International Journal of Production Research

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

Supervised learning-based approximation method for single-server open queueing networks with correlated interarrival and service times

Barış Tan & Siamak Khayyati

To cite this article: Barış Tan & Siamak Khayyati (2022) Supervised learning-based approximation method for single-server open queueing networks with correlated interarrival and service times, International Journal of Production Research, 60:22, 6822-6847, DOI: 10.1080/00207543.2021.1887536

To link to this article: https://doi.org/10.1080/00207543.2021.1887536

Published online: 20 Feb 2021.

INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH
2022, VOL. 60, NO. 22, 6822–6847
https://doi.org/10.1080/00207543.2021.1887536

Taylor & Francis
Taylor & Francis Group

Supervised learning-based approximation method for single-server open queueing networks with correlated interarrival and service times

Barış Tan and Siamak Khayyati

College of Administrative Sciences and Economics, College of Engineering, Koç University, Istanbul, Turkey

ABSTRACT

Efficient performance evaluation methods are needed to design and control production systems. We propose a method to analyse single-server open queueing network models of manufacturing systems composed of delay, batching, merge and split blocks with correlated interarrival and service times. Our method (SLQNA) is based on using a supervised learning approach to determine the mean, the coefficient of variation, and the first-lag autocorrelation of the inter-departure time process as functions of the mean, coefficient of variation and first-lag autocorrelations of the interarrival and service times for each block, and then using the predicted inter-departure time process as the input to the next block in the network. The training data for the supervised learning algorithm is obtained by simulating the systems for a wide range of parameters. Gaussian Process Regression is used as a supervised learning algorithm. The algorithm is trained once for each block. SLQNA does not require generating additional training data for each unique network. The results are compared with simulation and also with the approximations that are based on Markov Arrival Process modelling, robust queueing, and G/G/1 approximations. Our results show that SLQNA is flexible, computationally efficient, and significantly more accurate and faster compared to the other methods.

ARTICLE HISTORY
Received 9 December 2020
Accepted 1 February 2021

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

1. Introduction

Performance evaluation of manufacturing systems has been subject to numerous studies in the literature (Dallery and Gershwin 1992; Buzacott and Shanthikumar 1993; Papadopoulos and Heavey 1996). Analytical approximations and simulation have been used to predict the throughput, cycle time, buffer levels and other measures of interest.

While simulation methods can be used to model a given production system, the time to build the system with the desired level of detail and the time to obtain statistically significant results can be long. As a result, designing a production system that requires optimising many parameters using simulation requires substantial computational resources and time.

On the other hand, analytical approximations can be developed for a specific production system under more restrictive assumptions. Most of the analytical approximation methods are based on decomposing a given network into building blocks, analysing these building blocks to determine their output parameters, passing the output parameters to the computation of other building blocks, and then continuing this process repetitively according to an algorithm until a convergence criterion is met. Since this method requires computing the output characteristics of a building block many times, an analytical method is used to determine the output parameters efficiently under restrictive assumptions. Most of these methods also rely on Poisson arrivals, exponential and phase-type distributions to allow analytical tractability (Dallery and Gershwin 1992; Buzacott and Shanthikumar 1993) or use two-moment approximations to represent the arrival and service processes under the assumption that these processes do not exhibit any autocorrelation (Kuehn 1979; Buzacott and Shanthikumar 1993; Hopp and Spearman 2011). As a result, most of the analytical methods developed for the stochastic models of manufacturing systems do not incorporate possible autocorrelation of the interarrival and service times.

The interevent times observed in manufacturing systems exhibit significant autocorrelation. Figures 1(a,b) depict the (normalised) interarrival, service, and inter-departure time distributions and autocorrelation functions for a machine in Robert Bosch Reutlingen semiconductor plant. A study of close to 4.5 million interevent data from 363 machines at this plant yield that 58%

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

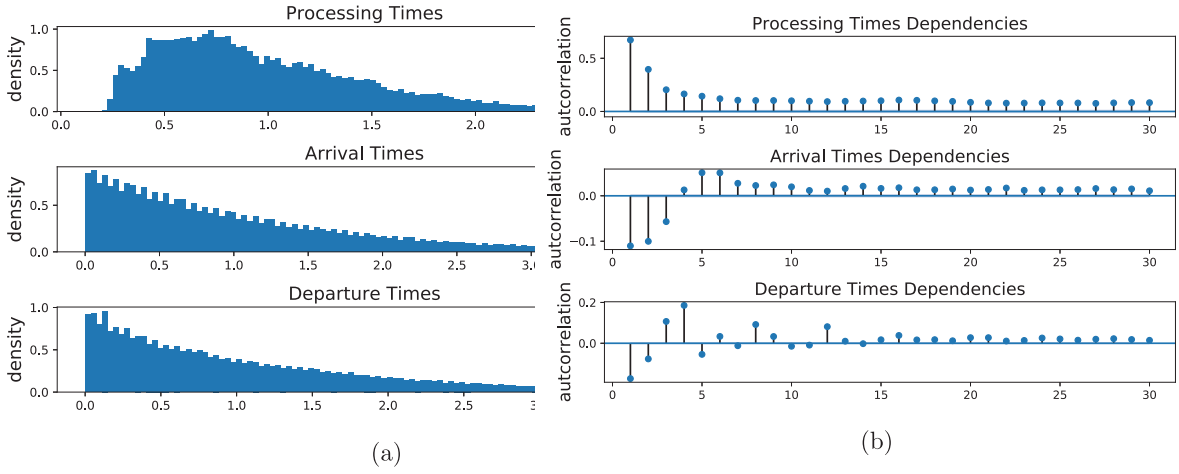

Figure 1. Empirical distribution. (a) and Autocorrelation (b) of interarrival, Service, and Departure Times of a Machine in Bosch Reutlingen semiconductor plant.

of the processing times have a first-lag autocorrelation greater than 0.25 and 18% of the observed interarrival times have a first-lag autocorrelation greater than 0.1 or less than −0.1 (Manafzadeh Dizbin 2020). Moreover, ignoring autocorrelation of interevent times yields significant errors in performance evaluation and control or production systems (Manafzadeh Dizbin and Tan 2019).

In this paper, we present a new approximation method to analyse open queueing networks with correlated interarrival and service times. The method is intended to combine the generality of simulation methods with the computational efficiency of analytical approximation methods. The approach presented in this study is based on generating training sets for the desired output variables for a wide range of input parameters for the delay, batching, merge, and split building blocks that are used to construct a queueing network model of a manufacturing system. Then a supervised learning algorithm trained with simulation is used to determine the functional relationship between the input stream characteristics and the output characteristics. For example, for the single-station delay block that represents a machine in a manufacturing system depicted in Figure 2, the input parameters are the mean, the coefficient of variation and the first-lag autocorrelations of the interarrival and service times, (μ_a, cv_a, ρ_a) and (μ_s, cv_s, ρ_s). The output parameters are the mean, the coefficient of variation, and the first-lag autocorrelation of the inter-departure time,

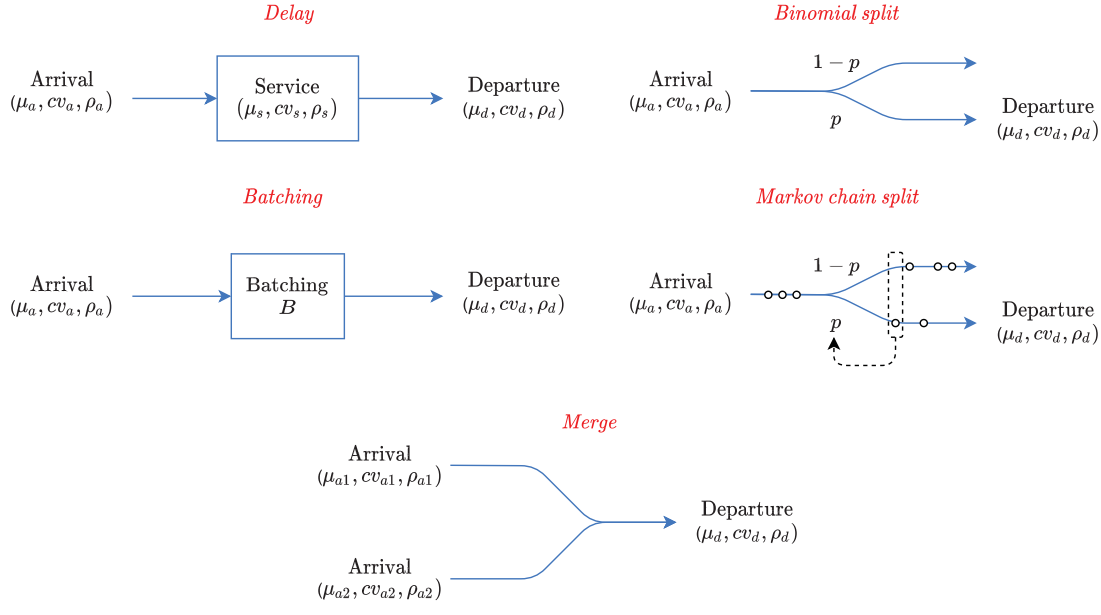

Figure 2. The building blocks of the single-server queuing network.

(μ_d, cv_d, ρ_d) and the mean and the coefficient of the cycle time, (CT, cv_{CT}). As the next step in the approximation method, the parameters of the output process are fed into the next block in an open queueing network as its input parameters. The queueing network models are built and analysed by combining the delay, batching, split, and merge building blocks. This method is referred as SLQNA (*S*upervised *L*earning based *Q*ueuing *N*etwork *A*nalyzer).

The main contribution of this study is presenting a computationally efficient and accurate approximation method for a single-server open queueing network that is composed of delay, batching, merge, and split building blocks with correlated interarrival and service times. Our extensive numerical experiments show that existing approximation methods that do not incorporate possible autocorrelation in interarrival and service times can yield significant errors in the cycle time prediction. Compared to the methods that incorporate correlated interarrival and service times based on Markov Arrival Process (MAP) representation of interarrival and service times, SLQNA is 50 times faster on average and yields an error that is 50% lower compared to this method. Furthermore, the computational performance of SLQNA is not affected by the size of the network and the parameters used as opposed to the limitation of the other methods.

The organisation of the remaining part of this paper is as follows. In Section 2, we review the pertinent literature. The approach to analyse the building blocks to construct the queueing networks is presented in Section 3. The supervised learning method to determine the output parameters, the method to generate correlated random variables, the training data and the accuracy of the output parameters obtained by using the supervised learning method are given in Section 4. The approximation method to analyse open queueing networks and the numerical experiments that compare the accuracy and the computational efficiency with the existing methods are presented in Section 5. Finally, the conclusions are given in Section 6.

2. Past work

In this section, we review the past work related to the analysis of queueing networks and using supervised learning methods in design of production systems.

2.1. Analysis of queueing networks with i.i.d. interarrival and i.i.d. service times

Most of the analytical approximation methods developed to analyse open queueing networks do not incorporate autocorrelation among interarrival and service times. In Queueing Network Analysis (QNA), approximations for the cycle time and the coefficient of variation of the inter-departure time are used to determine these values as functions of the mean and the coefficient of variation of the interarrival and service times. Since the departure rate is equal to the arrival rate in an open queueing network, the departure rate and the coefficient of variation of the departure process obtained from an analytical approximation are fed into the following block (Kuehn 1979; Buzacott and Shanthikumar 1993; Hopp and Spearman 2011). Other methods that model the buffer level processes as reflected Brownian motion are designed mainly to study systems with heavy traffic (Harrison and Nguyen 1990).

Exact analytical solutions that yield the output characteristics based on the interarrival and service time characteristics are available for only a few classes of queueing systems. For general systems, analytical approximations have been used. For a single station with general independent interarrival and service time distributions with independent interarrival and service times, i.e. $\rho_a = \rho_s = 0$, a widely known G/G/1 queue approximation for the average time in the queue is given by Kingman (1961):

$$CT_q = \frac{(cv_a^2 + cv_s^2)}{2} \frac{u}{1-u} \mu_s, \tag{1}$$

where $u = \mu_s/\mu_a$ is the utilisation and the cycle time is equal to the cycle time in the queue and the average service time, i.e. $CT = CT_q + \mu_s$.

Similarly, for the flow variability, an approximation for the inter-departure time coefficient of variation is given (Marshall 1968) as

$$cv_d^2 = u^2 cv_s^2 + (1-u^2) cv_a^2. \tag{2}$$

There are also other approximations for G/G/1 queue, e.g. Krämer and Langenbach-Belz (1976), Rasmussen and Williams (2006) and Buzacott and Shanthikumar (1993) among others. These approximations define the functional relationship between the inputs (μ_a, cv_a), (μ_s, cv_s), and the outputs (μ_d, cv_d) and CT and ignore the effects of ρ_a and ρ_s. The accuracy of these approximations even with the independence assumption of the interarrival and service times depends on the range of parameters used. The error introduced by these approximations can be significant in evaluating the performance of a manufacturing system (Akhavan-Tabatabaei, Ding, and Shanthikumar 2009).

In Section 4.4.1, we compare the accuracy of the approximations given by Kingman (1961), Krämer and Langenbach-Belz (1976), Rasmussen and Williams (2006) and Buzacott and Shanthikumar (1993) with the accuracy of the supervised learning algorithm presented in

this study for a station with correlated interarrival and service times. The results show that the average cycle time predicted by the delay block of SLQNA presented in this study is very accurate with a mean absolute percentage error (MAPE) of 1% while all the other approximations give an average error greater than 11%.

In Section 5.2, we compare the accuracy of SLQNA with QNA in predicting the total cycle time in different networks. This comparison shows that SLQNA is much more accurate than QNA (3% vs 11% MAPE for the range of parameters in the production line experiments and 4% vs. 8% MAPE in the experiments with the network with split and merge).

2.2. Analysis of queueing networks with correlated interarrival and i.i.d. service times

The number of studies that present approximation methods for the autocorrelated interevent times are limited. There are studies that use the renewal approximation of the autocorrelated arrivals for a system with correlated arrivals and renewal service times (Jagerman et al. 2004; Araghi and Balcıoğlu 2020). These studies do not provide an approximation for the departure process mean, coefficient of variation, and autocorrelation for the general correlated interevent times.

Whitt and You (2020) introduce a queuing network analysis method based on robust queuing and using the indices of dispersion for counts that is closely related to the autocorrelation function of a process. This method, referred as Rob-QNA in this study, assumes renewal service times and a first come first serve queuing discipline.

The comparison of the accuracy of SLQNA with Rob-QNA in predicting the total cycle time, given in Section 5.2, shows that SLQNA is much more accurate than Rob-QNA (3% vs 12% MAPE for the range of parameters in the production line experiments).

