive shortest remaining processing time discipline. Assuming Poisson arrivals, Grosof, Scully, and Harchol-Balter (2018) provide bounds for a preemptive multi-server system with the shortest remaining processing time discipline. Takagi (1996) gives approximations for the variance of the cycle time of an M/G/1 queue under the FCFS, SIRO and LCFS service disciplines.

Our approximation method is based on using supervised learning with the training data generated with simulation. Machine learning has been used in tandem with simulation to analyze production systems (Arinez et al. 2020). However, most of these studies consider a given structure of a system and train a model for a given specific system to predict average performance measures, e.g. Can and Heavey (2012) and Boulas, Dounias, and Papadopoulos (2017).

We contribute to the literature by proposing a method to analyze a queueing network composed of multiple servers separated with infinite capacity buffers at each stage operating under different service disciplines with correlated arrival and service times. Our method allows us to model two aspects of queueing networks simultaneously and accurately: the multiple servers operating under various sequencing rules and correlated interarrival and processing times. These aspects have been studied separately or in the context of single queues. Furthermore, as opposed to the previous machine learning-based methods, the approach we propose can be used to analyze

a given system without the need to generate the training data and building a prediction model again for the new system.

3. Building blocks for constructing queueing networks

In the following, we describe the Delay, Split, Merge and Batching building blocks used for modeling queueing networks. Figure 1 illustrates the building blocks of the queueing network with their input and output parameters and Table 1 gives the details of these blocks.

3.1. Notation

In this paper, we compare the predictions obtained by our supervised learning-based algorithm for the building blocks and for the queueing network with simulation and analytical approximations. Throughout the paper, for a given parameter X that is calculated based on a random variable x, we use $\tilde{X}$ for an analytical approximation for X, $\hat{X}$ for the GPR approximation for the building blocks or SLQNA approximation for the network for X and $\check{X}$ for the simulation approximation for X.

We focus on the *cycle time* as the main performance measure in this study. The cycle time is the time it takes for a flow unit to go through a given part of the network from the arrival time until the departure time. We refer the total time from the arrival to the first block in the network until the departure from the last block as *the total cycle time*. Since both the arrival time and the service time are random variables, the cycle time in each block and the total cycle time in the network are also random variables.

In particular, we denote the cycle time of a building block by ct, the average cycle time through a building block by CT, the analytical approximation, GPR approximation, and simulation approximation of CT as $\tilde{CT}$,$\hat{CT}$, $\check{CT}$ respectively. Similarly, the total cycle time and the expected total cycle time in the network are denoted by $\boldsymbol{ct}$ and $\mathbf{CT}$ respectively. The SLQNA approximation, and simulation approximation of the average total cycle time are denoted as $\tilde{\mathbf{CT}}$, $\check{\mathbf{CT}}$ respectively. When a performance measure related to the building block k is used in the analysis of a network, we denote the parameter X obtained for the building block k as X_k. The index k is omitted in the analysis of the building blocks in isolation. Table 2 gives the main notation used throughout the paper. The notation used locally in Sections 4–6 are not included in Table 2. Table 3 gives the description of the abbreviations used throughout the paper.

3.2. A general delay block with parallel servers and service disciplines

Overall, we consider 11 different delay blocks together with the Split, Markov Chain Split, Merge, and Batching blocks. The delay blocks are differentiated based on their

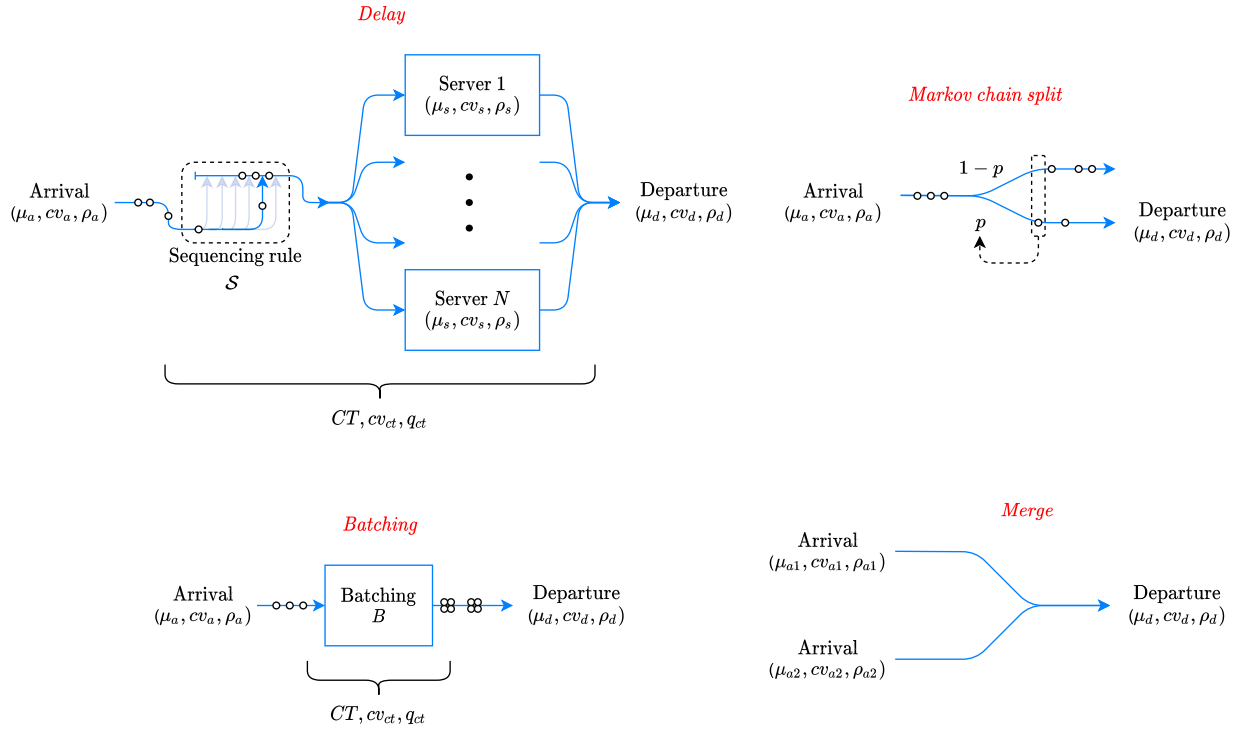

Figure 1. The building blocks of the queueing network.

Table 1. Description of the building blocks.

Block	Block type	Sequencing rule	Arrivals	Service times	Number of servers
1	Delay	FCFS	Correlated	Correlated	Single
2	Delay	LCFS	Correlated	Correlated	Single
3	Delay	SIRO	Correlated	Correlated	Single
4	Delay	SPT	Correlated	i.i.d	Single
5	Delay	LPT	Correlated	i.i.d	Single
6	Delay	FCFS	Correlated	Correlated	Multiple
7	Delay	LCFS	Correlated	Correlated	Multiple
8	Delay	SIRO	Correlated	Correlated	Multiple
9	Delay	SPT	Correlated	i.i.d	Multiple
10	Delay	LPT	Correlated	i.i.d	Multiple
11	Delay		Correlated	i.i.d	Infinite
12	Split		Correlated		
13	MC Split		Correlated		
14	Merge		Correlated		
15	Batching		Correlated		

Table 2. Description of notation.

Notation	Description
ct	Cycle time
CT	Mean cycle time
$\mathbf{CT}$	Mean total cycle time
$\hat{\mathbf{CT}}$	SLQNA approximation for total cycle time
$\check{\mathbf{CT}}$	Simulation approximation for total cycle time
cv_{CT}	Coefficient of variation of the cycle time
μ_a, cv_a, ρ_a	Mean, cv and first-lag autocorrelation of the arrival process
$\mu_{a1}, cv_{a1}, \rho_{a1}, \mu_{a2}, cv_{a2}, \rho_{a2}$	Mean, cv and first-lag autocorrelation of the incoming arrival processes to a merge block
μ_s, cv_s, ρ_s	Mean, cv and first-lag autocorrelation of the service times
μ_d, cv_d, ρ_d	Mean, cv and first-lag autocorrelation of the departure process
N	Number of servers in the Delay block
$\mathcal{S}$	Service discipline used by the Delay block
p_1, p_2	Probabilities governing the split process
B	Batch size
$q_{ct} = (q_{ct,1}, q_{ct,2}, \ldots, q_{ct,10})$	Quantiles of the cycle time distribution

Table 3. Description of abbreviations.

Abbreviation	Definition
SLQNA	Supervised-learning-based queuing network analyzer
GPR	Gaussian process regression
FCFS	First-come-first-served
SIRO	Service-in-random-order
LCFS	Last-come-first-served
SPT	Shortest processing time
LPT	Longest processing time
MAP	Markov arrival processes
MAE	Mean absolute error
MAPE	Mean absolute percentage error
APE	Absolute percentage error
`fft`	Fast Fourier transform
`ifft`	Inverse fast Fourier transform

dispatching rule, the number of servers, and the dependence of interarrival and service times. Single-server delay block with FCFS sequencing rule (Block 1), Split (Block 12), Markov Chain Split (Block 13), Merge (Block 14) and Batching blocks (Block 15) were developed and presented in (Tan and Khayyati 2021). We present a General Delay Block for 11 new cases for different service disciplines and number of servers (Blocks 2–11) in this section.

We consider dispatching rules that are local for a queue, i.e. that only depend on the information that is related to this queue and the jobs in it and use one attribute for making the dispatching decision. Hence, we consider the first-come first-server (FCFS), last-come first-serve (LCFS), service-in-random-order (SIRO), shortest processing time first (SPT) and longest processing time first (LPT) disciplines. Note that SPT and LPT apply to the queues where upon arrival, the processing time for the part is observed and will not change based on the part being processed earlier or later. For this reason, for the SPT and LPT queues, we only consider i.i.d service times. The SPT and LPT blocks are specifically noteworthy in studying queueing networks with correlated arrivals and service times. Because they can generate autocorrelation in the system even if the external flows and the service times are all i.i.d. In Section 6.3, we investigate the effects of the service discipline on the departure processes. For the FCFS, LCFS and SIRO disciplines, we consider correlated service times where the autocorrelation is a function of the characteristics of the servers. The inclusion of the SIRO sequencing rule is motivated by its ability to approximate sequencing rules that rely on attributes a job can have other than the processing time and the arrival time, e.g. the due date. The sequencing rule used in the Delay block is denoted with $\mathcal{S}$.

The other characteristic of a delay block is the number of servers N. We consider single-server queues, multi-server and infinite-server queues with homogeneous servers. The infinite servers block can also be used for modeling transport operations in production systems.

In addition to N and $\mathcal{S}$, the inputs to the Delay building blocks are the characteristics of the incoming stream

and the characteristics of the processing times. The characteristics of the incoming stream is summarized by the mean interarrival time μ_a, the coefficient of variation of the interarrival time cv_a, and the first-lag autocorrelation of the interarrival time ρ_a. We assume an exponential autocorrelation function, i.e. the kth-lag autocorrelation of the interarrival time process is ρ_a^k.

The processing time for one part is s. The characteristics of the service time stream for each server are summarized by the mean service time $\mu_s = \mathbb{E}[s]$, the coefficient of variation of the service time cv_s, and the first-lag autocorrelation of the service time ρ_s. We assume that the kth-lag autocorrelation of the service time process is ρ_s^k.

The outputs of the Delay building block include the characteristics of the outgoing stream characterized by the mean inter-departure time μ_d, the coefficient of variation of the inter-departure time cv_d, and the first-lag autocorrelation of the inter-departure time ρ_d.

In Section 5, we present three approximation methods to predict the cycle time distribution in queueing networks. These methods use either the mean and the coefficient of variations of the cycle time of the general delay block, CT and cv_{ct}, or the quantiles of the cycle time distribution $q_{ct} = (q_{ct,1}, q_{ct,2}, \ldots, q_{ct,10})$. In order to implement these methods, (CT, cv_{ct}) and q_{ct} are used as the other outputs of the General Delay Block.

A supervised learning algorithm given in Section 4 is used to predict the outputs of the General Delay building block that operates under the given service discipline $\mathcal{S}$ and the number of servers N to obtain $\hat{\mu}_d$, $\hat{cv}_d$, $\hat{\rho}_d$, $(\hat{CT}, \hat{cv}_{ct})$, and $\hat{q}_{ct}$ for the given inputs (μ_a, cv_a, ρ_a) and (μ_s, cv_s, ρ_s).

3.3. Other blocks to construct queueing networks

3.3.1. Split blocks

The split block splits an arriving stream into two departing streams based on the given split process. We consider the Bernoulli split process and Markov chain split process. The inputs to the Bernoulli splitting block are the characteristics of the incoming stream (μ_a, cv_a, ρ_a) and the split probability p and its outputs are the characteristics of the outgoing stream in the selected route (μ_d, cv_d, ρ_d).

The Markov chain splitting process is defined by two probabilities p_1 and p_2. p_1 is the probability of a part going to the first downstream route if the previous part has gone to the first downstream route and p_2 is the probability of a part going to the second downstream route if the previous part has gone to the second downstream route. The inputs to Markov chain splitting block are the characteristics of the incoming stream (μ_a, cv_a, ρ_a) and the Markov chain $\begin{bmatrix} p_1 & 1-p_1 \\ 1-p_2 & p_2 \end{bmatrix}$ that describes the split process. Its outputs are the characteristics of the outgoing stream in the selected route (μ_d, cv_d, ρ_d).

3.3.2. Merge block

The merge block is used for modeling two correlated streams merging into one stream. The inputs to the merge block are the characteristics of the two incoming streams $(\mu_{a1}, cv_{a1}, \rho_{a1})$ and $(\mu_{a2}, cv_{a2}, \rho_{a2})$ and its outputs are the characteristics of the merged stream (μ_d, cv_d, ρ_d).