2.3. Analysis of queueing networks with correlated interarrival and correlated service times

Markovian Arrival Processes (MAP) can be used to capture autocorrelation in interevent times of manufacturing systems. A method that is based on analysing single-server queueing networks with correlated interevent times by using MAPs has been introduced by Horváth, Horváth, and Telek (2010). This method, referred as MAPQNA in this study, uses MAP representations of the flows between the stations and the service times for modelling networks of MAP/MAP/1 queues with split and merge (Horváth, Horváth, and Telek 2010).

There are two main limitations in this approach for performance evaluation of manufacturing systems. The range of the mean, coefficient of variations and autocorrelation of a given process that can be modelled by using a MAP can be limited, especially for negative autocorrelations that are present in manufacturing systems. Extending the range of parameters comes at the expense of making the state-space larger and yields computational problems. The second limitation is related to the effect of truncating the infinite-length MAP that represents a departure process when it is used as an input MAP for the next queue in the network. To minimise the errors introduced by the truncation, MAPQNA uses a moment matching procedure given by Telek and Horváth (2007).

The comparison of the accuracy of SLQNA with MAPQNA, given in Section 5.2, in predicting the total cycle time shows that SLQNA is much more accurate than MAPQNA (3% vs. 6% MAPE for the range of parameters in the production line experiments and 4% vs. 8% MAPE for the experiments with the network with split and merge). Furthermore, the average computational time for SLQNA is 1 second while the average computational time for MAPQNA is 50 s on a personal computer. Moreover, the computational performance of SLQNA is very robust while MAPQNA may not yield a result depending on the system parameters.

2.4. Analysis of queueing networks with simulation and supervised learning

Simulation is used widely in the industry to evaluate the performance of production systems. However, most of the simulation studies ignore correlated interarrival and service times. Setting up the simulation and running it to get statistically significant results can also take a long time for a given manufacturing system.

Simulation has been used for generating training data for flow time prediction. Hung and Chang (1999) and Yang (2010) uses neural networks for generating meta-models for simulation and discusses the process of choosing the parameter sets for generating the training data.

The models that are built by using these learning methods can be used as surrogate models for improving the performance of simulation-optimisation methods (Mihoubi, Bouzouia, and Gaham 2020). Simulation has been also used in tandem with analytical models to improve their performance and scope (Shanthikumar and Sargent 1983). Horng and Lin (2013) use an artificial neural network surrogate model to improve the performance of an optimisation procedure based on an evolutionary algorithm and ordinal optimisation.

There are a number of studies that use Genetic Programming to predict the throughput of a serial line (Can and Heavey 2012; Boulas, Dounias, and Papadopoulos 2017) with independent interarrival and service

times. The performance of neural networks and genetic programming has also been compared in three different manufacturing systems (Can and Heavey 2012). De Sousa Junior et al. (2020) consider a shop floor resource allocation problem and adopt a solution method based on a genetic algorithm where the offsprings in each generation can be evaluated using simulation with parallel computing or using a surrogate model. De Sousa Junior et al. (2020) give an extensive comparison of machine learning methods for this task and shows the relative advantage of using Gaussian Process Regression (GPR). For a more detailed review of the application of machine learning in various areas of manufacturing, the reader is referred to Arinez et al. (2020).

These examples show the potential of supervised learning methods in evaluating the performance of production systems. However, they consider a given system and the results cannot be used to analyse manufacturing systems in a different configuration of machines. Our objective is combining the generality of simulation models with the computational efficiency of analytical approximations by developing building blocks that give the output parameters for the given inputs for open queuing network models of manufacturing systems. By combining these building blocks in flexible ways and feeding the output of one block into the next one, open queueing networks with correlated interarrival and service times can be analysed. In this paper, we present our general approach and results for open single-server queueing networks with delay, batching, split, and merge building blocks. The results for queueing networks with parallel servers and different sequencing and dispatching rules will be given in a following study.

We contribute to the literature by introducing an approximation method for single-server open queueing network models of manufacturing systems that are composed of delay, batching, merge, and split blocks with correlated interarrival and service times. The batching and Markov Chain split blocks have not been included in other approximation methods. The building blocks are built by deriving the functional relationships that relate the input characteristics to the output characteristics by using GPR that uses the simulation results as the training data. This approach yields the desired performance measures more accurately and faster compared to the alternative methods given in the literature.

3. Building blocks for constructing queuing networks

In this study, the Delay, Batching, Binomial and Markov Chain Split, and Merge building blocks are combined to construct a queueing model of a manufacturing system. These blocks are depicted in Figure 2.

These building blocks are simulated with correlated interarrival and service time processes to obtain their output characteristics (the mean, coefficient of variation, and the first-lag autocorrelation of the inter-departure times) depending on the building model parameters. The simulations obtained for a wide range of system parameters are then fed into a supervised learning algorithm as the training data. Simulation of these building blocks is discussed in the following part.

3.1. Delay, batching, split, and merge building blocks

3.1.1. Delay

The Delay block shown in Figure 2 can be used to represent a machine or the transportation time between the stations in a manufacturing system. In order to determine the output characteristics of the Delay building block that processes the incoming parts on first come first serve basis, we simulate the inter- arrival and service times. Let Y_a and Y_s be the $N \times 1$ vectors for the interarrival and service times generated with the procedure outlined in Section 2.1 with the given characteristics of the interarrival and service times (μ_a, cv_a, ρ_a) and (μ_d, cv_d, ρ_d), respectively. Let $T_{a,i}$ be the arrival time of the ith part, $T_{s,i}$ and $T_{d,i}$ be the time the service starts and the time the part departs from the system. Starting with $T_{a,0} = 0$, and $T_{s,0} = 0$,

$$\begin{aligned} T_{a,i} &= Y_{a,i} + T_{a,i-1}, \quad i = 1, 2, \ldots, \\ T_{s,i} &= \max\{T_{s,i-1}, T_{a,i}\}, \quad i = 1, 2, \ldots, \\ T_{d,i} &= Y_{s,i} + T_{s,1}, \quad i = 1, 2, \ldots. \end{aligned}$$

Then, the departure stream $Y_d = \{Y_{d,i}\}$ is determined as

$$Y_{d,i} = T_{d,i} - T_{d,i-1}, \quad i = 2, 3, \ldots, \tag{3}$$

The total time part i spends in the delay block, ct_i is

$$ct_i = T_{d,i} - T_{a,i}, \quad i = 1, 2, \ldots \tag{4}$$

The stream $Y_d = \{Y_{d,i}\}$ yields (cv_d, ρ_d) and the stream $\{ct_2, ct_2, \ldots\}$ yields (CT, cv_{CT}).

Since there is no loss in the system, the output rate is equal to the arrival rate. Therefore,

$$\mu_d = \mu_a. \tag{5}$$

The supervised learning algorithm presented in Section 4 is used to find the functions that give (cv_d, ρ_d) and (CT, cv_{CT}) for given (μ_a, cv_a, ρ_a) and (μ_s, cv_s, ρ_s).

3.2. Batching

Many machines in manufacturing systems process parts in batches. Since the size of the batches can be different in different parts of a production system, there is a need for a building block that models the change of a stream due to a change in the batch size. We model this process using the batching block. We assume the batching process is instantaneous when all the parts required to form a batch are available. However, the time to form a batch will generate a delay for an arriving part.

The building block for modelling batching a correlated input stream has not been implemented in other queueing network analysis algorithms. The inputs to the Batching block presented in this study are the incoming stream characterised by (μ_a, cv_a, ρ_a) and the batch size B. The output stream for the departing batches is characterised by (μ_d, cv_d, ρ_d). The mean and the coefficient of variation of the delay are (CT, cv_{CT}). Let Y_a denote the $N \times 1$ vector of the interarrival times and let $T_{d,i}$ denote the ith departure time from the batching block. Then

$$T_{d,i} = \sum_{j=(i-1)B+1}^{iB} Y_{a,j}, \tag{6}$$

and the inter-departure times denoted by $Y_{d,i}$ can be calculated using

$$Y_{d,i} = T_{d,i} - T_{d,i-1}, \quad i = 2, 3, \ldots. \tag{7}$$

The interdeparture process of the batching block is analysed by treating the parts collected into a batch as a single part in the downstream. In other words, parts arrive at the batching block as single units and the batches of these parts are treated as a single part departing the batching block. Therefore,

$$\mu_d = B\mu_a. \tag{8}$$

After arrival, each part has to wait until a batch is completed. Hence the cycle time for the batching block can be calculated as

$$CT = \sum_{i=1}^{B} \frac{i-1}{B}\frac{1}{\mu_a} = \frac{B-1}{2\mu_a}. \tag{9}$$

Since μ_d and CT are given in closed form, the supervised learning algorithm presented in Section 4 is used to find the functions that give (cv_d, ρ_d) and cv_{CT} for given (μ_a, cv_a, ρ_a) and the batch size B.

3.2.1. Split

We consider two split blocks: the Binomial Split block that routes an arriving stream to one of two routes based on a given probability and the Markov chain Split block where the routing process is governed by a first-order Markov Chain.

In the binomial split, the probability of a part going to each downstream route is independent from the routing of the previous parts. In the Markov chain block, the split probability governing the route that a part takes depends on the route of the part preceding it.

Split blocks can be used for modelling quality control stations in a production system. The defects in the parts might stem from various sources and be independent from each other. However, the defects might stem from malfunctions in the servers in which case the defective parts might arrive in clusters. To model the first setting, we use a binomial splitting process and for modelling the second setting we use a Markov chain based splitting process.

Binomial Split. In order to determine the output characteristics of the Binomial Split building block shown in Figure 2, an input arrival stream with the given characteristics is simulated. Let Y_s denote the split stream obtained by routing an arrival stream Y_a with characteristics (μ_a, cv_a, ρ_a) with probability p.

Let $T_{a,i}$ be the time of the ith arrival of the input stream and $T_{s,i}$ be the time of the ith arrival to the split stream. $T_{s,i}$ will be equal to $T_{a,i}$ with probability p. In other words, the inter-departure time of the split process will be the random sum of interarrival times

$$Y_{s,j} = \sum_{i=1}^{L} Y_{a,i}, \tag{10}$$

where L is a random variable that has the Geometric distribution with mean $1/p$, i.e. $Prob[L = n] = (1-p)^{(n-1)}p$. Accordingly, the mean departure rate to the first split stream is equal to the portion of the arrival rate with the split probability of p. Therefore,

$$\mu_d = \frac{\mu_a}{p}. \tag{11}$$

The coefficient of variation of the departure process can be written as

$$cv_d = \sqrt{1 - p(1 - cv_a^2)}. \tag{12}$$

The supervised learning algorithm presented in Section 4 has been used to find the function that yields ρ_d for given (μ_a, cv_a, ρ_a) and p, and μ_d and cv_d are available in closed form.

Markov Chain Split. The Markov Chain building block for splitting a correlated input stream based on a Markov chain that captures the dependence between the subsequent split routes has not been implemented in other queueing network analysis algorithms. The Markov chain

splitting process we propose 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 similarly 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 incoming stream characterised by (μ_a, cv_a, ρ_a) and the Markov chain $\begin{bmatrix} p_1 & 1-p_1 \\ 1-p_2 & p_2 \end{bmatrix}$. The Markov chain split is equivalent to a binomial split block when $p_1 = p$ and $p_2 = 1 - p$.

The mean inter-departure time for the first split stream is

$$\mu_d = \mu_a \frac{2 - p_1 - p_2}{1 - p_2}. \tag{13}$$

The supervised learning algorithm presented in Section 4 has been used to find the functions that yield cv_d and ρ_d for given (μ_a, cv_a, ρ_a), p_1, and p_2.

Using a Markov chain split allows capturing the dependence in the split process. Ignoring this dependence yields errors in capturing the output stream characteristics. Figure A2 in Appendix 3 shows the inaccuracy caused by modelling a Markov chain split as a Binomial split for a specific case. Strong dependencies in the split process can introduce a considerable amount of randomness with larger cv_d values.

3.2.2. Merge

In order to determine the output characteristics of the Merge building block shown in Figure 2, two input arrival streams with the given characteristics are simulated. Let Y_{a1} and Y_{a2} be the $N \times 1$ vectors for the two interarrival time streams generated with the procedure outlined in Section 2.1 with the given characteristics $(\mu_{a1}, cv_{a1}, \rho_{a1})$ and $(\mu_{a2}, cv_{a2}, \rho_{a2})$ respectively. Let T_{a1} and T_{a2} be the corresponding arrival times and let T_d be arrival times of the merged stream. Then the arrival times for the merged stream that will be processed based on the first-come first-served basis will be the sorted arrival times of $\{T_{a1}, T_{a2}\}$ for the arrivals received in $\min\{\max\{T_{a1}\}, \max\{T_{a2}\}\}$ time periods. The difference between the consecutive arrival times of T_d yields the interevent times for the merged stream Y_d. When two streams merge, the total departure rate of the merged stream will be equal to the sum of the two arrival rates. Therefore,

$$\frac{1}{\mu_d} = \frac{1}{\mu_{a1}} + \frac{1}{\mu_{a2}}. \tag{14}$$

Similar to the cases for the delay and split blocks, the supervised learning algorithm presented in Section 4 is used to find the functions that yield (cv_d, ρ_d) for given $(\mu_{a1}, cv_{a1}, \rho_{a1})$ and $(\mu_{a2}, cv_{a2}, \rho_{a2})$.

4. Functional dependency between the output and the input variables for the delay, batching, merge, and split blocks

Determining the functional dependency between the output and input variables for the delay, batching, merge, and split blocks can be viewed as a supervised learning problem. Given a training set, different supervised learning algorithms can be used to capture the relationship between the inputs and the outputs. We use simulation to generate the training data.

In principle, the training data generated for a wide range of system parameters can also be used to approximate the response surface. Although this approximation can also be used to determine the output characteristics for the given input parameters, the memory requirements for storing the response surface will be high. Therefore, the computational requirement to use the prediction of the building block many times in an approximation method will limit using this direct approach.

4.1. Gaussian process regression

Since the training data is obtained by simulation and therefore inherently noisy, the Gaussian Process Regression method has been selected as the most appropriate supervised learning method to determine the functional relationship between the outputs and the inputs of the Delay, Batching, Merge, and Split blocks. Gaussian Process Regression is a non-parametric kernel-based probabilistic method that works well with noisy training data (Rasmussen and Williams 2006).

In this part, we give a brief definition of GPR following (Quiñonero-Candela and Rasmussen 2005; Rasmussen and Williams 2006). For a given training set $\mathcal{D} = \{(\mathbf{x}_i, y_i), i \in \{1, \ldots, n\}\}$ where n is the total number of cases used for training, GPR approach assumes the relationship

$$y_i = f(x) + \epsilon_i \tag{15}$$

between the inputs $\mathbf{x}_i$ and outputs y_i, where $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$. Here, ϵ_i represents the variance of the noise. The goal is approximating $f_* = f(\mathbf{x}_*)$, the response value for a new observation $\mathbf{x}_*$.

Each case given in Table 1 is an input for the corresponding building block. The output variables for each building block are listed in Table 2. For example, when the goal is predicting the cycle time value for the delay block for the given input parameters, $\mathbf{x}_i = [\mu_a\ cv_a\ \rho_a\ \mu_s\ cv_s\ \rho_s]^{\mathrm{T}}$, and y_i is the corresponding CT calculated via simulation. The matrix that stores the training data points is denoted with X, the matrix that stores the test data points is denoted with X_*, and the vector of observed outputs is denoted with $\mathbf{y}$.

Table 1. Range of parameters and the number of cases used for training.

Block type	Parameter	Range	Number of cases
Delay	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	$\{-0.4, \ldots, 0.4\}$	
Batching	μ_a	$\{1\}$	2052
	cv_a	$\{0.1, \ldots, 1.1\} \cup \{1.4\}$	
	ρ_a	$\{-0.4, \ldots, 0.4\}$	
	B	$\{2, \ldots, 20\}$	
Binomial split	μ_a	$\{1\}$	630
	cv_a	$\{0.1, \ldots, 1.4\}$	
	ρ_a	$\{-0.4, \ldots, 0.4\}$	
	p	$\{0.1, \ldots, 0.9\}$	
Markov Chain split	μ_a	$\{1\}$	10,206
	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}	$\{-0.4, \ldots, 0.4\}$	

Table 2. The accuracy of GPR for the different parameters of the different blocks.

		Accuracy		
	Parameter	MAE	MAPE	RMSE
Split	cv_{d1}	0.00062	0.0627	0.0010
	ρ_{d1}	0.00054	2.6074	0.0008
	cv_{d2}	0.00067	0.0782	0.0012
	ρ_{d2}	0.00042	0.4251	0.0007
MC split	cv_{d1}	0.00126	0.1246	0.0019
	ρ_{d1}	0.00054	11.5859	0.0008
	cv_{d2}	0.00131	0.1303	0.0020
	ρ_{d2}	0.00051	20.6185	0.0007
Merge	cv_d	0.00067	0.0973	0.0023
	ρ_d	0.00091	0.6663	0.0056
Batching	cv_d	0.0049	2.1389	0.0080
	ρ_d	0.0285	13.1764	0.0387
Delay	cv_d	0.00186	0.2660	0.0028
	ρ_d	0.00286	0.7543	0.0054
	CT	0.1224	3.61436	0.3229
	cv_{CT}	0.00889	1.1002	0.0186

Gaussian process regression is a Bayesian approach that assumes a Gaussian process prior over functions. A Gaussian process refers to a collection of random variables that any finite number of them follow a joint Gaussian distribution. 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.

Therefore,

$$p(\mathbf{f}|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \mathcal{N}(0, K_{X,X}), \tag{16}$$

where $\mathbf{f} = [f(\mathbf{x}_1)f(\mathbf{x}_2)\ldots f(\mathbf{x}_n)]^{\mathrm{T}}$ is the vector of latent variables.

Bayesian rule yields the predictive distribution

$$p(\mathbf{f}_* \mid \mathbf{y}) = \mathcal{N}(K_{X_*,X}(K_{X,X} + \sigma_n^2 I)^{-1}\mathbf{y}, K_{X_*,X_*} - K_{X_*,X}(K_{X,X} + \sigma_n^2 I)^{-1}K_{X,X_*}), \tag{17}$$

where I is the identity matrix. Since the predictive distribution is Gaussian, the mean and the confidence interval of the predicted value for given $\mathbf{x}_*$ can be obtained from the above distribution.

The computational cost for the matrix inversion step for calculating the mean of this distribution increases rapidly with the number of data points as given in Equation (17). There are several methods to improve the computational performance of this method (Quiñonero-Candela and Rasmussen 2005). For allowing more flexibility, a set of basis functions can be integrated into the Gaussian process regression. With this approach, the best fit amongst a number of kernel functions and a set of basis functions can be selected.

The functional form resulting from Gaussian Process Regression method cannot be interpreted. Our experiments with Symbolic Regression yielded interpretable functions but their overall predictive performance was lower. Similarly, our experiments with neural networks also gave prediction performance comparable to the Gaussian Process Regression. GPR was chosen since it is more accurate compared to other methods and the uncertainty measurements on the predictions are available when GPR is used.

4.2. Training data

In order to generate the input–output data to train supervised learning algorithms, the ranges given in Table 1 are used for each stream. The ranges of the coefficient of variations and the first-lag autocorrelation are in line with our observations at Bosch Reutlingen plant. A total of 314,532 cases are simulated. For each simulation, 10,000 interevent times are used for each stream and replicated 100 times.

As described in Section 4.3.2, instead of generating the traces of interevent times each time for different parameters, the traces generated beforehand and stored in the memory are retrieved whenever they are used. This approach saves around 2 min of processing time for getting the results for the Delay and Merge blocks and 1 min for the Split block for each case. Even with the time saving obtained with this approach, generating the training data on a personal computer with a `Intel(R) Core(TM) i5-3340M 2.7GHz` CPU would approximately require 6 days for the Delay block, 1 h for the

Binomial split block, 11 h for the Markov chain split block, 1 h for the Batching block, and 107 days for the merge block on this computer. In order to speed up the time to generate the training data, parallel simulation is used on a computer cluster with 128 nodes to generate the training data. Note that the training data is generated only once to train the prediction models for the building blocks.

4.3. Simulating correlated interevent time sequences

In order to model non-negative interevent times with a wide range of coefficient of variation values, the Weibull distribution is used with the given mean μ and coefficient of variation cv.

4.3.1. Correlated Weibull sequences

The density function of the Weibull distribution is determined by two parameters α and β:

$$f(x) = \frac{\alpha}{\beta^{\alpha}} x^{\alpha-1} e^{-(x/\beta)^{\alpha}}, \quad x > 0. \tag{18}$$

The coefficient of variation of the Weibull distribution is determined by α:

$$cv = \sqrt{\frac{\Gamma\left(1+\frac{2}{\alpha}\right)}{\Gamma\left(1+\frac{1}{\alpha}\right)^2} - 1}, \tag{19}$$

where $\Gamma(\cdot)$ is the gamma function. The parameter α for a given cv is determined by solving the above equation. Once α is determined for a given cv, β is determined as

$$\beta = \frac{\mu\alpha}{\Gamma(1/\alpha)}. \tag{20}$$

The kth lag autocorrelation of the sequence $X_1, X_2, \ldots$ is expressed in terms of only the first-lag autocorrelation ρ_1 as

$$\rho_k = \rho_1^k, \quad i = 1, 2, \ldots. \tag{21}$$

The autocorrelation function for N correlated Weibull random variables can be expressed by a $N \times N$ correlation matrix S with its (i,j)th element given as

$$S_{i,j} = \rho_1^{|i-j|}, \quad i = 1, 2, \ldots, N,\ j = 1, 2, \ldots, N. \tag{22}$$

For a Weibull sequence, the exponential approximation for the autocorrelation function is an accurate representation (Novak 1973). Figure 3 shows the autocorrelation function of the departure process from a single station with correlated interarrival and service times. The comparison between the simulation and the exponential approximation given in Equation (21) shows that the autocorrelation function of the departure process can be approximated accurately by using only the first-lag autocorrelation.

4.3.2. Simulating correlated random variables

There are various methods to generate correlated random variables, e.g. Novak (1973), Kriege and Buchholz (2011) and Wang and Xin (2017). The methods developed for this purpose also include using a shared random variable and using Eigenvalue decomposition for generating autocorrelation (Geist 1979; Magnussen 2004). The method used in this study is based on generalising the approach given by Novak (1973) and Wang and Xin (2017).

In order to generate a sequence of N Weibull random variables with the kth lag autocorrelation given as $\rho_k = \rho_1^k$, $k = 1, 2, \ldots$, first, a sequence of N multinomial normal random variables with mean 0, standard deviation σ and correlation matrix $S' = \{S'_{i,j}\}$ where $S'_{i,j} = \rho_{in}^{|i-j|}$ is generated. The first-lag autocorrelation that is used to construct S', ρ_{in} is chosen appropriately to generate a sequence with the desired correlation matrix S.

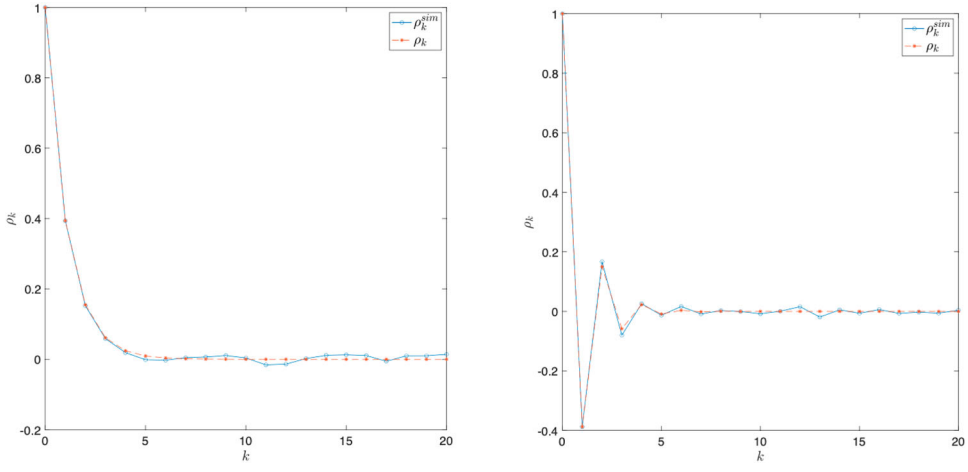

Figure 3. The simulated and the approximated autocorrelation function for the departure process of the Delay Block with correlated interarrival and service times modelled as Weibull sequences.

Let $X = \{X_i\}$ be a $N \times 1$ vector with its ith element is a normal random variable X_i and the sequence $X_1, X_2, \ldots$ be independent. A correlated sequence $Y = \{Y_i\}$ is built by using a linear transformation of the sequence X given by

$$Y = A^{\mathrm{T}} X. \tag{23}$$

In the above equation, the matrix A satisfies

$$AA^{\mathrm{T}} = S' \tag{24}$$

and it can be determined by using the Cholesky decomposition. This transformation yields the sequence $Y = \{Y_i\}$ where Y_i is distributed according to the Normal distribution with the mean 0 and standard deviation σ and the kth lag autocorrelation of the sequence is $\rho_k = \rho_{in}^k$.

Next, we generate N uniform random variables with the correlation matrix S' by using the standard normal probability distribution function $\Phi(cdot)$:

$$U_i = \Phi\left(\frac{Y_i}{\sigma}\right). \tag{25}$$

Finally, a sequence of N Weibull random numbers where W_i with mean μ, coefficient of variation cv, parameters α and β, correlation matrix S is generated from the uniform random variables U_i according to

$$W_i = \beta(-ln(U_i))^{1/\alpha}, \quad i = 1, 2, \ldots \tag{26}$$

Due to the nonlinear transformation given in Equation (25), the resulting correlation matrix of W_i, $i = 1, 2, \ldots$ is different from the correlation matrix initially used. However, there is a one-to-one relationship between the input first-lag autocorrelation denoted with ρ_{in} that is used to generate U_i and the first-lag autocorrelation of W_i, ρ_1. This function is uniquely determined by the coefficient of variation of the random variable. We determine the mapping function between ρ_{in} and ρ_1 by simulating correlated sequences for a wide range of cv values and then fitting a fifth-degree polynomial curve that yields ρ_{in} for a desired ρ_1 when cv is given. Since there are 14 different cv values used in the numerical experiments given in Table 1, 14 different $(\rho_1 \rightarrow \rho_{in})$ mapping functions are determined. Figure 4 shows the mapping functions that yield the input first-lag autocorrelation (ρ_{in}) for the desired first-lag autocorrelation (ρ_1) for two different values of coefficient of variation.

As a summary, in order to generate N Weibull random variables with mean μ, coefficient of variation cv, and correlation matrix $S = \{S_{i,j}\}$ where $S_{i,j} = \rho_1^{|i-j|}$, first, the input first-lag autocorrelation ρ_{in} is determined from the predicted $(\rho_1 \rightarrow \rho_{in})$ mapping function for the given cv. Then, N uniform numbers with 0 mean and correlation matrix S' are generated by using Equation (25). Finally, N Weibull numbers with the mean μ, coefficient of variation cv, and correlation matrix S are generated by using Equation (26). Figure A1 in Appendix 1 shows the histogram and autocorrelation function of simulated uncorrelated and correlated Weibull random variables generated by using the methodology presented in this section.

In the above process, determining the matrix A through the Cholesky decomposition given in Equation (24) is computationally demanding for large N. Since the matrix A is determined by the first-lag autocorrelation, in order to increase the computational efficiency, m replications of Y are generated by using Equation (23) where A is calculated only once for a given cv and ρ_1 by using Equation (24). Furthermore, for the delay and merge systems, replications generated for one of the inputs, i.e. arrival and service for the delay system and the two input streams for the merge system are used in different permutations to obtain different outputs.

We generate the traces for Weibull random variables with mean 1, the coefficient of variation cv, and the first-lag autocorrelation ρ_1 and then rescale the traces for a

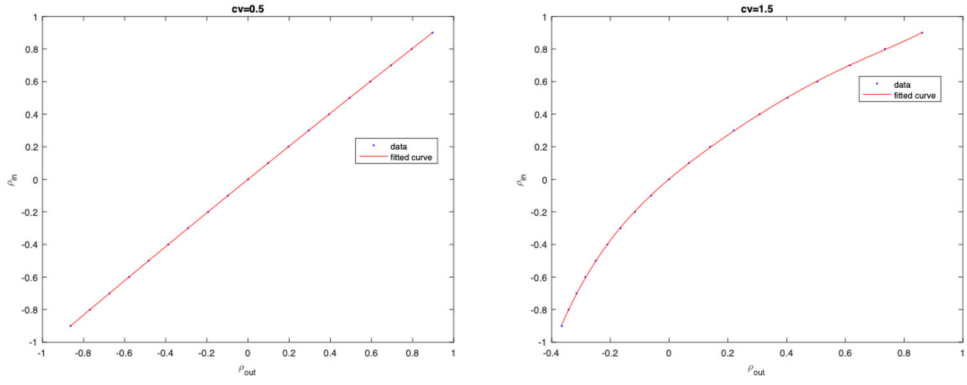

Figure 4. The input first-lag autocorrelation (ρ_{in}) for different values of the desired first-lag autocorrelation (ρ_{out}) and the polynomial fit for the $(\rho_1 \rightarrow \rho_{in})$ mapping function for $cv = 0.5$ and $cv = 1.5$.

given μ. Generating the traces for 1000 replications of 10,000 interevent times for a given cv and ρ_1 takes about 1 minute on a personal computer with a `Intel(R) Core(TM) i5-3340M 2.7GHz` CPU. In the numerical experiments, we use the simulated traces for different interarrival and service time processes given in Table 1. Since there are 14 different cv values and 9 different ρ_1 values used, 126 different trace matrices that contain 1000 replications of 10,000 correlated interevent times have been generated and stored in the memory to be retrieved later on whenever a correlated stream is needed. This approach eliminates the need for generating correlated traces for different cases repeatedly to obtain the training data for Delay, Batching, Merge, and Split building blocks.