3.3.3. Batching block

The batching block allows modeling the change of the number of units processed at different stages of the network. The inputs to this block are the characteristics of the incoming stream (μ_a, cv_a, ρ_a) and the batch size B and its outputs are the characteristics of the outgoing stream (μ_d, cv_d, ρ_d). Without loss of generality, we assume forming a batch when B parts are already available is instantaneous.

4. A supervised-learning-based prediction method for the output characteristics of the general delay block

The procedure we use for developing the supervised-learning-based prediction method first generates the training data for a supervised learning algorithm by simulating the dynamics for different blocks with correlated interarrival and service times for a wide range of system parameters. Then, a supervised learning algorithm is used as a supervised learning method to predict the output characteristics of the blocks. Finally, the output characteristics are fed into the following block to analyze a queueing network. The general method presented in this study is based on the SLQNA algorithm given in Tan and Khayyati (2021). In Sections 4.1, 4.2, and 4.3, we summarize the steps of SLQNA for completeness.

We generate 4,619,916 different cases with simulation to generate the training data for the General Delay Block in a computer cluster. For the 11 different General Delay Blocks with different service disciplines and number of stations, we trained 165 different GPR models to predict the outputs for each block for the given inputs. This library of the trained models is used to analyze any given queueing network without the need to generate the training data and build a prediction model again.

4.1. Supervised learning algorithms

Supervised learning refers to learning a function that resembles the relationship between the input and output

variables based on training data that consists of samples with both the inputs and outputs available. Many methods and algorithms have been proposed for this task, a number of which have proved efficient in dealing with noisy and large datasets, e.g. random forests, artificial neural networks, Gaussian process regression (GPR) and support vector machine (Breiman 2001; Rasmussen and Williams 2006; Gurney 1997). We use GPR as the supervised learning algorithm.

4.2. Gaussian process regression

GPR is a kernel based method that uses Gaussian processes to model the data. A Gaussian process is a process where the distribution of any observed finite set of points is Gaussian. In addition to the mean prediction value, GPR yields the variance for the prediction. The mean and variance of the prediction can be used to obtain the confidence interval for the prediction. We choose GPR as the main learning method in this work given its ability to model noisy data and its efficiency in modeling similar datasets (Tan and Khayyati 2021).

Based on Quinonero-Candela and Rasmussen (2005) and Rasmussen and Williams (2006), we give a brief description of GPR in the following. For a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i), i \in \{1, \ldots, n\}\}$ with n data-points, GPR models the data using

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \tag{1}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$ and ϵ_i represents the variance of the noise. Let $f_* = f(\mathbf{x}_*)$ denote the response value for a new observation $\mathbf{x}_*$ approximated by GPR.

Let X and X_* denote the matrix of the training and the test data respectively and let $\mathbf{y}$ denote the vector of the observed outputs. Additionally, let $K_{A,B}$ denote the kernel matrix for the data matrices A and B defined as $[K_{A,B}]_{i,j} = k(A_i, B_j)$, where A_i and B_j are the vectors representing the ith and jth data points in A and B respectively, and $k(A_i, B_j)$ is the covariance function.

Gaussian process regression assumes a Gaussian process prior over functions and since any finite collection of the points of a Gaussian process follows a Gaussian distribution,

$$p(\mathbf{f} \mid \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \mathcal{N}(0, K_{X,X}), \tag{2}$$

where $\mathbf{f} = [f(\mathbf{x}_1) f(\mathbf{x}_2) \ldots f(\mathbf{x}_n)]^T$ is the vector of latent variables. Using the Bayesian rule, predictive distribution for a GPR can be calculated as

$$p(\mathbf{f}_* \mid \mathbf{y}) = \mathcal{N}(K_{X_*,X}(K_{X,X} + \sigma_n^2 I)^{-1}\mathbf{y}, K_{X_*,X_*} - K_{X_*,X}(K_{X,X} + \sigma_n^2 I)^{-1} K_{X,X_*}), \tag{3}$$

where I is the identity matrix.

4.3. Generating the training data by simulating the general delay block with correlated interarrival and service times

The training data is generated by simulating the dynamics of the General Delay Block with correlated interarrival and service times under different service disciplines. For modeling the interarrival and service times, we use the Weibull distribution with the probability density function

$$f(x) = \frac{\alpha}{\beta^\alpha} x^{\alpha-1} e^{-(x/\beta)^\alpha}, \quad x > 0. \tag{4}$$

For a given coefficient of variation and mean, the parameters of the Weibull distribution can be uniquely determined.

For modeling the autocorrelation of the processes, we use the exponential decay model that only uses the first-lag autocorrelation ρ_1 to construct the autocorrelation function:

$$\rho_k = \rho_1^k, \quad i = 1, 2, \ldots. \tag{5}$$

Where ρ_i denotes the ith lag autocorrelation of an arrival process. The exponential decay model is a good fit for systems with Weibull inter-event times.

The procedure to generate correlated random variables involves generating normally distributed random variables with the desired autocorrelation structure and transforming them, first to correlated uniform variates and then, to correlated Weibull random variables. The parameters of the initial normal random variables are chosen while considering the changes that will result in the first-lag autocorrelation from the transformations. For the details on the generation of correlated inter-event times and simulating different building blocks, we refer the reader to Tan and Khayyati (2021).

4.4. Training data

The method we propose trains a supervised learning method with simulation data for each building block in isolation and combines the predictions obtained from the trained building blocks in a decomposition algorithm. The training is done offline only once and the library of trained models for different delay, batching, split and merge blocks are combined for a given network structure. This approach allows analysis of a given network without simulating and training a new model for each network.

Table 4 gives the range of parameters used to generate the training data for the building blocks. For generating the data for the delay blocks for each parameter set, we have used traces of length 10,000 and replicated the simulations 100 times. In total, the delay blocks for 4,619,916 different parameter combinations need to be simulated.

Table 4. Parameter sets for the queueing network blocks.

Block type	Parameter	Range	Number of cases
Delay (single-server FCFS, SIRO, LCFS)	μ_a	$\{1/0.1, 1/0.2, \ldots, 1/0.9\}$	142,884
	cv_a	$\{0.1, \ldots, 1.4\}$	
	ρ_a	$\{-0.4, \ldots, 0.4\}$	
	μ_s	$\{1\}$	
	cv_s	$\{0.1, \ldots, 1.4\}$	
	ρ_s	$\{-0.4, \ldots, 0.4\}$	
	N	$\{1\}$	
Delay (single-server SPT, LPT)	μ_a	$\{1/0.1, 1/0.2, \ldots, 1/0.9\}$	15,876
	cv_a	$\{0.1, \ldots, 1.4\}$	
	ρ_a	$\{-0.4, \ldots, 0.4\}$	
	μ_s	$\{1\}$	
	cv_s	$\{0.1, \ldots, 1.4\}$	
	ρ_s	$\{0\}$	
	N	$\{1\}$	
Delay (multiple server FCFS, SIRO, LCFS)	μ_a	$\{1/0.1, 1/0.2, \ldots, 1/0.9\}$	1,285,956
	cv_a	$\{0.1, \ldots, 1.4\}$	
	ρ_a	$\{-0.4, \ldots, 0.4\}$	
	μ_s	$1 \times N$	
	cv_s	$\{0.1, \ldots, 1.4\}$	
	ρ_s	$\{-0.4, \ldots, 0.4\}$	
	N	$\{2, \ldots, 10\}$	
Delay (multiple server SPT, LPT, infinite server)	μ_a	$\{1/0.1, 1/0.2, \ldots, 1/0.9\}$	142,884 for SPT and LPT and 15,876 for infinite server
	cv_a	$\{0.1, \ldots, 1.4\}$	
	ρ_a	$\{-0.4, \ldots, 0.4\}$	
	μ_s	$1 \times N$	
	cv_s	$\{0.1, \ldots, 1.4\}$	
	ρ_s	$\{0\}$	
	N	$\{2, \ldots, 10\}$	
Split	μ_a	$\{1\}$	630
	cv_a	$\{0.1, \ldots, 1.4\}$	
	ρ_a	$\{-0.4, \ldots, 1.4\}$	
	p	$\{0.1, \ldots, 0.9\}$	
MC Split	μ_a	$\{1\}$	5040
	cv_a	$\{0.1, \ldots, 1.4\}$	
	ρ_a	$\{-0.4, \ldots, 0.4\}$	
	p_1	$\{0.1, \ldots, 0.9\}$	
	p_2	$\{0.1, \ldots, 0.9\}$	
Merge	μ_{a1}	$\{1\}$	158,760
	cv_{a1}	$\{0.1, \ldots, 1.4\}$	
	ρ_{a1}	$\{-0.4, \ldots, 0.4\}$	
	μ_{a2}	$\{\frac{1}{0.1}, \ldots, \frac{1}{0.9}, 1\}$	
	cv_{a2}	$\{0.1, \ldots, 1.4\}$	
	ρ_{a2}	$\{-0.4, \ldots, 0.4\}$	
Batching	μ_a	$\{1\}$	2052
	cv_a	$\{0.1, \ldots, 1.1\} \cup \{1.4\}$	
	ρ_a	$\{-0.4, \ldots, 0.4\}$	
	B	$\{2, \ldots, 20\}$	

The simulation of all these cases would have required more than 8 years of computational time on a personal computer with a `Intel(R) Core(TM) i5-3340M 2.7GHz` CPU. Instead, we have generated samples of 30,000 parameter combinations from the larger blocks, i.e. the multi-server blocks resulting in 852,048 delay simulations in total. We have used a computer cluster with 139 nodes enabling the generation of the data points in less than three weeks. Similarly, the simulations for the remaining blocks would require more than 100 days on the same personal computer. This time was shortened to less than a week using the cluster. Once these costly computations have been performed and the GPR models have been trained using them, the SLQNA algorithm does not require generating new data points or any further training related to the system it is being used to analyze. The trained models can be used directly and very efficiently to predict the outputs for the given inputs.

4.4.1. Performance of GPR for predicting the output characteristics in the general delay block

Once the training data is generated, GPR is used to predict the outputs of the General Delay Block and obtain $(\hat{\mu}_d, \hat{cv}_d, \hat{\rho}_d)$, $(\hat{CT}, \hat{cv}_{ct})$, $\hat{q}_{ct}$ for the inputs (μ_a, cv_a, ρ_a), (μ_s, cv_s, ρ_s) for the given N and $\mathcal{S}$. Since there are 15 output values for the set of 6 input parameters for the Delay blocks, 15 different GPR models are trained for each of 11 different Delay blocks. Including all the blocks, in total, 188 GPR models have been trained by using GPR.

The matrix inversion step of the GPR algorithm given in Equation (3) is computationally costly. As the computational burden of the inversion process

Table 5. The accuracy of GPR predictions for $(\hat{cv}_d, \hat{\rho}_d)$, $(\hat{CT}, \hat{cv}_{ct})$ of the Multi-server Delay block.

			MAE				MAPE			
Sequencing rule	Service times	Number of servers	$\hat{cv}_d$	$\hat{\rho}_d$	$\hat{CT}$	$\hat{cv}_{ct}$	$\hat{cv}_d$	$\hat{\rho}_d$	$\hat{CT}$	$\hat{cv}_{ct}$
FCFS	Correlated	Single	0.001	0.002	0.122	0.008	0.266	0.754	3.614	1.100
LCFS	Correlated	Single	0.001	0.002	0.122	0.029	0.266	0.754	3.614	2.759
SIRO	Correlated	Single	0.001	0.002	0.122	0.011	0.266	0.754	3.614	1.189
SPT	i.i.d	Single	0.001	0.001	0.016	0.013	0.209	1.050	0.617	1.014
LPT	i.i.d	Single	0.001	0.001	0.149	0.011	0.208	0.861	2.354	1.113
FCFS	Correlated	Multiple	0.012	0.015	0.090	0.006	1.558	1.106	0.863	1.223
LCFS	Correlated	Multiple	0.012	0.015	0.090	0.024	1.558	1.106	0.863	2.197
SIRO	Correlated	Multiple	0.012	0.015	0.090	0.010	1.558	1.106	0.863	1.651
SPT	i.i.d	Multiple	0.007	0.007	0.100	0.033	0.887	6.688	1.564	4.834
LPT	i.i.d	Multiple	0.009	0.007	0.876	0.040	1.100	4.535	9.550	7.032
–	i.i.d	Infinite	0.001	0.001	0.000	0.000	0.185	6.668	0.000	0.000

increases rapidly with the training sample size, the training sample size is set as 10,000 based on numerical experiments that investigated the tradeoff between the accuracy and the computation time depending on the sample size. Note that SLQNA requires training only once, hence, larger sample sizes can be used based on the availability of computational time and resources.

In Table 5, we report the accuracy for the GPR prediction of the $(\hat{cv}_d, \hat{\rho}_d)$, $(\hat{CT}, \hat{cv}_{ct})$ of Multi-server Delay blocks in isolation. Since the mean departure rate is equal to the mean arrival rate, $\hat{\mu}_d = \mu_a$, there is no error introduced to predict the departure process mean. Similarly, for the infinite server block, since there is no waiting, the cycle time process is the same as the service process, i.e. $\hat{CT} = \mu_s$, $\hat{cv}_{ct} = cv_s$ and the prediction error is reported as 0 for these variables. We report both Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) values. The accuracy of the prediction of the output parameters of the Batching, Merge, and Split blocks has been reported in Tan and Khayyati (2021). These results indicate that GPR yields very accurate predictions for the output parameters for modeling all the blocks considered in this work. Since they yield accurate predictions for the output parameters for the given input parameters for each building block, SLQNA algorithm presented in Section 5 gives accurate predictions for queueing networks that uses the predictions for the building blocks.

4.4.2. Analytical approximations for the average cycle time under different service disciplines for a single queue

Analytical approximations could be used to predict the output parameter of the General Delay Block. However, there are no analytical approximations for a multi-server G/G/N queue where each parallel server has a general distribution, operates with different service characteristics and with correlated interarrival and service times. In the following, we discuss five single-queue approximations given in the literature for special cases of the general model we consider. The SLQNA algorithm has the ability to incorporate known approximation formulas to improve its performance even if the approximation formulas are not very accurate themselves. These approximations can be used as features in GPR approximation.