4.4. Accuracy of GPR

Using Gaussian Process Regression with the training data yields accurate prediction models for the building blocks. Table 2 gives the MAE, MAPE and RMSE values for the output characteristics for the Delay, Batching, Split, and Merge building blocks. For the Delay and the Merge blocks, as a portion of the data points have been used for training, we report the out-of-sample MAE, MAPE and RMSE values in Table 2. However, given that we have used all or nearly all of the data points in training GPR for the other blocks, we report their in-sample accuracy in Table 2. Additionally, the out-of-sample performances of all the blocks were examined extensively as parts of the networks we use for the numerical experiments in Section 5. Figure 5 gives the predicted and simulated coefficient of variation, first-lag autocorrelation, and expected cycle time values for the building blocks.

4.4.1. Delay

The Gaussian Process Regression has been used to obtain 5 functions that yield (cv_d, ρ_d) and (CT, cv_{CT}, ρ_{CT}) for given (μ_a, cv_a, ρ_a) and (μ_s, cv_s, ρ_s) for the Delay building block. In addition to the input parameters (μ_a, cv_a, ρ_a) and (μ_s, cv_s, ρ_s), the cycle time and the departure coefficient of variation approximations for a G/G/1 system with i.i.d. interarrival and service times, given by Equations (1) and (2), respectively, are also used as the additional input features to predict the cycle time and the departure coefficient of variation for the delay block with correlated interarrival and service times. 10,000 samples have been selected randomly from the total of 142,884 cases for training GPR.

The results shown in Table 2 and Figure 5 show that the prediction obtained by GRP is very accurate in the whole range of parameters. The RMSE errors obtained for prediction of cv_d, ρ_d, CT are 0.0028, 0.0054, and 0.32, respectively.

Comparison of the Single-Station Cycle Time Prediction with the Analytical Approximations. The supervised learning approach yields more accurate cycle time prediction compared to analytical approximations (Kingman 1961; Krämer and Langenbach-Belz 1976; Marchal 1976; Buzacott and Shanthikumar 1993) and the robust queueing approximation (Whitt and You 2020).

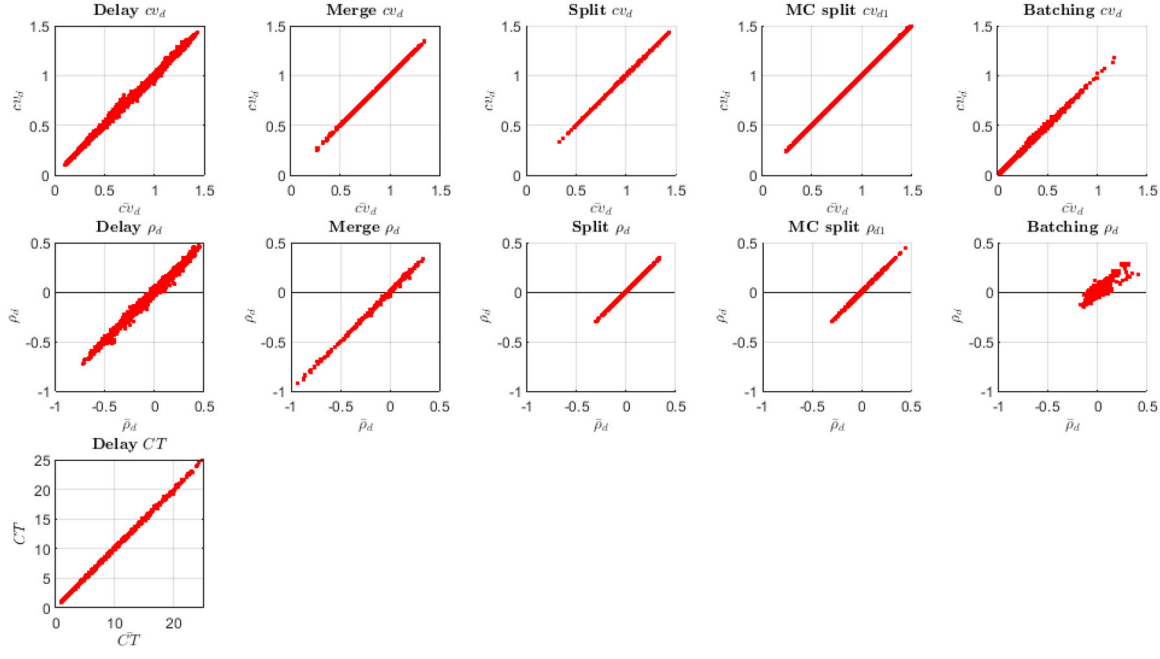

Figure 5. Predicted (CT) and simulated ($\bar{CT}$) cycle time values for the Delay block.

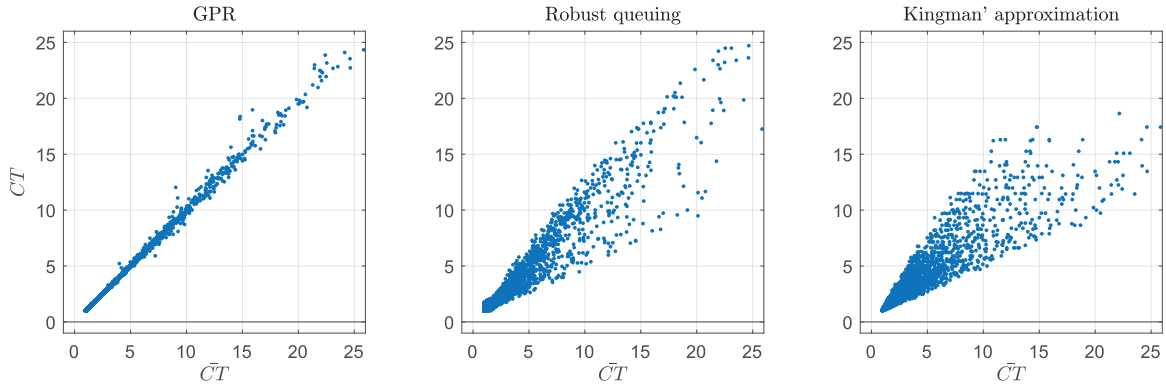

Figure 6. Comparison of cycle time values predicted by GPR, Robust Queueing, and Kingman's methods (CT) and simulation ($\bar{CT}$) for the Delay block.

Figure 6 shows the simulated cycle times for the delay block with correlated interarrival and service times and its approximated value obtained by using GPR, Kingman's approximation given in Equation (1) that ignores autocorrelation and the robust queuing approximation that only incorporates the autocorrelation in the arrival process for different interarrival and service time characteristics given in Table 1.

As Figure 6 shows the approximation given in Equation (1) yields results that can be quite different from the actual results. The prediction obtained by Kingman's approximation given in Equation (1) can be 100% longer or 75% shorter than the simulation depending on the system parameters. The average absolute error is 17% for all the cases with autocorrelated interarrival and service times. When the service time has no autocorrelation, but the interarrival time is correlated ($\rho_s = 0$) the average absolute error is 14.3% and for the cases with no autocorrelation ($\rho_a = 0, \rho_s = 0$), the average absolute error is 8%. Even for the case with no autocorrelation, the performance of the analytical approximation deteriorates significantly as the coefficient of variation of the service and interarrival time increases.

Table A1 in Appendix 2 gives the mean absolute percentage errors of cycle times obtained by using approximations Kingman (1961), Marchal (1976), Krämer and Langenbach-Belz (1976), three different approximations B&S 1, B&S 2, B&S 3 given in Buzacott and Shanthikumar (1993) and the error obtained by the single-station delay block of SLQNA. The results show that the average cycle time predicted by the delay block of SLQNA is very accurate with a MAPE of 1% while all the other approximations give a MAPE greater than 11%. In addition, as the accuracy of SLQNA is not affected by the arrival coefficient of variation, the performances of all the other methods deteriorates significantly for highly variable arrivals.

Since these approximations are derived under the assumption of i.i.d. interarrival and service times, Table A2 in Appendix 2 gives the percentage errors obtained by these approximations and SLQNA for the cases with no autocorrelation, i.e.$\rho_s = \rho_a = 0$. The results show that for the cases with no correlation, SLQNA still gives the most accurate results with a MAPE of 1% while the best approximations yield a MAPE of 2%.

The objective of using a supervised learning approach is obtaining a functional relationship between the input and output characteristics that is much more accurate for all parameter ranges and also incorporates autocorrelation in the interarrival and service times.

4.4.2. Batching

GPR is used to predict the output cv_d and ρ_d for the Batching building block based on the characteristics of the input flow and the batch size (μ_a, cv_a, ρ_a, B). The RMSE error for the output cv_d is 0.0080 and for the output ρ_d is 0.0387. Figure 5 gives the accuracy of GPR for predicting these output parameters.

4.4.3. Split

Gaussian Process Regression has been used to find two functions that yield (cv_d, ρ_d) for given (μ_a, cv_a, ρ_a) and p for the Binomial Split block. All of the 630 cases have been used for training GPR. For the Markov Chain Split block, two functions that yield (cv_d, ρ_d) for given (μ_a, cv_a, ρ_a) and p_1, p_2 are also obtained by using GPR. The RMSE errors obtained for prediction of cv_d and ρ_d are 0.0010 and 0.0008 for the Binomial Split. The RMSE errors for the interdeparture coefficient of variation from the Markov Chain Split Block, cv_{d1} and cv_{d2} are 0.0019 and 0.0020 and the RMSE errors for the first-lag autocorrelation of the interdeparture streams, ρ_{d1} and ρ_{d2} are 0.0008 and 0.0007, respectively.

4.4.4. Merge

For the Merge building block, two functions that yield (cv_d, ρ_d) for given $(\mu_{a1}, cv_{a1}, \rho_{a1})$ and $(\mu_{a1}, cv_{a1}, \rho_{a1})$ are determined by using GPR. 10,000 samples have been selected randomly from the total of 158,760 cases for training GPR. The RMSE errors obtained for prediction of cv_d and ρ_d are 0.0023 and 0.0056, respectively.

5. An approximation method for open queueing networks: SLQNA

A given manufacturing system can be analysed by decomposing the network into different Delay, Batching, Merge, and Split components and determining the output stream characteristics by using the input characteristics. For example, Figure 7 shows a workcell with a workstation, a quality control point, and a rework station and its decomposition into delay, merge, and split components.

The performance measures such as the average number of parts waiting in front of the workstation and for rework, the average time for an arriving part to complete its operation in this workcell with or without rework, among others can be determined by using the method presented in this study.

5.1. SLQNA algorithm

The SLQNA approximation uses the functions obtained by using the approach given in Section 4 to determine the mean, the coefficient of variation, and the first-lag autocorrelation of the inter-departure time as functions of the mean, coefficient of variation and first-lag autocorrelations of the interarrival and service times for each Delay, Merge, and Split Block.

Starting with the first station with its given interarrival time parameters, the departure process parameters of each station obtained by the prediction functions are used as the input arrival parameters at the following building block in the network.

This process is repeated until all the stations in the network are processed. The cycle times obtained at each station are then combined to determine the total cycle time of the system. This method also generates the departure coefficient of variation and the first-lag autocorrelation of the departure process at each node of the network. The SLQNA algorithm is given in the Appendix.

Although the approach used in other approximation methods such as QNA is similar since the parameters describing the output processes are passed to the next block as its input parameters, SLQNA differs from QNA since it also passes information about the autocorrelation structure of the process not only the distribution. Although MAPQNA also passes the information about the correlation structure, SLQNA is more accurate for capturing the autocorrelation sequence. This improved accuracy is due to capturing the autocorrelation structure of a Weibull sequence very accurately with the first-lag autocorrelation that is passed between the blocks.

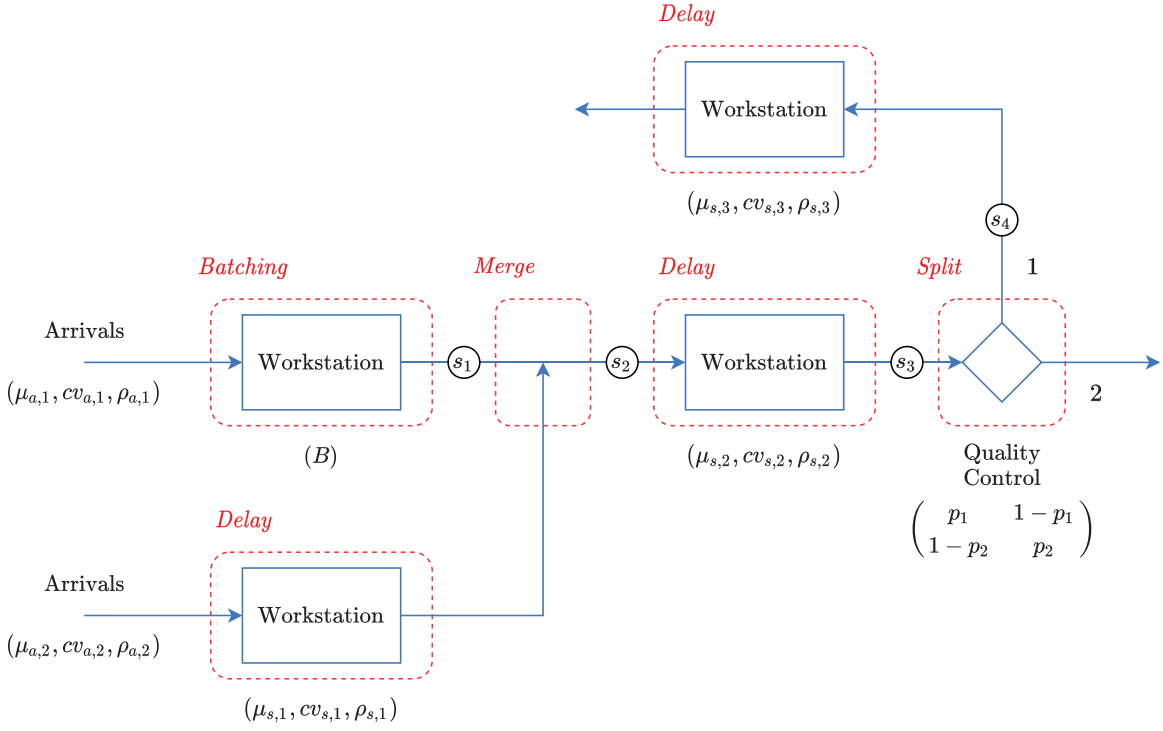

Figure 7. A workcell with two workstations, a batching station, and a quality control station and its decomposition into Merge, Delay, Batching, and Markov Chain Split components with the inspection points for departure process characterisation.

MAPQNA approximates a given autocorrelation structure by truncating the state space. The batching and Markov chain split building blocks are not implemented in QNA and MAPQNA.

5.2. Numerical experiments

In our experimental setup, we first investigate the accuracy of SLQNA in terms of predicting the departure process coefficient of variation and the first-lag autocorrelation at different points of a network that is constructed by using Delay, Batching, Merge, and Split building blocks.

We then compare the performance of SLQNA with simulation and with the other approximation methods for two different network structures. Production lines and a split-merge network with different number of stations (up to 25 stations for production lines and up to 12 stations for the split-merge network) are analysed for a range of system parameters. The system parameters include the number of stations, the mean, coefficient of variation, and the first-lag autocorrelation of the interarrival time process to the first station and the mean, coefficient of variation, and the first-lag autocorrelation of production time processes.

For the experiments with serial lines, we compare the performance of SLQNA with QNA that does not consider the interarrival and service time dependency, Rob-QNA that considers correlated interarrival times but i.i.d. service times, and MAPQNA that considers correlated interarrival and service times. For the experiments with the split-merge network, we compare the performance of SLQNA with QNA and MAPQNA. We report the computation times for each method.

The average total cycle time at the system level (from the first station to the processing time completion at the last station), denoted as CT is used as the main performance measure in the numerical experiments. For serial arrangement of stations, CT is the sum of the cycle times of all the stations in the line. For a network, CT is calculated as the weighted sum of all the cycle times of the single-server stations on the routes connecting the first station to the last station. The calculation of CT for a network is given in Appendix A.2.

We focus on the accuracy of the cycle time prediction measured as the mean absolute percentage error with respect to the simulation result. That is, $\text{MAPE}\bar{CT} = \frac{|CT-\bar{CT}|}{\bar{CT}}$ where $\bar{CT}$ is the actual cycle time (estimated by using simulation) and CT is the cycle time prediction obtained by using an approximation method. The 95% confidence intervals for the total cycle time obtained by using simulation are on average 0.2% of the average values and therefore can be used as a reliable estimate for the actual cycle time. The average results as well as the MAPE distribution for CT considering all the cases used in the numerical experiments are reported. The effects of the number of stations, arrival and service time coefficient of variations and first-lag autocorrelations, the station utilisation, and the split probability on the accuracy of SLQNA are also investigated through the numerical experiments.

5.3. Accuracy of SLQNA to capture departure process characteristics

The accuracy of a given approximation method depends on how well the method captures the characteristics of flows, the mean, coefficient of variation, and autocorrelation, at different locations of a given queueing network.

To examine the accuracy of the SLQNA algorithm in predicting the characteristics of the flows in a network, we use the workcell depicted in Figure 7. This network includes the Delay, Batching, Merge, and Split building blocks.

In this setup, $s_1, \ldots, s_4$ indicate the points where we inspect the flow characteristics. These points are consecutive points along one of the four routes a part can take in this system. Let $(\mu_{a,i}, cv_{a,i}, \rho_{a,i})$ denote the parameters of the arrival processes in the system, let $(\mu_{s,i}, cv_{s,i}, \rho_{s,i})$ denote the parameters of the service processes in the system.

Table 3 gives the parameter sets used in these experiments. In total, 81 cases were considered. Table 4 and Figure 8 give the accuracy of SLQNA for predicting the characteristics of the flows in different locations in the network. Based on these results, As a result, we can conclude that SLQNA captures the flow process characteristics accurately for the network shown in Figure 7.

5.4. Numerical experiments for production lines

In this section, we present the results of the numerical experiments for serial arrangement of workstations in

Table 3. Range of parameters used for the Workcell with delay, batching, merge, and Markov Chain split blocks.

Parameter	Range
$cv_{a,1}$	{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}

Table 4. The accuracy of SLQNA for predicting the output characteristics at different locations in the Workcell network with Delay, Batching, Merge, and Markov Chain Split building blocks.

	cv				ρ				
	s_1	s_2	s_3	s_4	s_1	s_2	s_3	s_4	*CT*
MAE	0.004	0.012	0.017	0.016	0.013	0.010	0.015	0.030	0.275
MAPE	0.821	1.463	2.073	1.886	168.432	29.163	79.652	228.448	4.080
RMSE	0.005	0.014	0.019	0.021	0.017	0.013	0.018	0.038	0.432

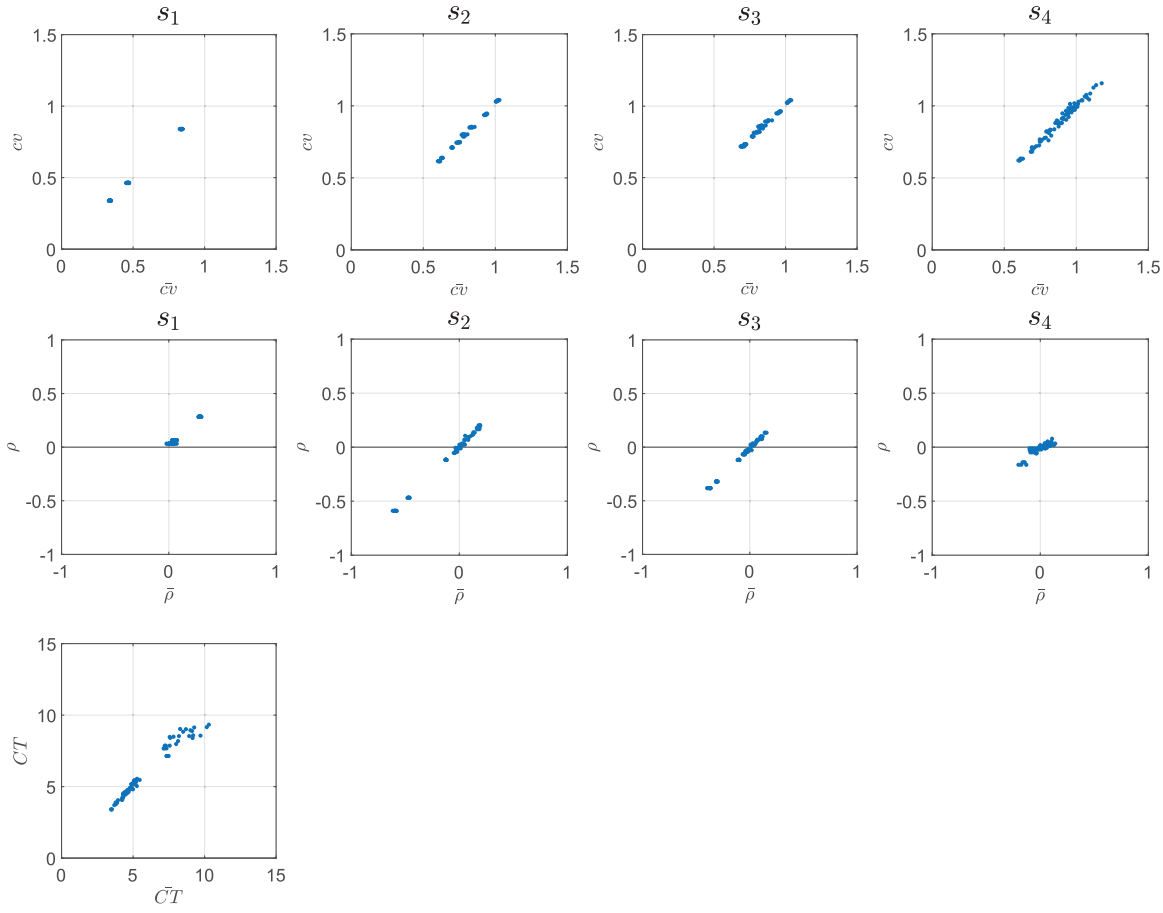

Figure 8. The accuracy of SLQNA in different locations in the Workcell network.

a production line with up to 25 stations and correlated interarrival and processing times. Figure 9 depicts the structure of the system analysed. Table 5 gives the range of parameters used in the experiments. Accordingly, the results based on 91,125 cases are reported.

5.4.1. Accuracy of SLQNA compared to other methods for production lines

In order to characterise the flows and predict the cycle times in a production line, SLQNA and MAPQNA methods allow incorporating autocorrelation in interarrival times and service times while Rob-QNA incorporates correlated interarrival times and i.i.d. service times and QNA considers only i.i.d. interarrival and service times. Accordingly, SLQNA can only be compared directly with MAPQNA when both the interarrival and service times are correlated. We include comparisons with MAPQNA, Rob-QNA, and QNA in our numerical experiments to report both the comparison of accuracy and also the effect of autocorrelation on the accuracy of these methods.

Table 6 reports MAPE obtained predicting *CT* using SLQNA, MAPQNA, Rob-QNA, and QNA in all cases depending on the autocorrelation of interarrival and service times. With a MAPE of 3.4%, SLQNA yields the most accurate predictions among all the other methods. The second most accurate approximation is MAPQNA with a MAPE of 6.4%. The results indicate that the comparative advantage of MAPQNA over QNA depends on the presence of autocorrelation in the system. When service processes are i.i.d., the autocorrelation of the arrival process vanishes in the downstream and in the instances

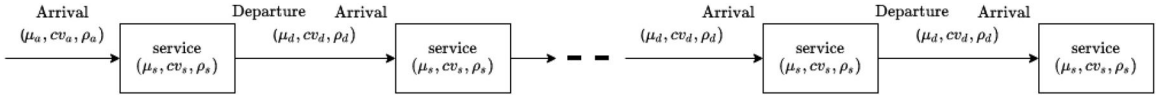

Figure 9. The structure of the Production Line.

Table 5. Range of parameters used for production line experiments with homogeneous stations.

Parameter	Range
μ_a	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}
μ_s	{1}
cv_a, cv_s	{0.4, 0.6, 0.8, 1.0, 1.2}
ρ_a, ρ_s	{−0.4, −0.3, −0.2, −0.1, 0, 0.1, 0.2, 0.3, 0.4}
N	{5, 10, 15, 20, 25}

with a large number of stations this leads to a large portion of the stations performing similar to an i.i.d. system. In these cases, MAPQNA and QNA have comparable performance. However, when the service process is correlated, all of the stations in the system can be affected by autocorrelation and as a result MAPQNA outperforms QNA in these cases. The similarity between the performance of Rob-QNA and QNA can be attributed to the use of Weibull distribution in this work in that while Rob-QNA is a distribution-free method, QNA has been developed based on distributions that can effectively approximate the Weibull distribution.

Figure 10 depicts the histograms of MAPE obtained predicting CT using SLQNA, MAPQNA, Rob-QNA, QNA in all cases depending on the autocorrelation of interarrival and service times. The figure shows that not only SLQNA is more accurate on average, its predictions are more robust. The cycle time predictions for individual stations follow a similar pattern.

The accuracy of SLQNA is affected by different system parameters including the autocorrelations of interarrival and service processes, the coefficient of variation of interarrival and service times, the utilisation of individual stations, and the number of stations. Our analysis given in Appendix A.3 shows that service time coefficient of variation affects the accuracy more compared to the interarrival time variability. Furthermore, higher utilisation and the presence of autocorrelation in the system decreases the accuracy of all the methods. However, the presence of dependency in the system affects SLQNA less compared to MAPQNA and Rob-QNA methods. Finally, the accuracy of SLQNA does not get affected negatively with the increasing number of stations.

5.4.2. Computational performance of SLQNA compared to other methods for serial lines

SLQNA is a time-efficient method for queuing network analysis as the computationally heavy tasks for SLQNA have to be performed only once. Once the trained blocks are available, predicting the performance measures for a new network can be performed in seconds. MAPQNA is a computationally expensive method. The main contributor to the computational effort to run MAPQNA is the MAP fitting step during the execution of the algorithm. QNA's computational requirement is very low as it uses closed-form approximations. Rob-QNA can be as fast as QNA and SLQNA if the indices of dispersion for counts for the processes in a given system are available. In this work, we calculate the indices of dispersion for counts for each set of flow parameters based on 1000 traces. Figure 11 gives the effect of the network size on the computational requirement for the methods based on a sample of 400 instances selected randomly from the cases studied for the production line experiments.

5.5. Numerical experiments for a network with merge and split

In this section, we report the results of experiments with a network that consists of a station that splits the arrival

Table 6. The accuracy of different methods in predicting the cycle time for the production line experiments.

	Mean absolute percentage error $100\frac{\lvert CT-\tilde{CT}\rvert}{\tilde{CT}}$			
	$\rho_a \in \{-0.4, \ldots, 0.4\}$ $\rho_s \in \{-0.4, \ldots, 0.4\}$	$\rho_a = 0$ $\rho_s \in \{-0.4, \ldots, 0.4\}$	$\rho_a \in \{-0.4, \ldots, 0.4\}$ $\rho_s = 0$	$\rho_a = 0$ $\rho_s = 0$
SLQNA	3.41	2.91	2.75	2.23
MAPQNA	6.43	4.14	5.60	2.72
Rob-QNA	11.51	10.23	5.38	4.87
QNA	10.86	9.86	4.55	2.52