Approximation for G/G/N queue with FCFS and i.i.d interarrival and service times: An approximation formula for the average cycle time for the multiple server FCFS delay block with i.i.d interarrival and service times is given by Sakasegawa (1977) as

$$\tilde{CT} = \left(\frac{c_a^2 + c_s^2}{2}\right)\left(\frac{u^{\sqrt{2(m+1)}-1}}{m(1-u)}\right)\mu_s + \mu_s, \qquad (6)$$

where u denotes the utilization in the block. Figure 2 compares the performance of this analytical approximation with the GPR predictions for the multi-server FCFS delay block. The MAPE value for GPR prediction is 0.89 while the MAPE for the analytical G/G/N approximation given by Equation (6) is 4.86. The results show that GPR predictions we obtained are significantly more accurate compared to the analytical G/G/N approximation. We use the analytical approximation for the average cycle time for the G/G/N queue with i.i.d interarrival and service times given in Equation (6) as another input variable for training the GPR model for the system with correlated interarrival and service times. Although, the analytical approximation is not accurate, including the analytical approximation in the training of the GPR model improves the prediction accuracy.

Approximation for M/G/1 queue with FCFS, SIRO, and LCFS service disciplines and i.i.d interarrival and service times: The expected cycle times for a G/G/1 system with i.i.d interarrival and service times under FCFS, SIRO, and LCFS service disciplines are the same. For a single-server queue with i.i.d exponential interarrival and i.i.d general service times (M/G/1), Takagi (1996) derives the approximation formulas for the second moment of the cycle time

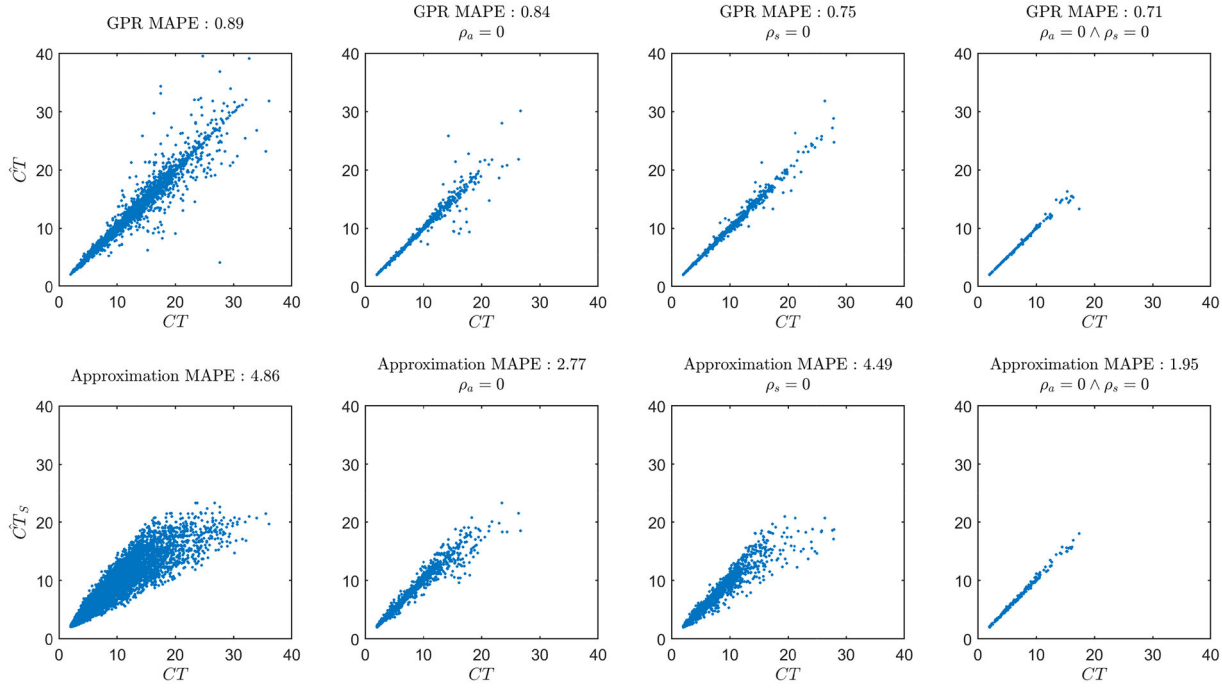

Figure 2. The performance of GPR and the analytical G/G/N approximation (Equation (6)) in predicting the average cycle time for the block with parallel servers and FCFS service rule and the effect of autocorrelation on their performance.

$\tilde{CT}^2$ for FIFO, SIRO, and LCFS service disciplines. For the FCFS service discipline,

$$\begin{aligned}\tilde{CT}^2 = \frac{1}{6(1-\mu_s\lambda_a)^2}\big(&12\mu_s^2 - 12\mu_s^3\lambda_a + 6\mu_s\mathbb{E}[s^2]\lambda_a \\ &- +3\mathbb{E}[s^2]^2\lambda_a^2 \\ &- 2\mu_s\mathbb{E}[s^3]\lambda_a^2 + 2\left(-6\mu_s^2 + 3\mathbb{E}[s^2] + 6\mu_s^3\lambda_a\right. \\ &- \left.-6\mu_s\mathbb{E}[s^2]\lambda_a + \mathbb{E}[s^3]\lambda_a\right)\big),\end{aligned} \tag{7}$$

for the SIRO service discipline,

$$\begin{aligned}\tilde{CT}^2 = &\frac{\mathbb{E}[s^2]\left(12 - 32\mu_s\lambda_a + 31\mu_s^2\lambda_a^2 - 13\mu_s^3\lambda_a^3\right)}{(3-2\mu_s\lambda_a)(1-\mu_s\lambda_a)^2(2-\mu_s\lambda_a)^2} \\ &+ \frac{2\left(\frac{\mathbb{E}[s^3]\lambda_a}{3} + \frac{\mathbb{E}[s^2]^2\lambda_a^2}{2(1-\mu_s\lambda_a)}\right)}{(1-\mu_s\lambda_a)(2-\mu_s\lambda_a)},\end{aligned} \tag{8}$$

and for the LCFS service discipline

$$\begin{aligned}\tilde{CT}^2 = &\frac{\mathbb{E}[s^2]^2\lambda_a^2}{2(1-\mu_s\lambda_a)^3} + \frac{\mathbb{E}[s^3]\lambda_a}{3(1-\mu_s\lambda_a)^2} \\ &+ \frac{\mathbb{E}[s^2]\left(1 - 2\mu_s\lambda_a + \mu_s^2\lambda_a^2\right)}{(1-\mu_s\lambda_a)^3}.\end{aligned} \tag{9}$$

In these equations, $\mathbb{E}[s^2]$ is the second moment of the processing time, $\mathbb{E}[s^2] = \mu_s^2(1+cv_s^2)$. $\mathbb{E}[s^3]$ is the third moment of the processing time and it can be calculated for a given processing time distribution. For example, for Weibull distribution, $\mathbb{E}(s^3) = \beta^3\Gamma(1+3/\alpha)$ where the parameters α and β can be determined for given μ_s and cv_s values. Once $\tilde{CT}$ and $\tilde{CT}^2$ are obtained, the analytical approximation for the coefficient of variation of the cycle time can be obtained as $\tilde{cv}_{ct} = \frac{\sqrt{\tilde{CT}^2 - (\tilde{CT})^2}}{\tilde{CT}}$.

Takagi (1996) shows that the cycle time variability is greater for the LCFS discipline compared to the SIRO discipline and greater for the SIRO discipline compared to the FCFS discipline for M/G/1 queue with i.i.d interarrival and service times. We investigate this relation in presence of correlated interarrival and service times.

Approximation for M/G/1 queue with SPT service disciplines and i.i.d interarrival and service times: The average waiting time for the M/G/1 queue with Poisson arrivals, i.i.d service times and shortest remaining processing time discipline is derived by Schrage and Miller (1966) as

$$\begin{aligned}&\tilde{CT} \\ &= \mu_s + \frac{\lambda}{2}\int_0^\infty \left(\frac{\int_0^p t^2\,\mathrm{d}F(t) + p^2[1-F(p)]}{[1-\rho(p)]^2}\right)\mathrm{d}F(p),\end{aligned} \tag{10}$$

where λ is the arrival rate, F the cumulative distribution of the service times and $\rho(p) = \lambda\int_0^p t\,\mathrm{d}F(t)$. The average waiting time for a preemptive system can be used as an approximation for a non-preemptive system considered in this study. The integrals for this approximation do not yield closed form formulas for Weibull processing times and need to be calculated numerically.

Table 6. Comparison of the accuracy of the analytical approximations for the FCFS, SIRO, LCFS and SPT single-server M/G/1 systems with the GPR predictions.

Approximation method	Accuracy	FCFS CT	FCFS cv_{ct}	SIRO cv_{ct}	LCFS cv_{ct}	SPT CT
Analytical	MAPE	32.93	22.28	33.99	44.21	23.76
	MAE	0.93	0.11	0.27	0.31	0.53
GPR	MAPE	3.61	1.10	1.19	2.76	1.56
	MAE	0.12	0.01	0.01	0.03	0.03

Table 6 compares the accuracy of the GPR prediction we present in this study with the analytical approximations for the data set given in Table 4. Note that while the GPR prediction is developed for a G/G/N queue operating with FCFS, LCFS, SIRO, SPT, and LPT service disciplines with correlated interarrival and service times, the analytical approximations are available for a single-server queue operating with FCFS, SIRO, LCFS (given by Takagi (1996)), and SPT (given by Schrage and Miller (1966)) with i.i.d exponential interarrival times and i.i.d service times. Table 6 shows that the GPR predictions we obtained are significantly more accurate compared to the available analytical approximation methods.

5. Approximation methods for the cycle time distribution in open queueing networks

In the following, we present three methods to predict the distribution of the total cycle times in a queueing network.

5.1. Supervised-learning-based queueing network analysis (SLQNA) algorithm

The SLQNA algorithm predicts the various characteristics of a queueing network by modeling its blocks using the trained machine learning models to predict the changes in the flows in the system depending on the blocks they go through. SLQNA characterizes a flow by its rate, coefficient of variation and first-lag autocorrelation. The algorithm goes through the blocks in the system to identify the blocks whose input parameters are known or have been estimated before and estimates their output parameters and continues this until there is no unprocessed block remaining. In this process, SLQNA predicts the mean and the variability of the cycle times for each block in the system.

5.2. Cycle time distribution in open queueing networks

The cycle times at various workstations are not independent in a general queueing network. However, treating these random variables as independent makes the analysis easier and can be fairly accurate in many cases. Under the independence of the cycle times assumption, the distribution of the total cycle time in a system can be approximated as the mixture distribution of the convolution of the cycle time distributions for the individual blocks.

We consider three methods for predicting the cycle time distribution based on the output of SLQNA algorithm. The first method is approximating the cycle time distribution as a Gamma distribution that uses the two-moment predictions of the General Delay Block. The second method fits a Gamma distribution to the cycle time distribution at each block and then determines the total cycle time distribution from the convolution of the distributions for each block. The last approach is based on predicting the quantiles of the cycle time distribution for each block by using GPR and then determining the total cycle time distribution from convolution of the cycle time distributions represented by the quantiles. Figure 3 gives a schematic view of these methods. In the following part, we discuss these methods.

5.2.1. Distribution approximation based on two-moment predictions of the general delay block (`2MomFitGamma`)

In this method, we first determine the first two moments of the total cycle time distribution by using the first and second moments of the cycle time distribution for each block under the independence of the cycle times assumption. Let $r_{i,j}$ denote the index of the ith block on the jth route in the system that a part may travel through and let f_k denote the probability density function of the cycle times for block k. Let L_j denote the number of blocks on the jth path, i.e. the blocks that a part goes through in path j are $r_{1,j}$, then $r_{2,j}, \ldots$ and finally $r_{L_j,j}$. Then, the mean and the coefficient of variation of the total time can be obtained using

$$\hat{\mathbf{CT}} = \sum_j p_j \sum_{k=1}^{L_j} \hat{CT}_{r_{k,j}}, \tag{11}$$

$$\hat{cv}_{\mathbf{ct}} = \frac{\sqrt{\sum_j p_j \left(\sum_{k=1}^{L_j} \left(\hat{cv}_{ct,r_{k,j}} \hat{CT}_{r_{k,j}} \right)^2 + \left(\sum_{k=1}^{L_j} \hat{CT}_{r_{k,j}} \right)^2 \right) - (\hat{\mathbf{CT}})^2}}{\hat{\mathbf{CT}}}, \tag{12}$$

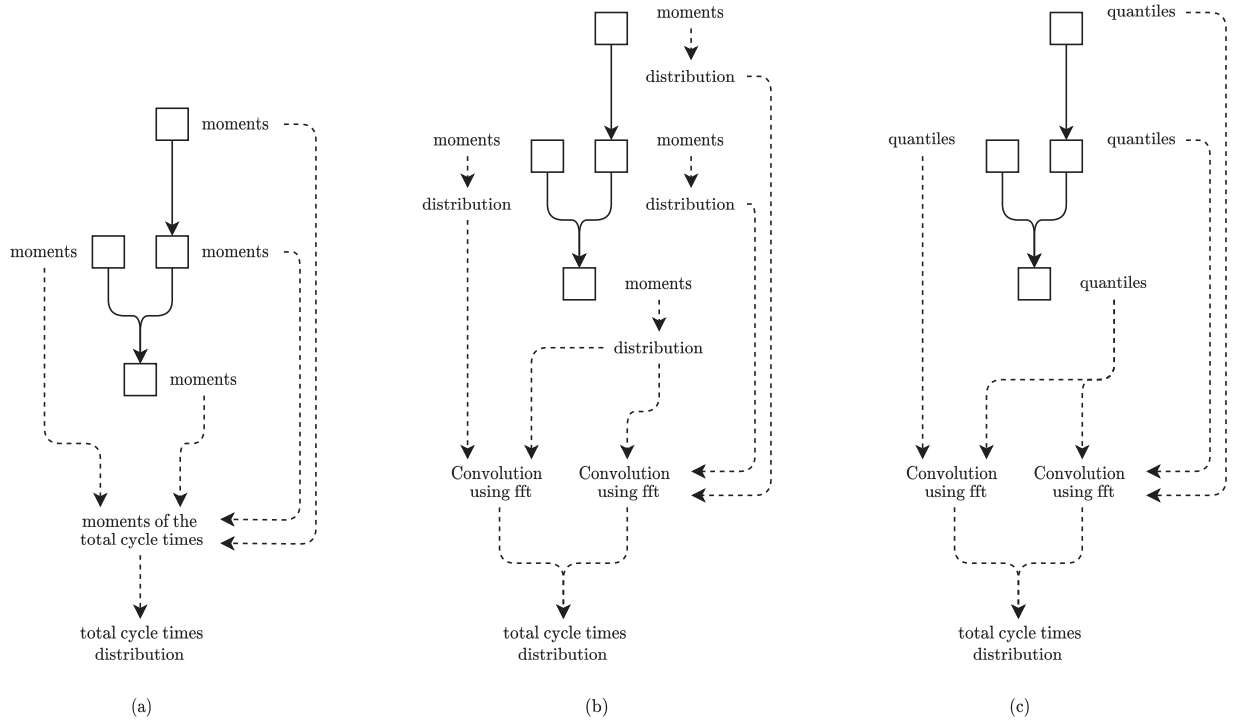

Figure 3. The methods used for predicting the distribution of the cycle time distribution. (a) Using the predicted moments of the total cycle time (`2MomFitGamma`). (b) Using moment based fitted distributions (`GammaConv`). (c) Using the quantiles (`QuanConv`).

Equation (11) gives the mean of the total cycle time distribution as the weighted sum of the average cycle times of each path. Equation (12) first calculates the second moment of the cycle time distribution for each path using the mean and the coefficient of variation of the cycle times for the blocks that are on that path based on the independence assumption of the cycle times at different blocks. Then, the second moment of the cycle time distribution for the network is calculated using the probabilities for the paths. Finally, the second moment of the total cycle time distribution along with its mean is used for calculating the approximation for the coefficient of variation of the total cycle time distribution.

After the first two moments of the total cycle times are predicted, various distributions can be fitted using these moments. Based on our experiments with different distributions, we use the Gamma distribution for modeling the total cycle time. Let k and θ denote the shape and the scale parameters of the Gamma distribution with the probability density function $f(x) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{\frac{-x}{\theta}}$. Based on the estimations about the mean and coefficient of variation of the total cycle time, these parameters can be calculated using

$$k = \left(\hat{cv}_{\mathbf{ct}}\right)^{-2}, \tag{13}$$

$$\theta = \left(\hat{cv}_{\mathbf{ct}}\right)^{2} \hat{\mathbf{CT}}. \tag{14}$$

We refer to this method as `2MomFitGamma` that summarizes the process of determining the first two moments of the total cycle time from the first two moments of the building blocks and then fitting a gamma distribution with the moments obtained for the total cycle time to approximate the distribution.

5.2.2. Predicting the total cycle time distribution using the moments-based fitting of the cycle time distribution of each block (`GammaConv`)

In this method, we first fit a distribution $\hat{f}$ to the cycle time distribution for each block based on the two predicted moments of the cycle time at this block. Then, the approximate cycle time distribution can be calculated as the mixture of the convolution of the cycle times of the building blocks in each path. That is,

$$\hat{f}_{\mathbf{CT}} = \sum_j p_j \left(\hat{f}_{r_{1,j}} * \hat{f}_{r_{2,j}} * \cdots * \hat{f}_{r_{L_j,j}} \right) \tag{15}$$

where $\hat{f}_{r_{i,j}}$ is the approximated probability density function of the cycle time of the ith block in the jth route, and $*$ denotes the convolution operator defined as $(f * g)(z) = \int_{-\infty}^{\infty} f(z-t)g(t)\,\mathrm{d}t$.

This approximate distribution can be calculated using Fast Fourier Transform (`fft`) and inverse Fast Fourier

Transform (ifft) as

$$\hat{f}_{\text{CT}} = \sum_j p_j \times \texttt{ifft}\left(\prod_{k=1}^{L_j} \texttt{fft}\left(\tilde{f}_{r_{kj}}\right)\right), \tag{16}$$

where $\prod$ stands for element-wise product. For implementing the Fourier transform and the inverse transform, we use the discrete Fourier transform. For this purpose, first, the distributions are transformed into sequences. Let $\{\frac{1}{z}, \frac{2}{z}, \ldots, \frac{N}{z}\}$ denote the sequence used for the discretization of the distributions and let $S_g = \{g(\frac{1}{z}), g(\frac{2}{z}), \ldots, g(\frac{N}{z})\}$ and $S_h = \{h(\frac{1}{z}), h(\frac{2}{z}), \ldots, h(\frac{N}{z})\}$ denote the fast Fourier transform of the discretized distributions and let $\{S_g\}_n = g(\frac{1+n}{z})$. Then, the Fourier transform of $g * h$ can be calculated as $S_{g*h} = \{\mathbb{D}(S_g, 1) \times \mathbb{D}(S_h, 1), \mathbb{D}(S_g, 2) \times \mathbb{D}(S_h, 2), \ldots, \mathbb{D}(S_g, N) \times \mathbb{D}(S_h, N)\}$ where

$$\mathbb{D}(S, k) = \sum_{n=0}^{N-1} \{S\}_n e^{-\frac{i2\pi}{N} kn}. \tag{17}$$

We refer to this method as GammaConv to summarize that the total cycle time distribution is approximated from the convolution of the cycle time distributions of the building blocks that are obtained by fitting a Gamma distribution to each building block's cycle time distribution using their first two moments.

5.2.3. Cycle time distribution based on percentile predictions of the general delay block (QuanConv)

In order to implement this method, we model the distribution for each block using 10 quantiles of its distribution with equal probability. Then we train 10 different GPR models for the first quantile and the distances between the next quantiles as the output variables. As these quantities are all positive, there is no issue with the order of the predicted quantiles due to the errors in prediction. After the quantiles are predicted for each block in this manner, for a given sequence of cycle time values, the quantiles are transformed into a discrete distribution and then combined using the mixture and convolution operators similar to the GammaConv approximation. For given quantiles $q_{ct,1}, q_{ct,2}, \ldots, q_{ct,10}$, the discretized distribution can be calculated using $f(n/z) = \frac{1}{11} \times \frac{1}{|P_i|} : \frac{n}{z} \in P_i$, where $P_i = \{\frac{n}{z} : q_{ct,i} \leq \frac{n}{z} \leq q_{ct,i+1}\}$.

We refer to this method as QuanConv to summarize that this method yields an approximation for the total cycle time distribution by determining the convolution of the building blocks' cycle time distributions that are formed by predicting the cycle time quantiles for each block.

We compare these three methods to obtain the cycle time distribution prediction in Section 6.2.1. While QuanConv yields a more accurate prediction for the cycle time distribution, especially when the cycle time distribution is multi-modal, our comparisons show that 2MomFitGamma performs quite well in most of the instances.

6. Numerical experiments

In this section, we discuss the results of four sets of numerical experiments. Section 6.1 gives numerical experiments that are conducted using a network with multiple routes. In this network, parts can travel through different routes of the network that combines the various blocks considered in this study. Section 6.2 gives extensive numerical experiments using serial lines. These experiments show the accuracy and efficiency of SLQNA algorithm to predict the mean, coefficient of variation and distribution of the total cycle time in the production line. Section 6.3 gives extensive numerical experiments that investigate the effects of different sequencing rules on the departure process and discusses the selection of the most suitable sequencing rule based on the desired value of the performance measure of interest.

In these experiments, in order to assess the performance of SLQNA in predicting the cycle time distribution, we use the K-S statistic and the lead times for high service levels. Figure 4 depicts these performance measures. In words, K-S statistic measures the maximum absolute difference between the cumulative probability distribution obtained by simulation and the predicted cumulative probability distribution. We also compare the absolute percentage error between the lead times that are set for a given service level obtained by simulation and the predicted cycle time distribution.

In Section 6.4, we present a case study to determine the minimum number of machines in a workstation and the sequencing rule to achieve a desired cycle time performance in a manufacturing system.

6.1. Multi-route network

To assess the performance of SLQNA in a setup where various building blocks are present and the parts can travel through various routes, we use the set up given in Figure 5. Table 7 gives the parameter sets used for these experiments.

Figure 6 gives the performance of the SLQNA algorithm in approximating the characteristics of the flows in various parts of the system and the moments of the total cycle time distribution. As the figure shows, SLQNA predicts the inter-departure time coefficient of variation and first-lag autocorrelation and also the mean and coefficient of variation of the cycle time very

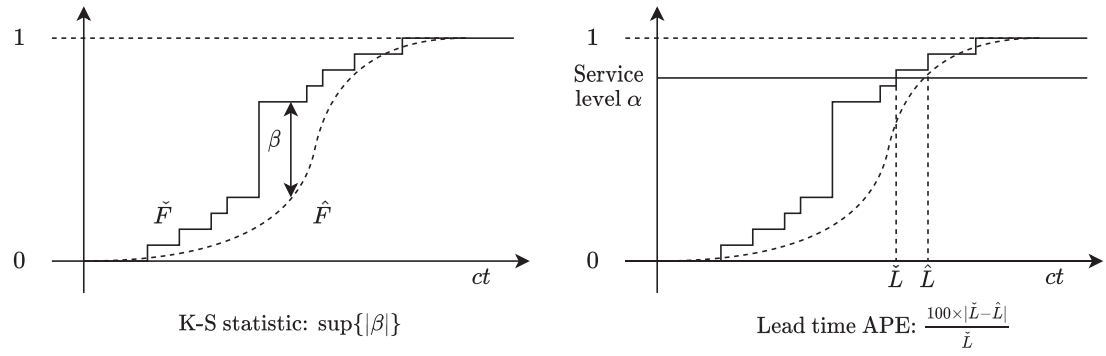

Figure 4. The performance measures used for assessing the fitted distributions for the total cycle time.

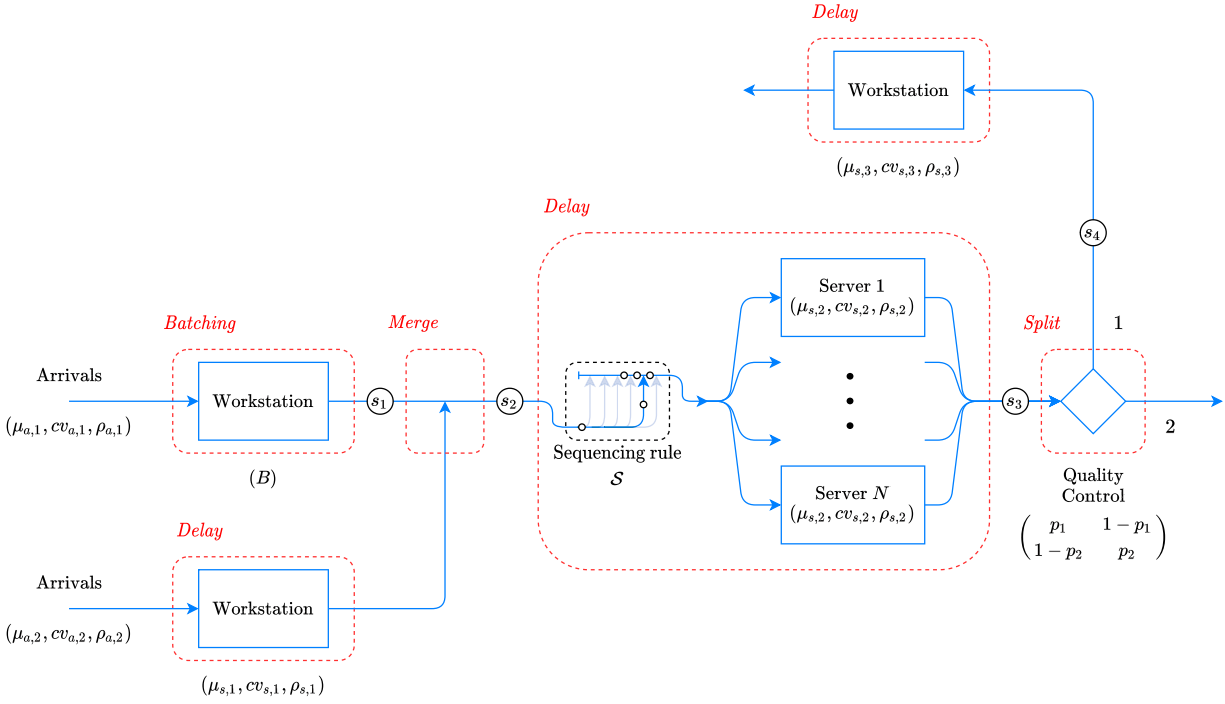

Figure 5. The multi-route network experimental set up.

Table 7. Range of parameters used for the multi-route network experiments.

Parameter	Range
$\mathcal{S}$, Sequencing rule	{FCFS, SIRO, LCFS, SPT, LPT}
N, # of servers	{1, 2, 5, 10}
$cv_{a,1}$	{0.1, 1, 1.4}
$\rho_{a,1}$	{−0.4, 0, 0.4}
B	{2, 10, 20}
p_1	{0.1, 0.5, 0.9}
p_2	{0.5}
$\mu_{a,1}$	$\{5/B\}$
$\mu_{a,2}$	{5}
$\mu_{s,1}$, $\mu_{s,2}$	{1}
$\mu_{s,3}$	{2}
$cv_{a,2}$,$cv_{s,1}$,$cv_{s,2}$,$cv_{s,3}$,	{1}
$\rho_{a,2}$	{0.4}
$\rho_{s,1}$, $\rho_{s,2}$, $\rho_{s,3}$	{0}

accurately. Figure 7 gives a few examples of the total cycle time distributions predicted based on the two-moment approximation (`2MomFitGamma`). Figure 8 gives the K-S statistic for these distributions in addition to the accuracy of SLQNA in predicting the lead times that would satisfy 95% and 98% service levels. These results show that SLQNA algorithm yields very accurate predictions of the cycle time distribution and output characteristics at different blocks of this multi-route network.