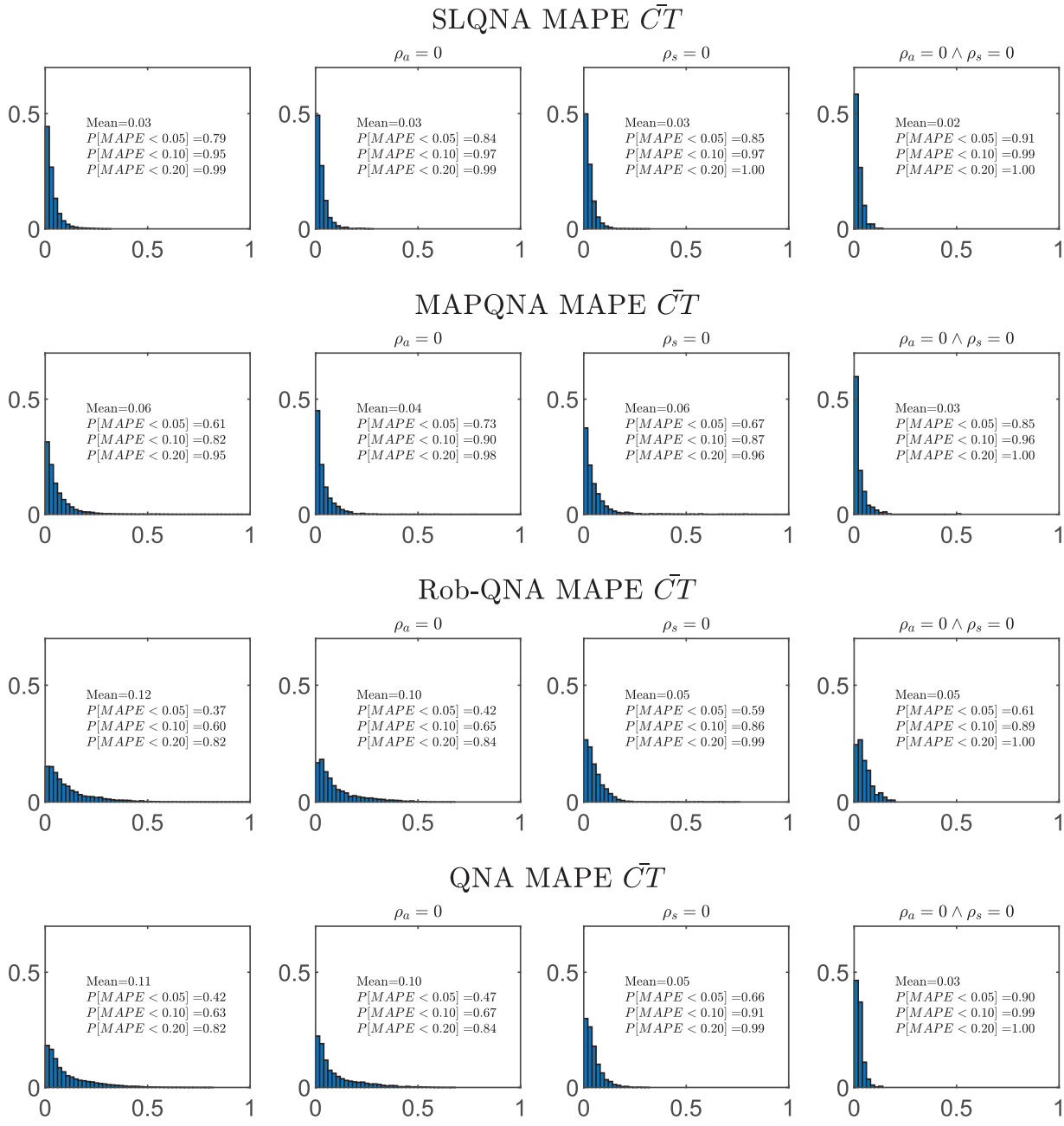

Figure 10. Histogram of the Mean Absolute Percentage Error (MAPE) of the cycle time predictions obtained by using different methods for the production line experiments.

stream into two lines with varying number of stations that merge at the last station. The network analysed is shown in Figure 12. By using this network, we compare the performance of the SLQNA with QNA and MAPQNA methods for a range of parameters.

Table 7 gives the set of parameters used for these experiments, where $\lambda_a = 1/\mu_a$, $\lambda_{s,1} = 1/\mu_{s,1}$ and $\lambda_{s,2} = 1/\mu_{s,2}$. 1178 cases have been examined in total. Each case consists of $2l + 2$ stations.

5.5.1. Accuracy of SLQNA compared to other methods for the split-Merge network

Table 8 gives the mean absolute percentage error in the cycle time predictions obtained by SLQNA, MAPQNA, and QNA in 1178 cases given in Table 7.

The results indicate that SLQNA is twice more accurate than the other methods for all cases. Even for the cases where the interarrival and service times are not correlated, SLQNA is still more accurate than QNA that was

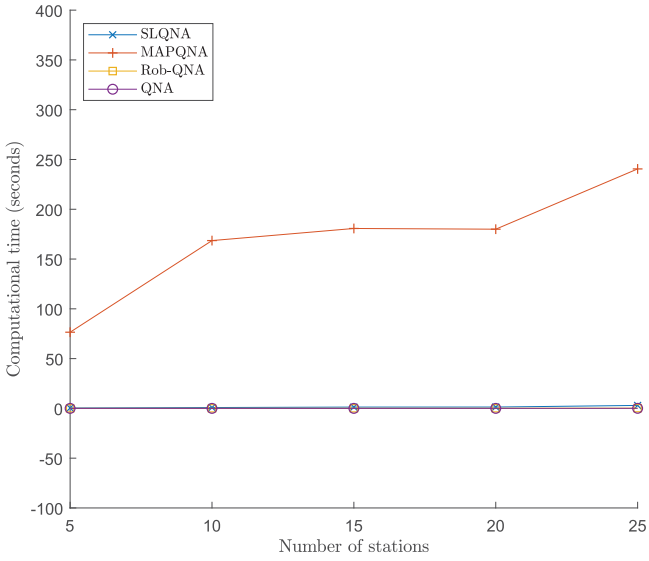

Figure 11. Effect of number of stations on the average computational time of the methods for the production line experiments.

developed under the assumption of i.i.d. interarrival and service times.

Figure 13 shows the histogram of MAPE values for the cycle time obtained by using SLQNA, MAPQNA, and QNA. The results show that SLQNA is more accurate and more robust compared to MAPQNA and QNA.

Our analysis given in Appendix A.4 investigates the effect of system parameters that are the number of stations in the network, the split probability, arrival coefficient of variation, service time coefficient of variations of two lines and the first-lag autocorrelations of the arrival and service processes on the SLQNA accuracy for the Split-Merge Network. The results show that that the accuracy of the approximation methods increases as the number of stations in the network increases and the split probability is varied. Furthermore, SLQNA is considerably less sensitive to the changes in utilisation in the system compared to the other methods.

5.5.2. Computational performance of SLQNA compared to other methods for the split-merge network

Figure 14 gives the effect of the network size on the computational time of the methods for 50 randomly selected

Table 7. The range of parameters used for the experiments with the network with split and merge.

Parameter	Range
λ_a	{1}
$\lambda_{s,1}, \lambda_{s,2}$	{1.25, 1.75, 2.5}
$cv_a, cv_{s,1}, cv_{s,2}$	{0.4, 0.8}
$\rho_a, \rho_{s,1}, \rho_{s,2}$	{−0.3, 0, 0.3}
p	{0.2, 0.5, 0.8}
l	{1, 3, 5}

parameter sets. On average, evaluation of each case given in Table 7 using SLQNA took 1 s while the evaluation of each case by MAPQNA took 50 s on a personal computer with a `Intel(R) Core(TM) i5-3340M 2.7GHz` CPU. Due to computational time limits, the size of the search space for approximate MAPs used by MAPQNA was limited to 16 and cases where larger MAPs were required for MAPQNA were dropped from further consideration.

6. Conclusions

In this study, we propose a supervised learning based approximation method to analyse single-server open queueing network models of manufacturing systems composed of delay, batching, split, and merge building blocks with correlated interarrival and service times. We determine the mean, coefficient of variation, and first-lag autocorrelation of the output stream of the building blocks as a function of the mean, coefficient of variation and first-lag autocorrelations of the input interevent times by using a supervised learning approach. For the output parameters that cannot be determined analytically, we simulate each block for a wide range of system parameters in parallel in a multi-node computer cluster and use the input-output sets as training sets for the Gaussian Process Regression. Our results show that the results obtained by using the Gaussian Process Regression are very accurate and better than the available analytical approximations to determine the output characteristics based on the input characteristics.

These building blocks are then used to build single-server open queueing networks to evaluate the performance of manufacturing systems in a computationally efficient way. The method we propose in this study,

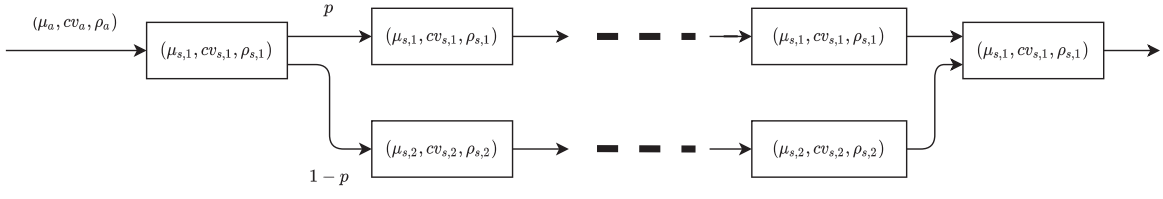

Figure 12. The structure of the network with merge and split.

Table 8. The accuracy of different methods in predicting the cycle time for the network with split and merge.

	Mean absolute percentage error $100\frac{\lvert CT-\bar{CT}\rvert}{\bar{CT}}$			
	$\rho_a \in \{-0.3, 0, 0.3\}$ $\{\rho_{s,1} \in \{-0.3, 0, 0.3\}\}$ $\{\rho_{s,2} \in \{-0.3, 0, 0.3\}\}$	$\rho_a = 0$ $\{\rho_{s,1} \in \{-0.3, 0, 0.3\}\}$ $\{\rho_{s,2} \in \{-0.3, 0, 0.3\}\}$	$\rho_a \in \{-0.3, 0, 0.3\}$ $\rho_{s,1} = 0$ $\rho_{s,2} = 0$	$\rho_a = 0$ $\rho_{s,1} = 0$ $\rho_{s,2} = 0$
SLQNA	3.74	2.93	3.50	2.41
MAPQNA	8.19	5.25	6.73	3.64
QNA	8.30	5.94	6.95	3.58

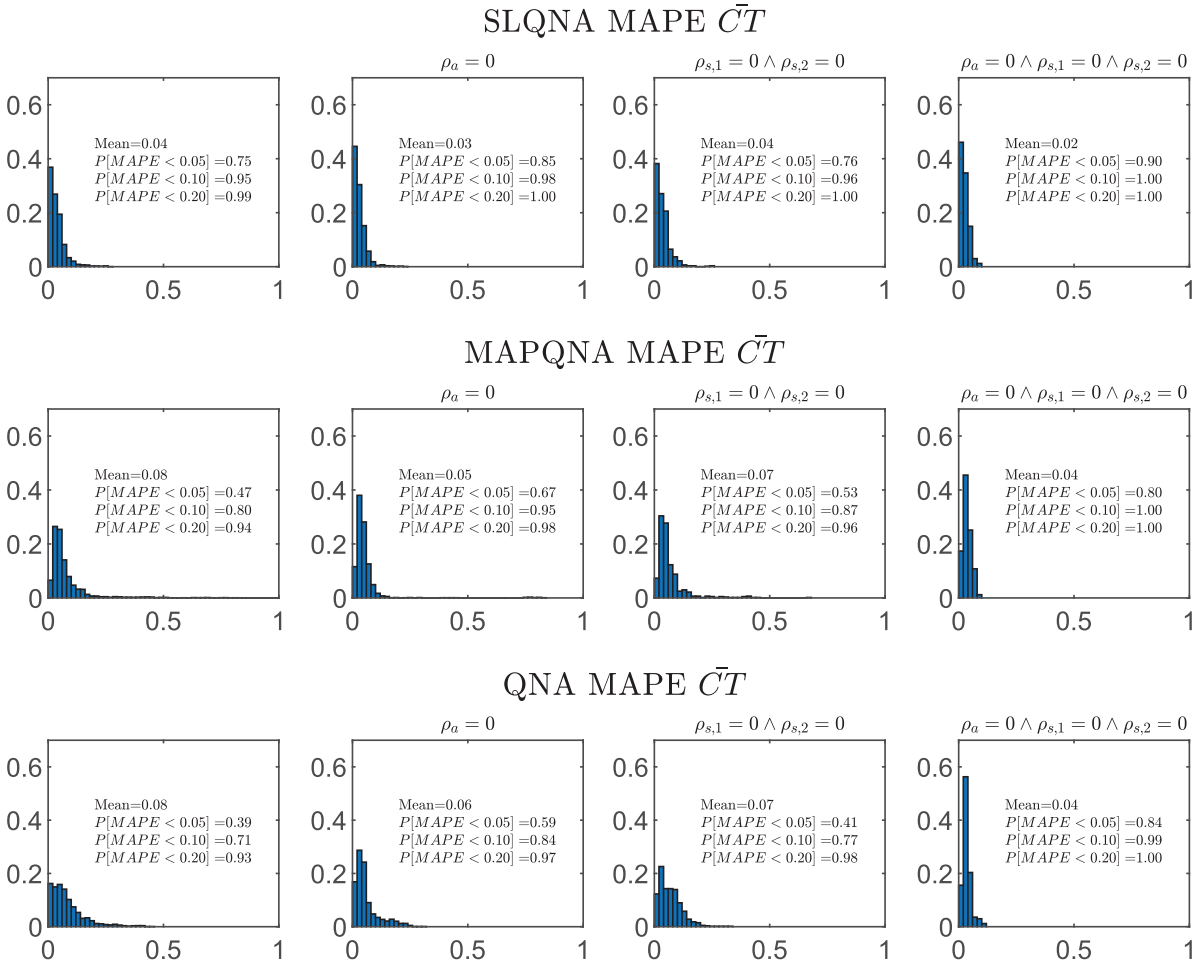

Figure 13. Histogram of the mean absolute percentage error (MAPE) of the cycle times predicted by using different methods for the network with split and merge.

SLQNA allows the analysis of new network structures without having to train supervised learning models for each structure. Our experiments show that SLQNA predicts the output stream mean, coefficient of variation and the first-lag autocorrelation at different points in a network accurately. SLQNA is compared to other approximation methods when these methods are used to predict the cycle time in production lines and also in a split-merge network. The results show that the accuracy of SLQNA is much better compared to the other available methods, and it is computationally very efficient. Furthermore, SLQNA with the trained functions allows obtaining the results in one second on a personal computer.

This work can be extended in several directions to evaluate the performance of manufacturing systems with different configurations. First, the functions that relate inputs to outputs will be developed for different building blocks. These new building blocks will include the building blocks for parallel stations, stations that process material with different sequencing and dispatching rules, and the building blocks with finite buffers. Next, a decomposition method that will connect these building blocks will be developed for configurations that involve loops.

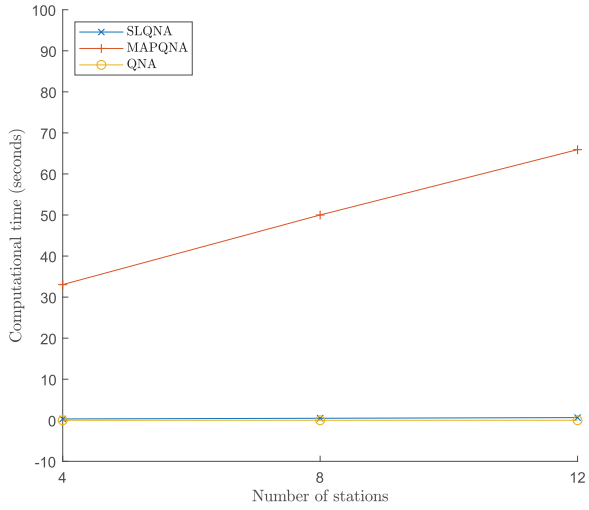

Figure 14. Effect of the size of the network with split and merge on the computational time of the methods.