6.2. Serial line with homogeneous stations

In this set of experiments, we use a serial line with up to 10 homogeneous stations operating with up to 10 parallel servers at each station operating with different sequencing rules to assess the performance of SLQNA in predicting the mean and the coefficient of variation of the

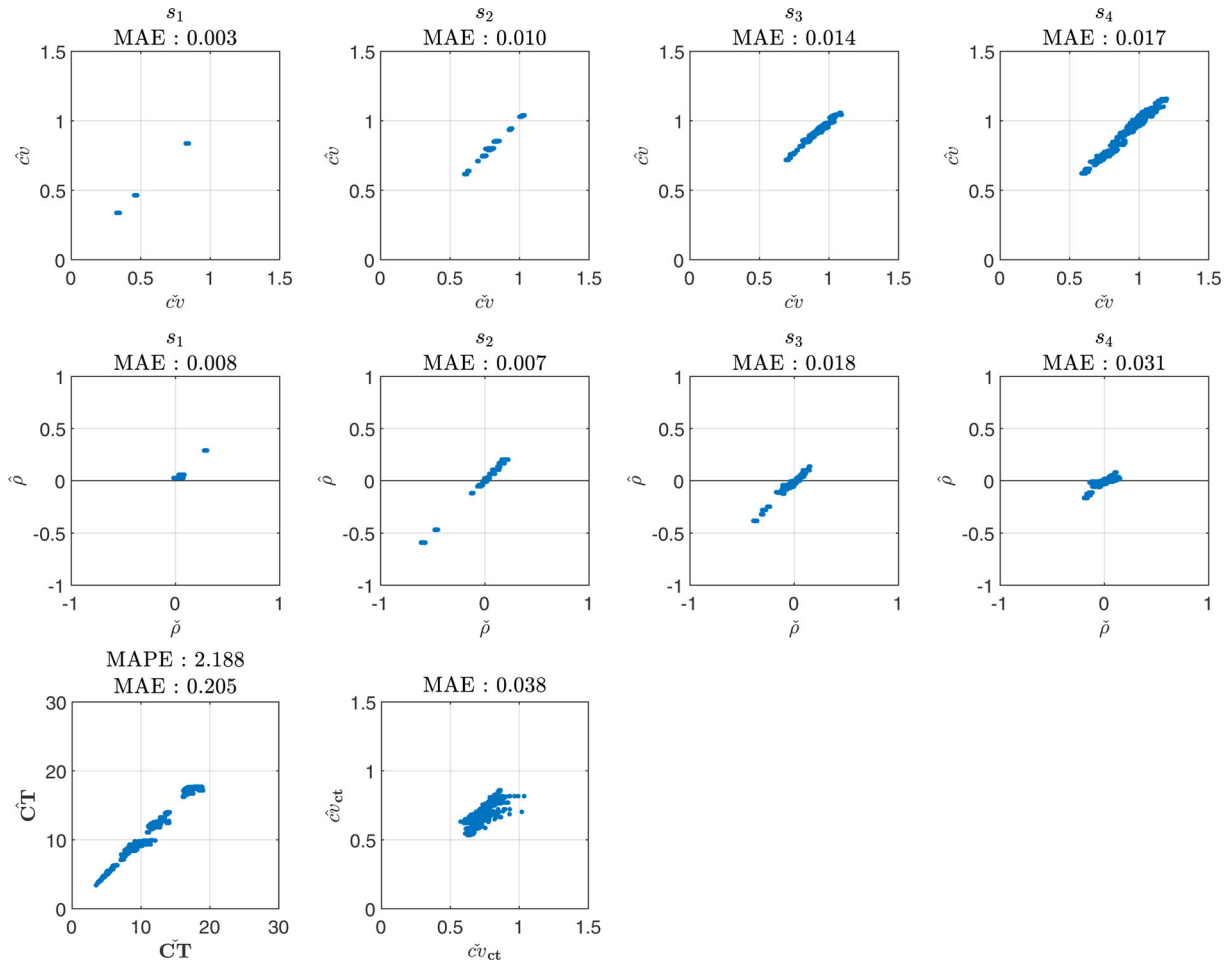

Figure 6. Accuracy of SLQNA for predicting the characteristics of the flows in the system and the moments of the total cycle time distribution for the multi-route network.

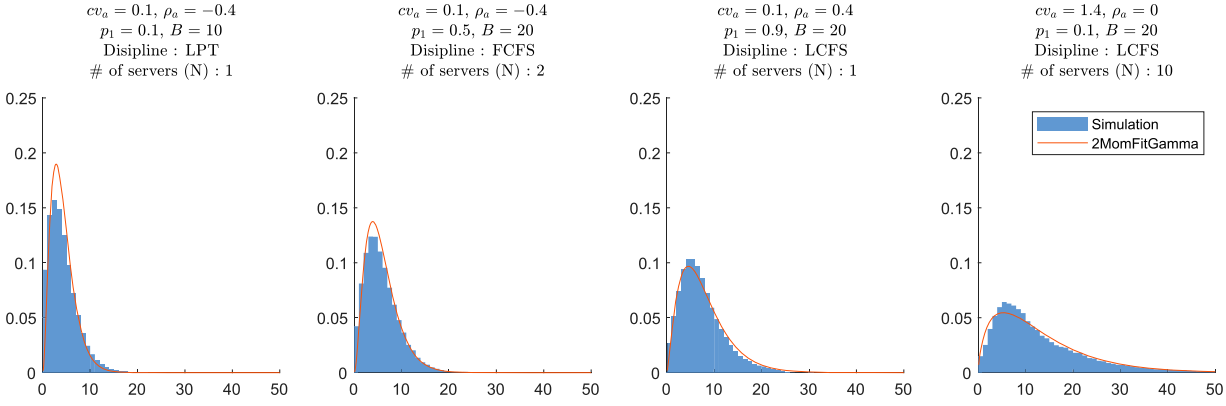

Figure 7. Accuracy of (`2MomFitGamma`) method for predicting the cycle time distribution in the multi-route network for four different cases.

total cycle time of a system in addition to the distribution of the cycle times. Figure 9 depicts this experimental setup. Table 8 gives the parameters used in these experiments. Figure 10 gives the performance of SLQNA in predicting the mean and the coefficient of variation of the total cycle time for the serial line. The SLQNA algorithm predicts the average cycle time in serial line experiments with a MAPE of 4% on average and the coefficient of variation of the cycle time with a MAE of 0.03. Figure 11 gives four specific examples of using `2MomFitGamma` approximation for predicting the distribution of the cycle times. Figure 4 depicts the performance measures we

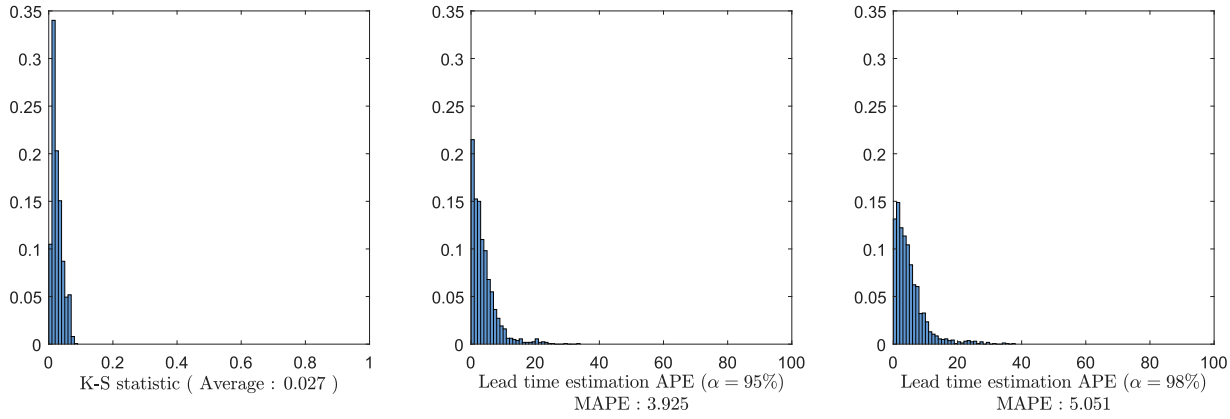

Figure 8. The performance measures used for assessing the fitted distributions obtained by (`2MomFitGamma`) for the total cycle time in the multi-route network experiments.

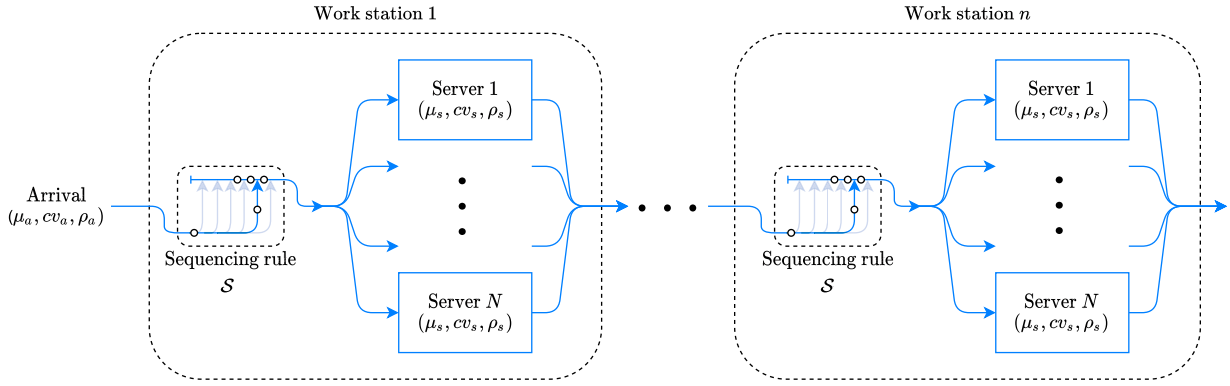

Figure 9. The experimental setup for production line experiments with homogeneous stations.

Table 8. Range of parameters used for production line experiments with homogeneous stations.

Parameter	Range
μ_a	{1/0.2, 1/0.5, 1/0.8}
μ_s	{1}
cv_a, cv_s	{0.2, 0.7, 1.2}
ρ_a, ρ_s	{−0.4, 0, 0.4}
n,Number of stations	{2, 5, 10}
$\mathcal{S}$,Sequencing rule	{FCFS, SIRO, LCFS, SPT, LPT}
N, # of servers	{1, 2, 5, 10}

use for assessing the accuracy of the predicted distributions. Figures 12 and 13 show the accuracy of these predictions using the performance measures related to the overall shape of the distribution (K-S statistic) and the lead time for given service levels. These results show that the SLQNA algorithm allows predicting the cycle time distribution accurately and setting the lead time for a given service level accordingly. Table 9 gives the time performance of the SLQNA algorithm for these experiments. As the table shows the runtime of the SLQNA algorithm is shorter than 1 s on a notebook computer. The number of servers in each block does not increase the computational time requirement of SLQNA and a 10-station with 10 servers at each station can be evaluated to determine the cycle time distribution and other performance measures in 0.376 s.

6.2.1. Comparing the different methods for predicting the distribution of cycle times in a heterogeneous serial line

To compare the three different methods for predicting the cycle times distribution given in Section 5.2, we use a serial line with heterogeneous stations. The cycle time distribution in a short serial line with heterogenous stations can have a different shape compared to the more regular distribution observed in a serial line with homogenous stations. In addition, low variability in the service times can result in cycle time distributions that are represented better using the quantiles. In this case, the approximation method that uses the quantiles (`QuanConv`) is expected to predict the total cycle time distribution more accurately.

In this set of experiments, the line has two single servers that operate under FCFS with processing rates varied independently in the set {0.5, 1}, and identical coefficient of variation varied in the set {0.2, 0.5, 0.7} and $\rho_s = 0$. The arrival rate to this system is set as $\lambda_a = 0.2$ with $cv_a = 1$ and ρ_a varied in the set {−0.4, 0, 0.4}.

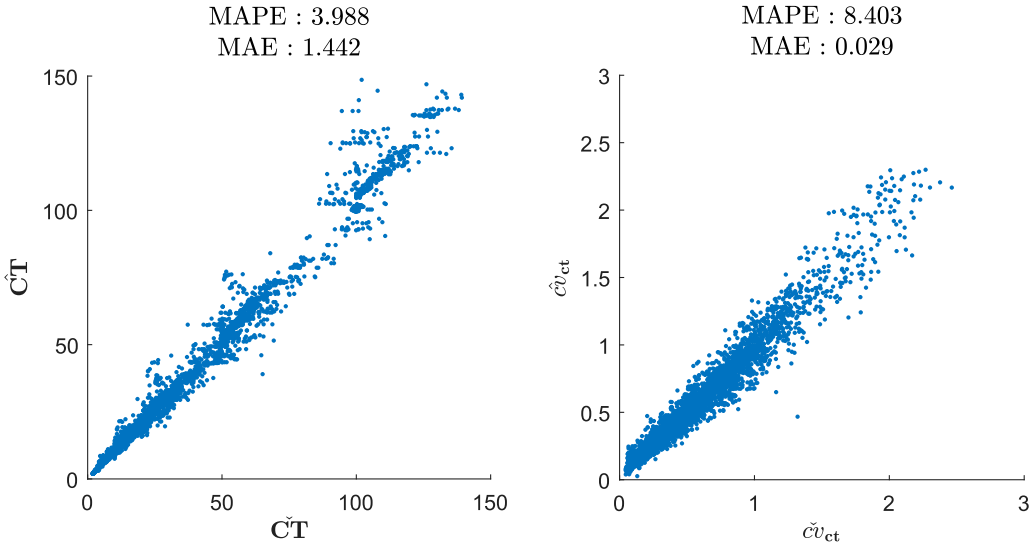

Figure 10. The accuracy of SLQNA for predicting the total cycle time and its coefficient of variation for the homogeneous serial line experiments.

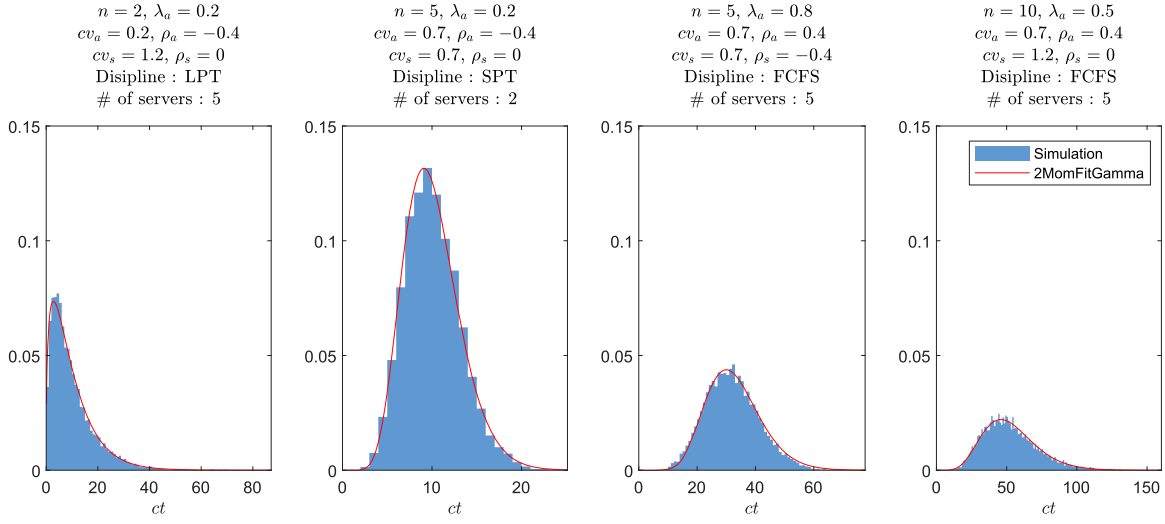

Figure 11. The accuracy of `2MomFitGamma` method for predicting the cycle time distribution in the homogeneous serial line experiments for four different cases.

Based on these parameter sets, 36 cases were examined using this setting. Figure 14 gives the performance of the three methods of predicting the cycle time distribution for these cases. As expected, the quantile approximation yields more accurate results compared to the other two methods in this specific case. Figure 15 gives a specific case with the fitted distributions obtained by using the different cycle time distribution prediction methods for a three-station line.

6.3. Effect of service disciplines on the cycle time distribution and the departure process

The previous results show that SLQNA can be used to predict the cycle time distribution and the departure process very accurately. In this section, we use SLQNA to investigate the effect of service disciplines on the cycle time distribution and the departure process in a queueing network with different blocks and correlated arrival and service times. We use the single-server Delay block with the parameter sets given in Table 4 for these experiments. We also present the results for a homogeneous serial line.

6.3.1. Effect of service disciplines on the cycle time distribution

Many factors affect the distribution of the cycle times in a queueing system. Among them, the service discipline has been known to be critical. There are many service disciplines that have been extensively studied and new service disciplines are being designed. The service disciplines that only consider the time that has been passed

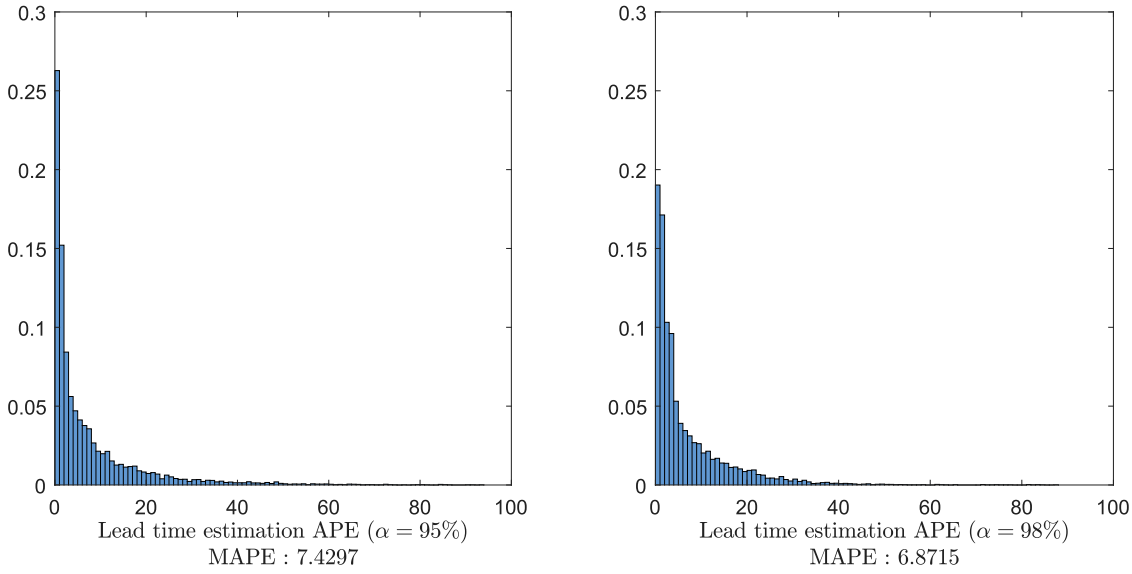

Figure 12. The accuracy of `2MomFitGamma` method for predicting the lead time for a given service level for the homogeneous serial line experiments.

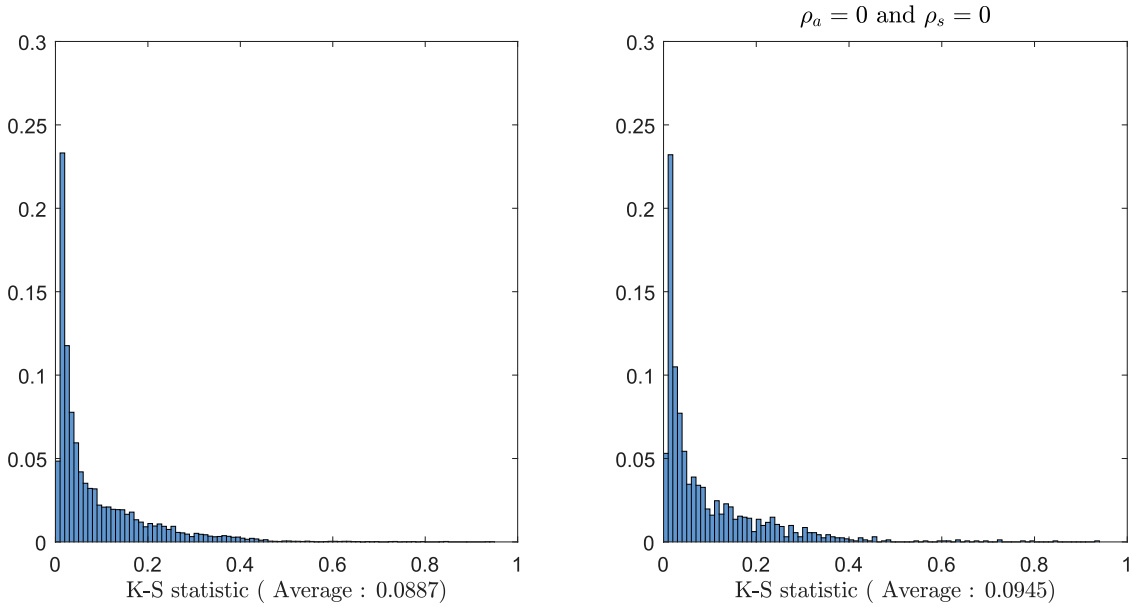

Figure 13. The K-S statistics for `2MomFitGamma` method for predicting the cycle time distribution for the homogeneous serial line experiments.

since a part has arrived at a queue do not influence the cycle times on average. However, they can have a considerable effect on the variability of the cycle times and other performance measures such as service level that depend on the cycle time distribution.

Figure 16 depicts the coefficient of variation of the cycle times when the service discipline is changed from FCFS to LCLS or SIRO in the single-server Delay block over the parameter set given in Table 4. The SIRO discipline has slightly larger cv_{CT} values compared to FCFS, but the LCFS discipline can create very large variability in the cycle times. The SPT and the LPT disciplines are well-known disciplines that make use of the processing time of the parts to decide on which part will be processed first. These disciplines have a considerable effect

Table 9. Runtime performance of the SLQNA algorithm for the homogeneous serial line experiments with n stations and N servers at each station in seconds.

		N			
		1	2	5	10
n	2	0.134	0.076	0.076	0.077
	5	0.327	0.190	0.189	0.190
	10	0.627	0.376	0.375	0.376

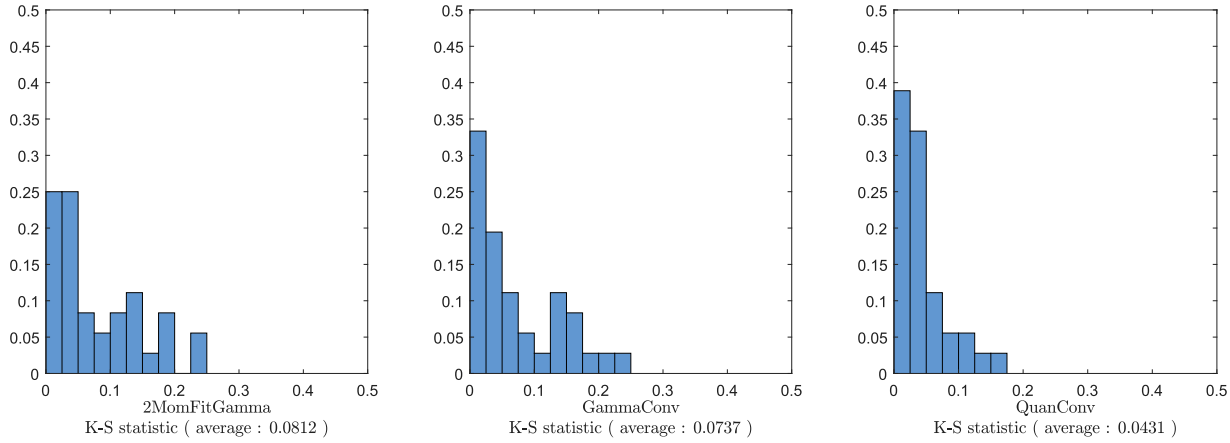

Figure 14. The performance of the different methods of predicting the cycle time distribution for the heterogeneous serial line experiments.

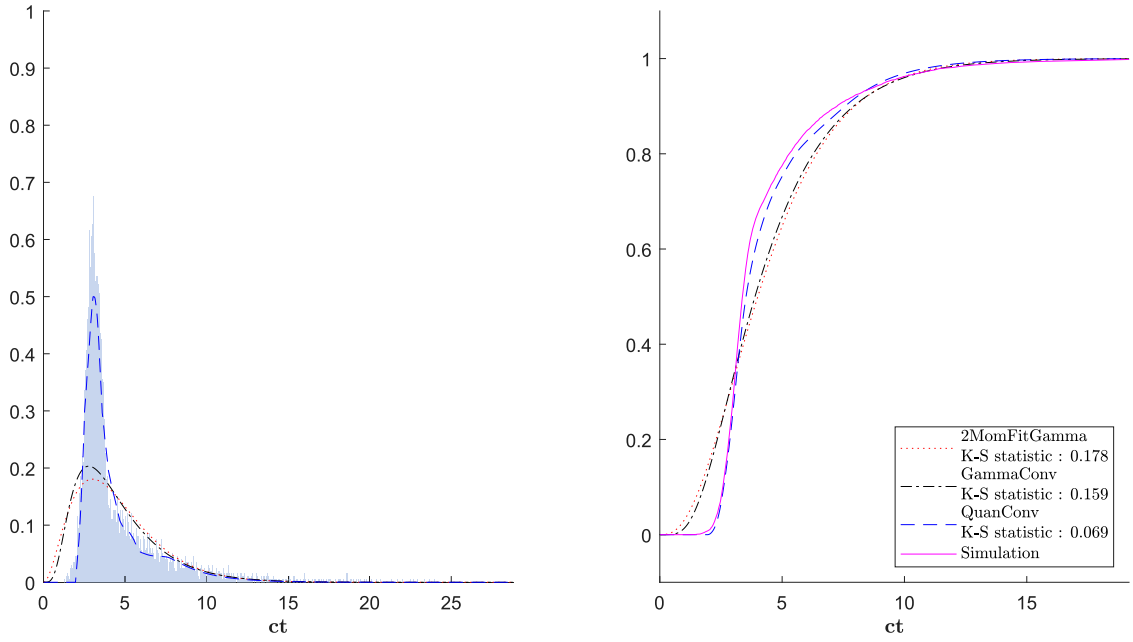

Figure 15. The performance of the different methods of predicting the cycle time distribution in a specific heterogeneous serial line experiment ($\rho_a = 0.4$, $\mu_{s,1} = 0.5$, $\mu_{s,2} = 1$, $cv_{s,1} = 0.2$, $cv_{s,2} = 0.2$).

on the cycle times. Namely, SPT can decrease the average total cycle time to a great extent. Additionally, SPT has been proven the best discipline for decreasing the average cycle times in many settings. However, both of these disciplines increase the variability of the cycle times that might be detrimental to some performance measures such as the shortest lead time for a high service level. Figure 17 depicts the effect of changing the service discipline from FCFS to SPT on the average cycle time and the coefficient of variation of the cycle time. These results show that the distribution of the cycle time, rather than only the average performance, should be considered to decide on using a given sequencing rule.