Similarly, analysing queueing networks subject to blocking requires a different approach to combine the output characteristics of the building blocks. The approach presented in this study can also be used to predict the cycle time distribution. Finally, combining exact results that can be obtained for those systems that can be analysed analytically with the simulation results can speed up the time required to obtain the training set. These are left for future research.

Our results show that combining the power of supervised learning approaches with the analytical approaches yields a powerful tool to evaluate the performance of manufacturing systems accurately in a computationally efficient way. The regression-based approximation method presented in this study allows us to analyse single-server open queueing network models of manufacturing systems composed of delay, batching, split, and merge building blocks with correlated interarrival and service times accurately and efficiently.

Acknowledgments

Research leading to these results has received funding from the EU ECSEL Joint Undertaking under grant agreement no. 737459 (project Productive4.0) and from TUBITAK (217M145).

Funding

Research leading to these results has received funding from the EU Electronic Components and Systems for European Leadership (ECSEL) Joint Undertaking under grant agreement no. 737459 (project Productive4.0) and from Türkiye Bilimsel ve Teknolojik Araştirma Kurumu (TUBITAK) [217M145].

Notes on contributors

Barş Tan is a Professor of Operations Management and Industrial Engineering and the Vice President for Academic Affairs at Koç University, Istanbul, Turkey. His areas of expertise are in design and control of production systems, supply chain management, and stochastic modelling. 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.

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.

ORCID

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

References

Akhavan-Tabatabaei, R., S. Ding, and J. G. Shanthikumar. 2009. "A Method for Cycle Time Estimation of Semiconductor Manufacturing Toolsets With Correlations." In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, 1719–1729. IEEE.

Araghi, M., and B. Balcıoğlu. 2020. "Using Discrete Event Simulation to Fit Probability Distributions for Autocorrelated Service Times." *INFOR: Information Systems and Operational Research* 58 (1): 124–140.

Arinez, J. F., Q. Chang, R. X. Gao, C. Xu, and J. Zhang. 2020. "Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook." *ASME Journal of Manufacturing Science and Engineering* 142 (11): 110804. doi:10.1115/1.4047855.

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.

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

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.

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.

De Sousa Junior, W. T., J. A. B. Montevechi, R. de Carvalho Miranda, M. L. M. de Oliveira, and A. T. Campos. 2020.

"Shop Floor Simulation Optimization Using Machine Learning to Improve Parallel Metaheuristics." *Expert Systems with Applications* 150: 113272.
Geist, J. M. 1979. "Computer Generation of Correlated Gaussian Random Variables." *Proceedings of the IEEE* 67 (5): 862–863.
Harrison, J. M., and V. Nguyen. 1990. "The QNET Method for Two-moment Analysis of Open Queueing Networks." *Queueing Systems* 6 (1): 1–32.
Hopp, W. J., and M. L. Spearman. 2011. *Factory Physics*. Waveland Press.
Horng, S.-C., and S.-Y. Lin. 2013. "Evolutionary Algorithm Assisted by Surrogate Model in the Framework of Ordinal Optimization and Optimal Computing Budget Allocation." *Information Sciences*233: 214–229.
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.
Hung, Y.-F., and C.-B. Chang. 1999. "Using An Empirical Queueing Approach to Predict Future Flow Times." *Computers & Industrial Engineering* 37 (4): 809–821.
Jagerman, D. L., B. Balcıoglu, T. Altıok, and B. Melamed. 2004. "Mean Waiting Time Approximations in the G/G/1 Queue." *Queueing Systems* 46 (3–4): 481–506.
Kingman, J. 1961. "The Single Server Queue in Heavy Traffic." *Mathematical Proceedings of the Cambridge Philosophical Society* 57 (4): 902–904.
Krämer, W., and M. Langenbach-Belz. 1976. "Approximate Formulae for the Delay in the Queueing System GI/G/l." In *Congressbook, 8th ITC, Melbourne*, 235.1–235.8.
Kriege, J., and P. Buchholz. 2011. "Correlated Phase-Type Distributed Random Numbers As Input Models for Simulations." *Performance Evaluation* 68 (11): 1247–1260.
Kuehn, P. 1979. "Approximate Analysis of General Queuing Networks by Decomposition." *IEEE Transactions on Communications* 27 (1): 113–126.
Magnussen, S. 2004. "An Algorithm for Generating Positively Correlated Beta-distributed Random Variables with Known Marginal Distributions and a Specified Correlation." *Computational Statistics & Data Analysis* 46 (2): 397–406.
Manafzadeh Dizbin, N. 2020. "On Performance Evaluation and Optimal Control of Manufacturing Systems." PhD thesis, 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.
Marchal, W. G. 1976. "An Approximate Formula for Waiting Time in Single Server Queues." *AIIE Transactions* 8 (4): 473–474.
Marshall, K. T. 1968. "Some Inequalities in Queuing." *Operations Research* 16 (3): 651–668.
Mihoubi, B., B. Bouzouia, and M. Gaham. 2020. "Reactive Scheduling Approach for Solving a Realistic Flexible Job Shop Scheduling Problem." *International Journal of Production Research*. doi:10.1080/00207543.2020.1790686.
Novak, L. 1973. "Generating Correlated Weibull Random Variables for Digital Simulations." In *1973 IEEE Conference on Decision and Control including the 12th Symposium on Adaptive Processes*, 156–160. IEEE.
Papadopoulos, H., and C. Heavey. 1996. "Queueing Theory in Manufacturing Systems Analysis and Design: A Classification of Models for Production and Transfer Lines." *European Journal of Operational Research* 92 (1): 1–27.
Quiñonero-Candela, J., and C. E. Rasmussen. 2005. "A Unifying View of Sparse Approximate Gaussian Process Regression." *Journal of Machine Learning Research* 6 (Dec): 1939–1959.
Rasmussen, C. E., and C. K. Williams. 2006. *Gaussian Processes for Machine Learning*. Cambridge, MA: The MIT Press.
Shanthikumar, J. G., and R. G. Sargent. 1983. "A Unifying View of Hybrid Simulation/analytic Models and Modeling." *Operations Research* 31 (6): 1030–1052.
Telek, M., and G. Horváth. 2007. "A Minimal Representation of Markov Arrival Processes and a Moments Matching Method." *Performance Evaluation* 64 (9–12): 1153–1168.
Wang, B., and F. Xin. 2017. "Generation of Weibull Distribution Clutter Based on Correlated Gaussian Sequence." *Journal of Physics: Conference Series* 887 (1): 012059.
Whitt, W., and W. You. 2020. "A Robust Queueing Network Analyzer Based on Indices of Dispersion." Preprint, arXiv:2003.11174.
Yang, F. 2010. "Neural Network Metamodeling for Cycle Time-throughput Profiles in Manufacturing." *European Journal of Operational Research* 205 (1): 172–185.

Appendices

Appendix 1. Comparison of the Simulation of i.i.d. and correlated Weibull random sequences

Figure A1 gives the histogram and the autocorrelation function for a simulated i.i.d. stream and a simulated negatively correlated stream. As depicted here, the method given here is able to generate traces with the desired properties.

Appendix 2. Comparison of the single station cycle time prediction with the analytical approximations

The supervised learning approach yields more accurate cycle time prediction compared to analytical approximations (Kingman 1961; Marchal 1976; Krämer and Langenbach-Belz 1976; Buzacott and Shanthikumar 1993) and the robust queueing approximation (Whitt and You 2020).

Table A1 gives the mean absolute percentage errors of cycle times obtained by using approximations (Kingman 1961; Marchal 1976; Krämer and Langenbach-Belz 1976), three different approximations B&S 1, B&S 2, B&S 3 given in Buzacott and Shanthikumar (1993) and the error obtained by the single-station delay block of SLQNA. Table A2 gives the percentage errors obtained by these approximations and SLQNA for the cases with no autocorrelation, i.e. $\rho_s = \rho_a = 0$.

Appendix 3. Effect of approximating the Markov Chain Split with a Binomial Split

To motivate the modelling of a Markovian split process, we study an isolated Markovian split block with $p_1 = p_2$ and its binomial model. We have varied the parameters of the incoming flow and p_1, p_2 for this purpose. Figure A2 depicts the results of these experiments. These results show that ignoring

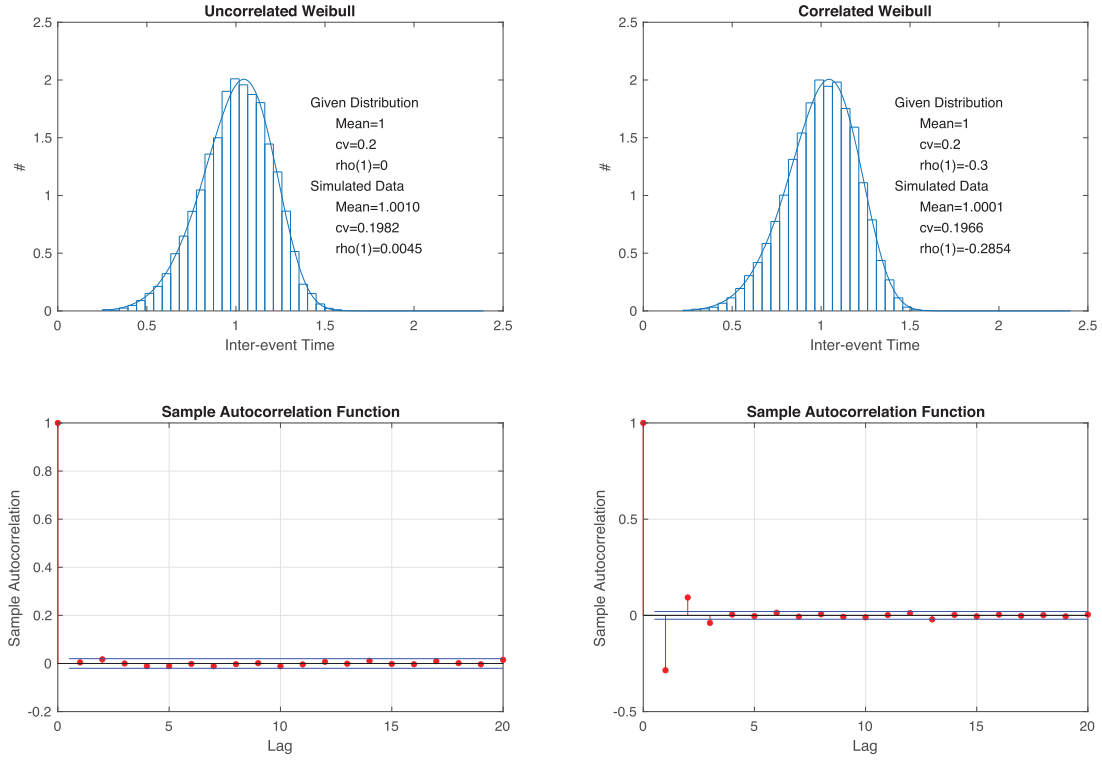

Figure A1. Histogram and autocorrelation function of the simulated uncorrelated and correlated Weibull random sequences for a specific case.

Table A1. Mean absolute % error (MAPE) of single-station *CT* approximations.

	cv_a															
Approx.	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	< 1	Avg
Kingman	13	13	12	12	11	11	11	11	12	13	14	16	16	18	12	13
Marchall	7	8	8	8	9	9	10	11	12	13	14	15	16	17	9	11
K&L	9	9	10	11	13	14	16	17	19	21	24	26	28	31	13	18
B&S 1	7	8	8	8	9	9	10	11	12	13	14	15	16	17	9	11
B&S 2	7	8	8	8	9	9	10	11	12	13	14	15	16	17	9	11
B&S 3	13	14	15	16	17	17	17	16	14	13	17	25	35	49	16	20
SLQNA	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1

Table A2. Mean absolute % error (MAPE) of single-station *CT* approximations ($\rho_s = \rho_a = 0$).

	cv_a															
Approx.	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	< 1	Avg
Kingman	11	10	10	9	8	6	5	3	2	1	2	3	4	5	71	6
Marchall	3	3	3	3	3	2	2	2	1	1	1	2	3	4	2	2
K&L	5	6	7	8	9	10	11	13	14	16	19	21	24	26	9	13
B&S 1	3	3	3	3	3	2	2	2	1	1	1	2	3	4	2	2
B&S 2	2	2	3	3	2	2	2	1	1	1	1	2	3	3	2	2
B&S 3	11	12	13	14	14	14	13	10	6	1	8	17	29	44	12	15
SLQNA	0	0	0	0	1	0	1	1	1	1	1	1	1	1	0	1

strong dependencies in the split process, i.e. higher p_1, p_2 values can result in considerable underestimation of output CV. This can be attributed to the fact that larger p_1, p_2 values would result in long periods with no arrivals in each of the downstream paths.

Appendix 4. Pseudo codes

In the following, we give a summary of the SLQNA, MAPQNA, and QNA algorithms used in this work. For details on the Rob-QNA algorithm for tree-structured networks, we refer the reader to Whitt and You (2020).

A.1 SLQNA, MAPQNA, and QNA

In the following, we give general pseudocode for the SLQNA, MAPQNA, and QNA methods. For analysing general network structures, we use the following representation for the networks. The network is defined by a set of nodes, and the

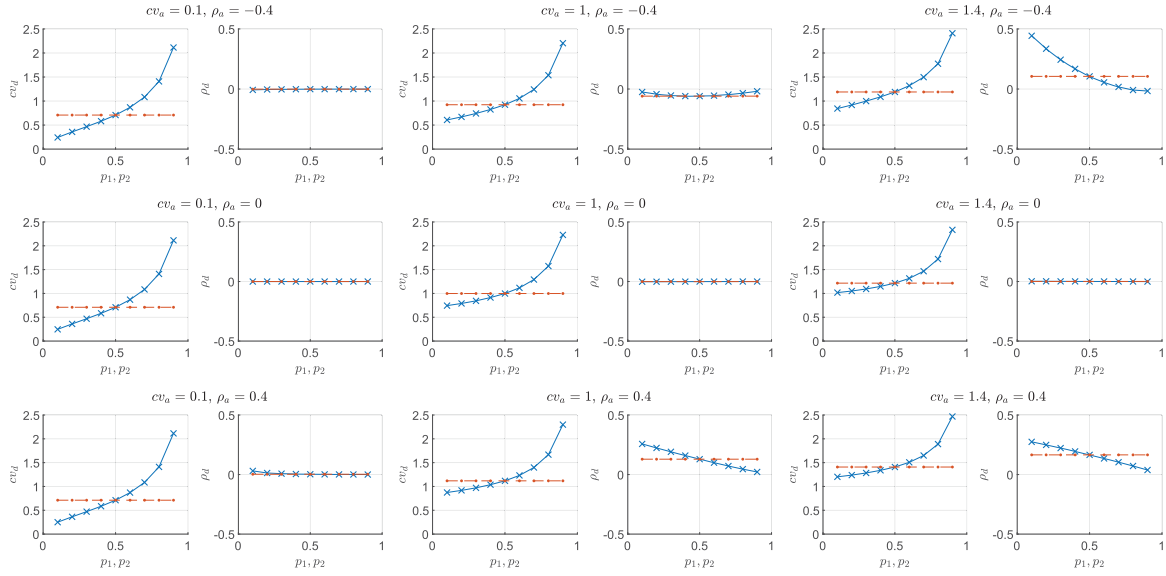

Figure A2. Effect of approximating a Markov chain split with a Binomial split when $p_1 = p_2$. The solid line depicts the output of the Markov chain split and the dashed line the output of its binomial model ($p = 0.5$).

probability matrix that governs the path of the parts in the system. Let $p_{i,j}$ denote the probability of a part moving to node j after leaving node i.