6.3.2. Selecting the service discipline that yields the shortest lead time for a given service level

The efficiency of the SPT sequencing rule in decreasing the average waiting times in queues has been widely studied. However, using the SPT sequencing rule increases the variability of the cycle times as depicted in Figure 17. The increase in variability of the cycle time may overshadow the benefits of a decrease in the average cycle times when setting a lead time for a given service level, denoted by LT_j for service discipline j.

Figure 18 depicts a specific homogeneous serial line with relatively heavy traffic. In this specific system, the lead time that needs to be set to achieve a 98% service

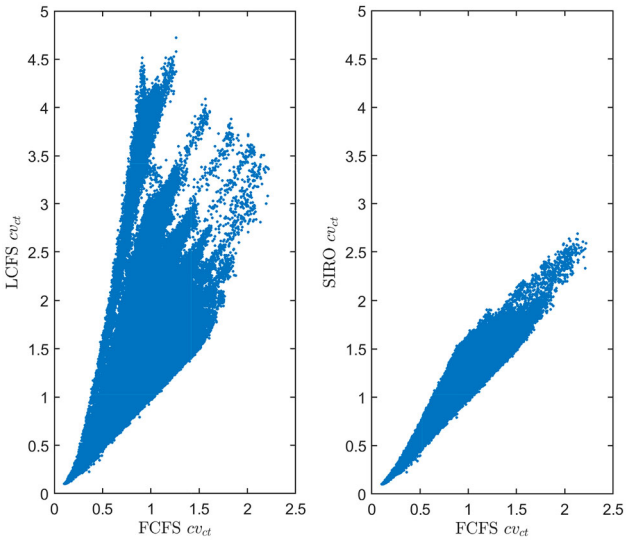

Figure 16. The effect of the sequencing discipline on the variability of the cycle times for FCFS, LCFS, and SIRO service disciplines for the single-server Delay block.

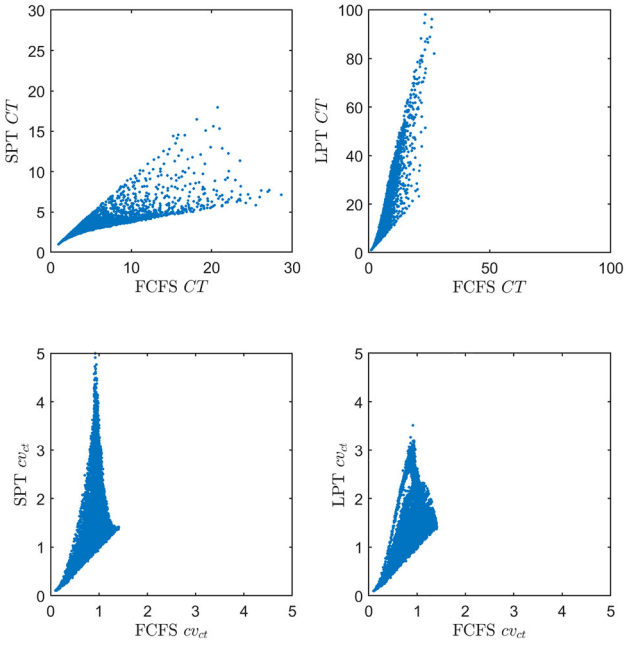

Figure 17. The effect of the SPT and LPT service disciplines on the cycle times and their variability for the single-server Delay block.

is shorter when FCFS is used as opposed to SPT, i.e. $LT_{\text{FCFS}} < LT_{\text{SPT}}$. In this system, since the utilization is 0.8, the rapid arrival of new parts might cause a part with a relatively large processing time to experience long waiting times as there is always a part with a smaller processing time present. These large waiting times influence the performance of the system when the performance measure such as the lead time for a given service level is affected by the shape of the tail of the cycle times distribution.

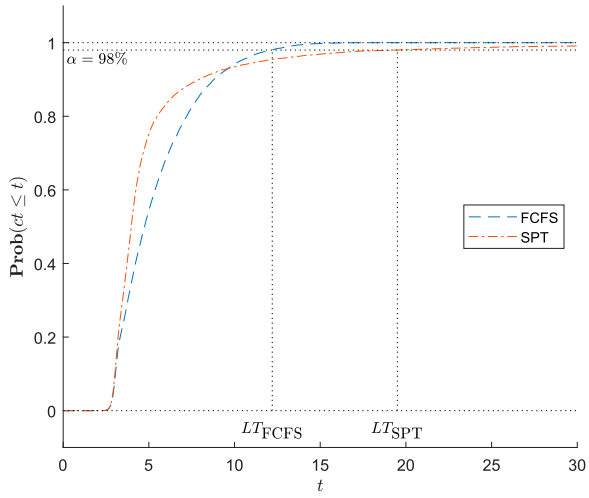

Figure 18. The cumulative probability distribution for the total cycle time in a homogeneous serial line where the lead time that must be set to satisfy a 98% service level is shorter when FCFS is used instead of SPT ($n = 3, \lambda_a = 0.8, cv_a = 1, \rho_a = 0, \mu_s = 1, cv_s = 0.1, \rho_s = 0$).

These experiments show that the SLQNA algorithm that predicts the cycle time distribution accurately can be used to investigate the effects of sequencing rules on the cycle time variability and also to select the right sequencing rule based on a performance measure that depends on the cycle time distribution rather than only its mean.

6.3.3. Effect of service disciplines on the departure process

An effect of using the SPT and the LPT service disciplines is that they can yield strongly correlated streams even when their inputs are not correlated. Figure 19 depicts the first-lag autocorrelation for different utilization levels for SPT. This effect can be more pronounced when the utilization is larger. This can be attributed to the service discipline having more options to choose from in such systems. The autocorrelation of the output stream of these blocks is highly dependent on the variability of the input stream and the service times.

When a sequencing rule generates a positive autocorrelation, the magnitude of the autocorrelation increases with an increase in the utilization and an increase in the service times coefficient of variation. This can be attributed to the fact that in low utilization systems, the sequencing rule is deciding between fewer parts available in the buffer and for this reason it less frequently changes the order of the parts being processed. Similarly, a service time process with low coefficient of variation yields service times that are closer to each other and a change in the order of parts being processed has less effect on the consecutive process times on the server.

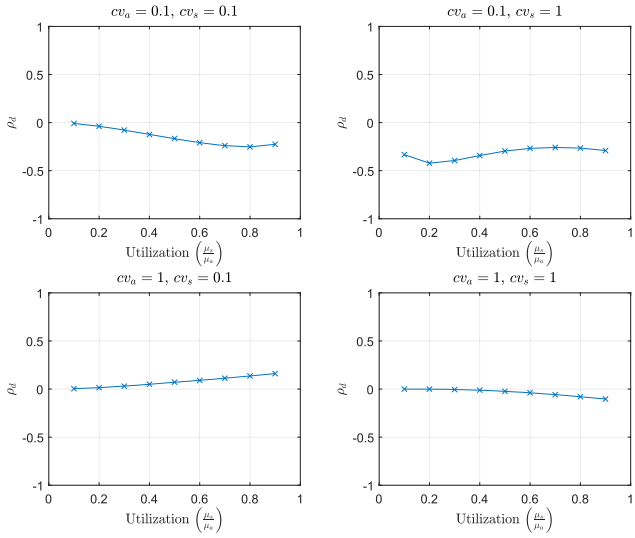

Figure 19. The output autocorrelation function of the single-server SPT block for $\rho_a, \rho_s = 0$.

6.4. Optimizing the resource allocation and the selection of sequencing rules subject to cycle time performance criteria in a serial line

In this section, we present a case study motivated by a decision problem observed in manufacturing systems. The decision problem we analyze is allocation of resources and the selection of sequencing rules in a production system in order to achieve a desired cycle time performance.

For assessing the performance of the SLQNA algori thm as an approximation method that can be used in optimizing queueing networks, we consider a serial line with three workstations. Each workstation has a number of parallel machines and operates with different sequencing rules. The goal is assigning the minimum number of machines and selecting a sequencing rule in Workstation 2 in order to meet the criteria related to the cycle time. Figure 20 depicts this system and Table 10 gives the sets for the parameters.

We consider the probability of meeting the desired lead time LT_d with a given probability as the cycle time criterion in this optimization problem. The optimization problem is described in Equation (18a).

Table 10. Parameters used for resource allocation and selection of sequencing rules experiment.

Parameter	Value/Decision variable domain
α	0.95
μ_a	1/2.5
μ_{s1}	1
μ_{s2}	1
μ_{s2}	1/2
$cv_a, cv_{s1}, cv_{s2}, cv_{s3}$	1
ρ_a	0.4
ρ_s	0
n, Number of stations	3
$\mathcal{S}_1$, Sequencingrule	FCFS
$\mathcal{S}_2$, Sequencingrule	{FCFS, SIRO, LCFS, SPT, LPT}
$\mathcal{S}_3$, Sequencingrule	SIRO
N_1, #ofservers	5
N_2, #ofservers	{3, . . . , 10}
N_3, #ofservers	2

$$\min_{\mathcal{S}_2, N_2} \quad N_2 \tag{18a}$$

$$\text{s.t.} \quad \text{prob}\,(\boldsymbol{ct} \leq \mathrm{LT}_d) \geq \alpha \tag{18b}$$

$$N_2 \in \{3, \ldots, 10\} \tag{18c}$$

$$\mathcal{S}_2 \in \{\text{FCFS, SIRO, LCFS, SPT, LPT}\} \tag{18d}$$

We also consider the problem of minimizing the number of machines in Workstation 2 to achieve the desired expected cycle time CT_d as an alternative cycle time performance criterion. For this case, Equation (18b) in above formulation is replaced with $\mathbf{CT} \leq \mathrm{CT}_d$.

In order to compare the solution obtained by using SLQNA with the optimal solution for this problem, we use simulation for total enumeration of the solution space using a computer cluster. The simulations in the cluster took one hour to determine the optimal solution. These simulations would have taken 8 h on a personal computer. On the other hand, SLQNA algorithm yields the optimal solution in less than 14 s.

Figure 21 gives the accuracy of SLQNA in determining the minimum N_2 value for different desired lead times LT_d with a service level of 95% and for different

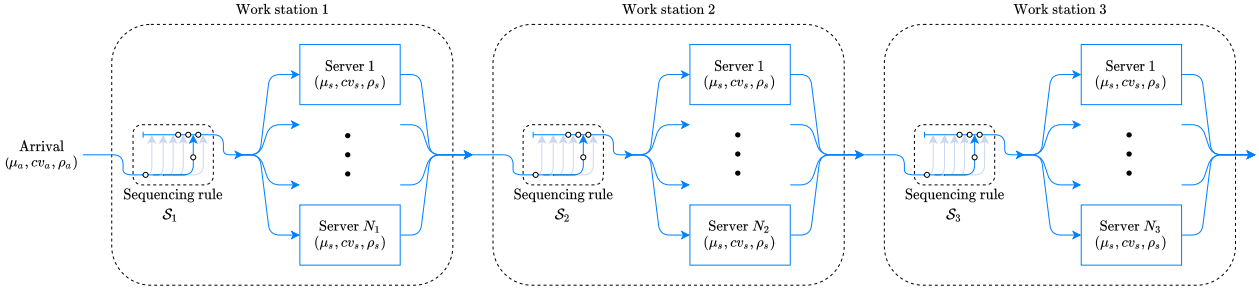

Figure 20. The experimental setup for the resource allocation and selection of sequencing rules.

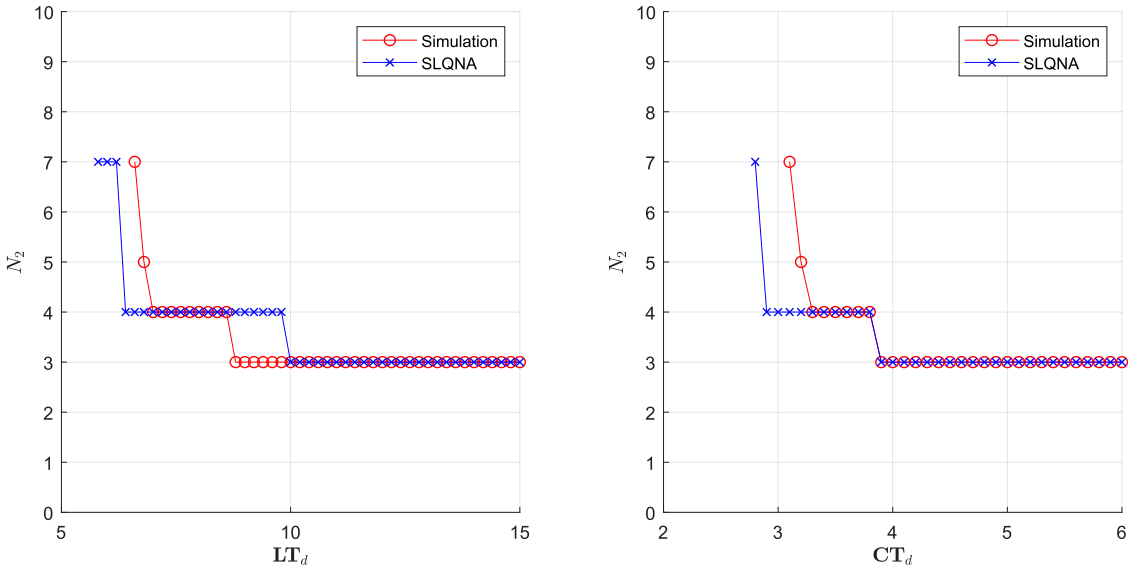

Figure 21. The accuracy of SLQNA in identifying the minimum number of machines to be used (N_2) according to different desired lead time LT_d and the desired average cycle time CT_d values.

desired average cycle times CT_d for this system. Excluding 5 data points, corresponding to the desired lead times of 5.8−6.6 and the desired average cycle time values of 2.8−3.1 among 75 different values LT_d and 60 different CT_d values, the solution obtained by SLQNA is either identical or suggests one additional machine compared to the optimal solution. There are only four values associated with very short LT_d and CT_d times where the solution suggested by SLQNA algorithm does not yield the desired service level due to the approximation error of the cycle time distribution. The results show that SLQNA algorithm can be used effectively to determine the resource allocation for the desired cycle time performance problem.

7. Conclusions

In this work, we present a supervised-learning-based approximation method (SLQNA) to determine the cycle time distribution and the mean, the coefficient of variation, and the first-lag autocorrelation of the inter-departure times in a queueing network that includes multiple servers at each block operating with FCFS, SIRO, LCFS, SPT and LPT sequencing rules with correlated interarrival and service times, batching, merge, and split building blocks separated with buffers with infinite capacity. Our results show that SLQNA is fast and very accurate. Having the capability to analyze systems with different sequencing rules and correlated interarrival and service times makes SLQNA a useful tool to analyze complex manufacturing systems such as semiconductor manufacturing plants.

The method we propose is based on training a supervised learning method with simulation data for each building block in isolation and then using the predictions obtained from the trained building blocks in a decomposition algorithm to analyze a given network. The training is done offline only once and the library of trained models for different delay, batching, split and merge blocks are combined for a given network structure. This approach allows analysis of a given network without simulating and training a new model. As a result, SLQNA combines the generality of simulation with the efficiency of analytical approximation methods and yield accurate results that could only be obtained by a very long simulation of a given system.

Predicting the distribution of the total cycle time in a queueing network in addition to its mean value allows setting a lead time for a given service level accurately and also determining a sequencing rule that allows a shorter lead time.

The approach presented in this study can be extended in different directions. In order to model a wider range of production systems, the SLQNA algorithm can be extended to systems with finite buffers and reentrant loops. Finite buffers and reentrant loops make the output characteristics of downstream blocks affect a given block. As a result, blocks cannot be analyzed in isolation and then combined with the decomposition algorithm presented in this study. Therefore, modeling networks with finite buffers and reentrant loops requires developing a different decomposition algorithm.

Another research direction is extending the sequencing rules to include priority rules. Many production systems use priority classes for different production types as

a sequencing rule. Considering various priority classes can generate block designs that are too large for generation of training data. This challenge can be overcome by using sampling methods and ensemble learning methods.

The method we present can also be extended to analyze queueing network models of production systems where production and material flow are controlled by using information about the state of the system. For example, analyzing systems where the control decisions are based on the inventory positions and other information related to the downstream buffers or the whole system requires a different modeling approach for the blocks.

Finally, the collected arrival and departure data from a network can be used to fit a building block that generates a departure stream that is statistically similar to the collected departure data. Data-driven modeling combined with the SLQNA algorithm allows developing a versatile, accurate and efficient data-driven modeling and analysis tool. These are left for future work.

As a summary, we propose SLQNA as an accurate and fast approximation method to predict the departure processes and the cycle time distribution in open queueing networks that are built by combining parallel stations that work under different service characteristics and correlated interarrival and service, and with batching, merge, and split blocks. This tool with its ability to analyze systems with different service disciplines and correlated interarrival and service times can be used to analyze complex manufacturing systems.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

Research leading to these results has received funding from the EU ECSEL Joint Undertaking [grant number 737459] (project Productive4.0) and from TUBITAK [grant number 217M145].

Notes on contributors

Siamak Khayyati is a Postdoctoral Fellow at Koç University. He received a BS degree in Industrial Engineering from Sharif University of Technology and PhD degree in Industrial Engineering and Operations Management from Koç University. His research interests are in design and control of production systems and artificial intelligence applications in manufacturing.

Barış Tan is a Professor of Operations Management and Industrial Engineering at Koç University, Istanbul, Turkey. His areas of expertise are in design and control of production systems, supply chain management, and stochastic modeling. He received a BS degree in Electrical & Electronics Engineering from Bogazici University, and ME in Industrial and Systems Engineering, MSE in Manufacturing Systems, and PhD in Operations Research from the University of Florida.

ORCID

Siamak Khayyati http://orcid.org/0000-0002-1230-6715
Barış Tan http://orcid.org/0000-0002-2584-1020

References

Alkaff, A., M. N. Qomarudin, and S. E. Wiratno. 2020. "Matrix-Analytic Solutions in Production Lines Without Buffers." *Computers & Operations Research* 119: 104903.

Angelopoulos, A., E. T. Michailidis, N. Nomikos, P. Trakadas, A. Hatziefremidis, S. Voliotis, and T. Zahariadis. 2020. "Tackling Faults in The Industry 4.0 Era – A Survey of Machine-Learning Solutions and Key Aspects." *Sensors* 20 (1): 109.

Arinez, J. F., Q. Chang, R. X. Gao, C. Xu, and J. Zhang. 2020. "Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook." *Journal of Manufacturing Science and Engineering*142 (11): 110804.

Boulas, K., G. Dounias, and C. Papadopoulos. 2017. "Approximating Throughput of Small Production Lines Using Genetic Programming." In *Operational Research in Business and Economics*, 185–204. Springer.

Breiman, L. 2001. "Random Forests." *Machine Learning* 45 (1): 5–32.

Buzacott, J. A., and J. G. Shanthikumar. 1993. *Stochastic Models of Manufacturing Systems*. Englewood Cliffs, NJ: Prentice Hall.

Cadavid, J. P. U., S. Lamouri, B. Grabot, R. Pellerin, and A. Fortin. 2020. "Machine Learning Applied in Production Planning and Control: A State-of-the-Art in the Era of Industry 4.0." *Journal of Intelligent Manufacturing* 31: 1531–1558.

Can, B., and C. Heavey. 2012. "A Comparison of Genetic Programming and Artificial Neural Networks in Metamodeling of Discrete-Event Simulation Models." *Computers & Operations Research* 39 (2): 424–436.

Chen, J., Z. Jia, L. Huang, and Y. Dai. 2020. "Transient Performance Evaluation of Flexible Production Lines with Two Bernoulli Machines and Dedicated Buffers." In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 836–841. IEEE.

Dallery, Y., and S. B. Gershwin. 1992. "Manufacturing Flow Line Systems: A Review of Models and Analytical Results." *Queueing Systems* 12 (1-2): 3–94.

Diamantidis, A., J.-H. Lee, C. T. Papadopoulos, J. Li, and C. Heavey. 2020. "Performance Evaluation of Flow Lines with Non-Identical and Unreliable Parallel Machines and Finite Buffers." *International Journal of Production Research* 58 (13): 3881–3904.

Fan, R.-N., F.-Q. Ma, and Q.-L. Li. 2020. "Optimization Strategies for Dockless Bike Sharing Systems via Two Algorithms of Closed Queuing Networks." *Processes* 8 (3): 345.

Fowler, J. W., and O. Rose. 2004. "Grand Challenges in Modeling and Simulation of Complex Manufacturing Systems." *Simulation* 80 (9): 469–476.

Ghosh, S., and A. D. Banik. 2018. "Computing Conditional Sojourn Time of A Randomly Chosen Tagged Customer in a BMAP/MSP/1 Queue Under Random Order Service Discipline." *Annals of Operations Research* 261 (1-2): 185–206.

Govil, M. K., and M. C. Fu. 1999. "Queueing Theory in Manufacturing: A Survey." *Journal of Manufacturing Systems* 18 (3): 214–240.

Grosof, I., Z. Scully, and M. Harchol-Balter. 2018. "SRPT for Multiserver Systems." *Performance Evaluation* 127: 154–175.

Gurney, K. 1997. *An Introduction to Neural Networks*. London: CRC Press.

Harrison, J. M., and V. Nguyen. 1990. "The QNET Method for Two-Moment Analysis of Open Queueing Networks." *Queueing Systems* 6 (1): 1–32.

Harrison, P. 1984. "A Note on Cycle Times in Tree-like Queueing Networks." *Advances in Applied Probability* 16 (1): 216–219.

Hopp, W. J., and M. L. Spearman. 2011. *Factory Physics*. Boston: Waveland Press.

Horváth, A., G. Horváth, and M. Telek. 2010. "A Joint Moments Based Analysis of Networks of MAP/MAP/1 Queues." *Performance Evaluation* 67 (9): 759–778.

Jeong, K.-C., and Y.-D. Kim. 1999. "An Approximation Method for Performance Analysis of Assembly/Disassembly Systems with Parallel-Machine Stations." *IIE Transactions* 31 (4): 391–394.

Jiang, L., and R. E. Giachetti. 2008. "A Queueing Network Model to Analyze the Impact of Parallelization of Care on Patient Cycle Time." *Health Care Management Science* 11 (3): 248–261.

Kimura, T. 1994. "Approximations for Multi-Server Queues: System Interpolations." *Queueing Systems*17 (3-4): 347–382.

Kuehn, P. 1979. "Approximate Analysis of General Queuing Networks by Decomposition." *IEEE Transactions on Communications* 27 (1): 113–126.

Kumar, S., and P. Kumar. 2001. "Queueing Network Models in the Design and Analysis of Semiconductor Wafer Fabs." *IEEE Transactions on Robotics and Automation* 17 (5): 548–561.

Kuo, Y.-H., and A. Kusiak. 2019. "From Data to Big Data in Production Research: The Past and Future Trends." *International Journal of Production Research* 57 (15-16): 4828–4853.

Liu, M., S. Fang, H. Dong, and C. Xu. 2021. "Review of Digital Twin About Concepts, Technologies, and Industrial Applications." *Journal of Manufacturing Systems* 58: 346–361.

Manafzadeh Dizbin, N. 2020. "On Performance Evaluation and Optimal Control of Manufacturing Systems." PhD diss., Koç University, Istanbul, Turkey.

Manafzadeh Dizbin, N., and B. Tan. 2019. "Modelling and Analysis of The Impact of Correlated Inter-Event Data on Production Control Using Markovian Arrival Processes." *Flexible Services and Manufacturing Journal* 31 (4): 1042–1076.

Mönch, L., J. W. Fowler, and S. J. Mason. 2012. *Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analysis, and Systems*. Vol. 52. Heidelberg: Springer.

Nitz, M., D. Smith, B. Wysocki, D. Knoell, and T. Wysocki. 2021. "Modeling of an Immune Response: Queuing Network Analysis of the Impact of Zinc and Cadmium on Macrophage Activation." *Biotechnology and Bioengineering* 118 (1): 412–422.

Patchong, A., and D. Willaeys. 2001. "Modeling and Analysis of an Unreliable Flow Line Composed of Parallel-Machine Stages." *IIE Transactions* 33 (7): 559–568.

Priore, P., D. De La Fuente, A. Gomez, and J. Puente. 2001. "A Review of Machine Learning in Dynamic Scheduling of Flexible Manufacturing Systems." *Artificial Intelligence for Engineering Design Analysis and Manufacturing* 15 (3): 251–263.

Quinonero-Candela, J., and C. E. Rasmussen. 2005. "A Unifying View of Sparse Approximate Gaussian Process Regression." *Journal of Machine Learning Research* 6: 1939–1959.

Rasmussen, C. E., and C. K. Williams. 2006. *Gaussian Processes for Machine Learning*. Cambridge: MIT Press.

Sakasegawa, H. 1977. "An Approximation Formula $L_q \simeq \alpha.\rho^{\beta}/(1-\rho)$." *Annals of the Institute of Statistical Mathematics*29 (1): 67–75.

Schömig, A. K., and M. Mittler. 1995. "Autocorrelation of Cycle Times in Semiconductor Manufacturing Systems." In *Proceedings of the 27th Conference on Winter Simulation*, 865–872. IEEE Computer Society.

Schrage, L. E., and L. W. Miller. 1966. "The Queue M/G/1 with the Shortest Remaining Processing Time Discipline." *Operations Research* 14 (4): 670–684.

Shanthikumar, J. G., S. Ding, and M. T. Zhang. 2007. "Queueing Theory for Semiconductor Manufacturing Systems: A Survey and Open Problems." *IEEE Transactions on Automation Science and Engineering* 4 (4): 513–522.

Shin, J., D. Grosbard, J. R. Morrison, and A. Kalir. 2019. "Decomposition Without Aggregation for Performance Approximation in Queueing Network Models of Semiconductor Manufacturing." *International Journal of Production Research* 57 (22): 7032–7045.

Shin, Y. W., and D. H. Moon. 2021. "A Unified Approach for an Approximation of Tandem Queues with Failures and Blocking Under Several Types of Service-Failure Interactions." *Computers & Operations Research* 127: 105161.

Takagi, H. 1996. "A Note on the Response Time in M/G/1 Queues with Service in Random Order and Bernoulli Feedback." *Journal of the Operations Research Society of Japan* 39 (4): 486–500.

Tan, B., and S. Khayyati. 2021. "Supervised Learning-Based Approximation Method for Single-Server Open Queueing Networks with Correlated Interarrival and Service Times." *International Journal of Production Research*. doi:10.1080/00207543.2021.1887536.

Tan, B., and S. Lagershausen. 2017. "On the Output Dynamics of Production Systems Subject to Blocking." *IIE Transactions* 49 (3): 268–284.

Wang, M., H. Huang, and J. Li. 2021. "Transient Analysis of Multiproduct Bernoulli Serial Lines with Setups." *IEEE Transactions on Automation Science and Engineering* 18 (1): 135–150.

Wu, J.-S., and W. Chan. 1989. "Maximum Entropy Analysis of Multiple-Server Queueing Systems." *Journal of the Operational Research Society* 40 (9): 815–825.

Yang, X., and A. S. Alfa. 2009. "A Class of Multi-Server Queueing System with Server Failures." *Computers & Industrial Engineering* 56 (1): 33–43.

Zhang, H.-Y., Q.-X. Chen, J. M. Smith, N. Mao, A.-L. Yu, and Z.-T. Li. 2017. "Performance Analysis of Open General Queuing Networks with Blocking and Feedback." *International Journal of Production Research* 55 (19): 5760–5781.

Zhang, R., and M. Pavone. 2016. "Control of Robotic Mobility-on-Demand Systems: A Queueing-Theoretical Perspective." *International Journal of Robotics Research* 35 (1-3): 186–203.