Let $\check{x}_i$ denote the value of characteristic x for the flow incoming to node i and $\hat{x}_i$ denote the value of characteristic x for the flow leaving node i and $\overrightarrow{x}_{i,j}$ value of the characteristic x for the flow from node i to node j, e.g. $\overrightarrow{\rho}_{i,j}$ being the first lag autocorrelation of the flow from node i to node j. In addition, let $\check{\epsilon}_i, \hat{\epsilon}_i, \overrightarrow{\epsilon}_{i,j} \in \{0, 1\}$ denote binary variables indicating if the characteristics of the corresponding flow/node is known/has been calculated.

Hence, a network can be specified by $\{\check{\mu}_i\}, \{\check{cv}_i\}, \{\check{\rho}_i\}, \{\mu\}, \{cv_i\}, \{\rho_i\}, \{p_{i,j}\}$ where the values related to incoming flows are only known for external arrivals. Without loss of generality, for notational convenience, we give the pseudo codes for network representations where the incoming flow to a node is either external or internal. For node i with external inflows, the characteristics of the incoming flows are given and $\check{\epsilon}_i = 1$ and for node j with internal inflows, $\check{\epsilon}_j = 0$.

For MAPQNA, a network can be specified by $\{\check{\mathbf{M}}\mathbf{0}_i\}, \{\check{\mathbf{M}}\mathbf{1}_i\}, \{\mathbf{M0}_i\}, \{\mathbf{M1}_i\}, \{p_{i,j}\}$. For QNA, a network can be specified by $\{\check{\mu}_i\}, \{\check{cv}_i\}, \{\mu\}, \{cv_i\}, \{p_{i,j}\}$.

Let $delay_{method}(\{\check{x}\}, \{x\}, y), split_{method}(\{\check{x}\}, p, y), merge_{method}(\{\check{x}\}, \{\check{x}\}, y)$ denote the block functions that give the characteristic y for the output of the three blocks of the queuing network analysis methods and let $sojourn_{method}(\{\check{x}\}, \{x\})$ denote the cycle time prediction function, where $method \in \{\text{SLQNA}, \text{MAPQNA}, \text{QNA}\}$. For SLQNA, $x \in \{\mu, cv, \rho\}$, for MAPQNA, $x \in \{\mathbf{M0}, \mathbf{M1}\}$ and for QNA, $x \in \{\mu, cv\}$. Algorithm 1 gives the generic pseudo code for these three methods. The Batching block is incorporated into the SLQNA algorithm in a way that is similar to the Delay block. The inputs of the Batching block include the batch size B instead of the characteristics of the service times for the Delay block. Similarly, for incorporating the MC split block, in addition to the characteristics of the input stream, p_1, p_2 are used as inputs.

A.2 Total cycle time calculation based on the cycle time for individual stations

For a production system with merge and split, the cycle time distribution from the source to the sink is calculated as the weighted summation of the cycle time distributions for different routes that a part can take. Let $r \in \mathcal{R}$ denote a route of length l_r where r_k denotes the kth node in the route r. The total cycle time of the system can be calculated as

$$CT = \sum_{r \in \mathcal{R}} \text{Prob}(r) \left(\sum_{k=1}^{l_r} CT_k \right), \tag{A1}$$

where

$$\text{Prob}(r) = \prod_{k=2}^{l_r} p_{r_{k-1}, r_k}, \tag{A2}$$

and CT_k denotes the cycle time prediction for station k.

Appendix 5. Effect of system parameters on the SLQNA accuracy

A.3 Effect of system parameters on the SLQNA accuracy for production lines

Effect of interarrival and Service Time Variability. Table A3 shows the effect of cv_a and cv_s on the accuracy of different methods in predicting CT. The results show that service time coefficient of variation affects the results more compared to the interarrival time variability. This is partly due to the experimental setup where all the stations are homogeneous. As a result, a change in cv_s affects all the production time processes while a change in cv_a affects the interarrival process to the first station. A decrease in the value of cv_s has a larger effect on decreasing the accuracy of MAPQNA. This can be attributed to the fact that representing processes with a lower coefficient of variation requires larger MAPs.

Table A3. The Effect of cv_a and cv_s on the accuracy of the methods in predicting the cycle time for the production line experiments.

	Mean absolute percentage error $100\frac{\lvert CT-\tilde{CT}\rvert}{\tilde{CT}}$									
	cv_a					cv_s				
	0.4	0.6	0.8	1	1.2	0.4	0.6	0.8	1	1.2
SLQNA	4.29	3.08	2.60	2.52	4.22	6.95	3.74	2.92	2.80	3.70
MAPQNA	7.49	5.53	5.69	2.93	7.60	10.83	8.88	8.31	5.11	5.26
Rob-QNA	17.24	14.34	10.79	9.17	11.61	5.31	7.97	10.23	13.98	11.76
QNA	19.76	15.22	11.23	9.13	9.56	5.83	6.22	9.29	13.36	11.39

Algorithm 1 Generic pseudo code for SLQNA, MAPQNA and QNA.

while $\exists i : \hat{\epsilon}_i = 0 \vee \check{\epsilon}_i = 0$ **do**
  for $1 \leq i \leq n$ **do** ▷ Determining the characteristics of the flows
    for $1 \leq j \leq n$ **do**
      if $p_{i,j} > 0 \wedge \overrightarrow{\epsilon}_{i,j} = 0 \wedge \hat{\epsilon}_i = 1$ **then**
        if $p_{i,j} = 1$ **then** $\overrightarrow{x}_{i,j} \leftarrow \hat{x}_i \forall x$ **end if**
        if $p_{i,j} < 1$ **then** $\overrightarrow{x}_{i,j} \leftarrow split_m(\{\hat{y}_i \forall y\}, p_{i,j}, x) \forall x$, $\overrightarrow{\epsilon}_{i,j} \leftarrow 1$ **end if** ▷ Split
      end if
    end for
  end for
  for $1 \leq i \leq n$ **do** ▷ Determining the characteristics of the incoming flows to nodes
    if $\check{\epsilon}_i = 0$ **then**
      $U = \{j : p_{j,i} > 0\}$ ▷ Identifying the immediate upstream stations
      if $\overrightarrow{\epsilon}_{u,i} = 1 \forall u \in U$ **then**
        if $|U| = 1$ **then** $\check{x}_i \leftarrow \overrightarrow{x}_{u \in U,i} \forall x$ **end if**
        if $|U| = 2$ **then** $\check{x}_i \leftarrow merge_m(\{\overrightarrow{y}_{u,i} \forall y\} \forall u \in U, x) \forall x$ **end if** ▷ Merge
        $\check{\epsilon}_i \leftarrow 1$
      end if
    end if
  end for
  for $1 \leq i \leq n$ **do** ▷ Determining the characteristics of the outgoing flows from the nodes
    if $\hat{\epsilon}_i = 0 \wedge \check{\epsilon}_i = 1$ **then** $\hat{x}_i \leftarrow delay_m(\{\check{y}_i \forall y\}, \{y_i \forall y\}, x) \forall x$, $\hat{\epsilon}_i \leftarrow 1$ ▷ Delay
    end if
  end for
end while
for $1 \leq i \leq n$ **do** ▷ Calculating the cycle times
  $CT_i \leftarrow sojourn_m(\{\check{x}_i \forall x\}, \{x_i \forall x\})$
end for

Effect of interarrival and Service Time Autocorrelations. Table 6 and Figure 10 reported the effect of the existence of interarrival time and service time dependency on the accuracy of the methods in predicting the cycle time in a production line. The presence of dependency in the system affects SLQNA less than other methods. MAPQNA also remains relatively accurate as the autocorrelation in the system increases. Rob-QNA remains accurate when the service times are independent. This is due to the fact that the Rob-QNA algorithm assumes independent service times.

Tables A4 and A5 show the effect of the magnitude of ρ_a and ρ_s on the accuracy of the cycle time predicts obtained by different methods. These results indicate that the presence of autocorrelation in the system decreases the accuracy of all the methods, but this effect is less pronounced for SLQNA and MAPQNA, e.g. $|\rho_s|$ increasing from 0 to 0.4 almost triples the inaccuracy of QNA but just less than doubles the inaccuracies of SLQNA and MAPQNA. Unlike the effects of ρ_s, the effect of ρ_a is less symmetrical for positive and negative values. Positive arrival autocorrelation affects SLQNA and MAPQNA more. This can be attributed to the zigzag pattern of the negative ρ_a values that effects the total autocorrelation content of the processes $|\sum_{i=1}^{\infty}(-0.4)^i| = 0.28 < |\sum_{i=1}^{\infty}(0.4)^i| = 0.66$.

Effect of Utilisation. Table A6 shows the effect of utilisation on the total Cycle Time MAPE obtained by using different methods. For SLQNA, QNA and Rob-QNA, higher utilisation values mostly imply lower accuracy. However, Rob-QNA and QNA are more sensitive to this aspect of the system. MAPQNA performs best for mid-level utilisation values and is less accurate for very high-traffic and very low-traffic systems.

Effect of Number of Stations. Table A7 shows the effect of the number of stations on the Cycle Time MAPE obtained by using different methods for the whole range of parameters. The results show that although the accuracy of QNA deteriorates as the number of stations increases, the accuracy of SLQNA does not get affected negatively with the increasing number of stations.

A.4 Effect of the system parameters on the SLQNA accuracy for the split-merge network

Table A8 shows that the accuracy of the approximation methods increase as the number of stations in the network increases and the split probability is varied between 0.2 and 0.7. The accuracy of the approximation methods are the worst when the split probability is 0.5. The errors obtained by using MAPQNA and QNA are more than twice the error obtained by SLQNA.

Table A9 shows that an increase in the coefficient of variation of the interarrival and service times improves the performance of MAPQNA. QNA under-performs in cases with higher values of autocorrelation.

Table A10 shows the effect of service rates that increase the utilisation of the stations. MAPQNA and QNA appear to be less accurate in high traffic. However, SLQNA is considerably less sensitive to the changes in utilisations in the system.

Table A4. Effect of ρ_a on the accuracy of the methods for predicting the cycle time in the production line experiments.

	Mean absolute percentage error $100\frac{\lvert CT-\tilde{CT}\rvert}{\tilde{CT}}$								
	ρ_a								
	−0.4	−0.3	−0.2	−0.1	0	0.1	0.2	0.3	0.4
SLQNA	2.71	3.00	2.55	2.72	2.91	3.34	3.94	4.82	5.62
MAPQNA	5.19	5.30	4.31	4.37	4.14	5.52	6.61	10.57	15.68
Rob-QNA	10.80	12.03	10.23	10.19	10.23	11.25	11.85	11.97	16.79
QNA	12.81	13.97	11.18	10.24	9.86	9.80	9.63	9.10	10.58

Table A5. Effect of ρ_s on the accuracy of the methods for predicting the cycle time in the production line experiments.

	Mean absolute percentage error $100\frac{\lvert CT-\tilde{CT}\rvert}{\tilde{CT}}$								
	ρ_s								
	−0.4	−0.3	−0.2	−0.1	0	0.1	0.2	0.3	0.4
SLQNA	5.75	3.53	3.01	2.88	2.75	2.88	2.91	2.97	3.33
MAPQNA	8.94	6.01	5.56	5.51	5.60	5.58	6.34	7.34	7.71
Rob-QNA	12.98	13.87	10.93	7.73	5.38	8.97	13.94	17.31	17.52
QNA	13.24	14.30	11.56	7.90	4.55	6.62	11.69	15.39	15.55

Table A6. Effect of utilisation on the accuracy of the methods for predicting the cycle time in the production line experiments.

	Mean absolute percentage error $100\frac{\lvert CT-\tilde{CT}\rvert}{\tilde{CT}}$								
	Utilisation								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
SLQNA	2.69	2.97	2.44	2.91	2.51	2.91	3.85	5.26	6.02
MAPQNA	10.42	7.43	4.57	3.76	3.73	4.59	6.21	7.34	11.08
Rob-QNA	3.62	4.28	5.87	7.88	9.80	11.60	15.70	21.37	28.55
QNA	2.32	3.47	4.62	6.64	9.21	12.06	16.04	21.48	26.84

Table A7. Effect of number of stations on the accuracy of the methods for predicting the cycle time in the production line experiments.

	Mean absolute percentage error $100\frac{\lvert CT-\tilde{CT}\rvert}{\tilde{CT}}$				
	Number of stations				
	5	10	15	20	25
SLQNA	3.67	3.54	3.29	3.23	3.17
MAPQNA	8.77	6.48	5.53	5.22	5.01
Rob-QNA	10.39	11.15	11.79	12.16	12.75
QNA	10.43	10.70	10.88	11.12	11.48

Table A8. Effect of the number of stations and the split probability on the accuracy of the methods for predicting the cycle time in the network with split and merge.

	Mean absolute percentage error $100\frac{\lvert CT-\tilde{CT}\rvert}{\tilde{CT}}$					
	Number of stations			p		
	4	8	12	0.2	0.5	0.7
SLQNA	4.40	3.47	2.87	3.10	5.01	2.99
MAPQNA	10.60	6.76	5.56	8.58	8.24	7.74
QNA	9.72	7.54	6.66	7.96	9.26	7.59

Table A9. Effect of the interarrival and service time coefficient of variation and autocorrelation on the accuracy of the methods for predicting the cycle time in the network with split and merge.

	Mean absolute percentage error $100\frac{\lvert CT-\bar{CT}\rvert}{\bar{CT}}$														
	cv_a		$cv_{s,1}$		$cv_{s,2}$		ρ_a			$\rho_{s,1}$			$\rho_{s,2}$		
	0.4	0.8	0.4	0.8	0.4	0.8	−0.3	0	0.3	−0.3	0	0.3	−0.3	0	0.3
SLQNA	6.01	3.24	5.41	2.95	4.50	3.17	4.69	2.94	2.97	3.69	3.61	3.97	3.64	3.61	4.02
MAPQNA	8.17	8.20	7.83	8.36	8.61	7.87	7.29	5.26	12.96	9.06	7.07	8.45	8.22	7.77	8.62
QNA	12.09	7.46	6.36	9.23	8.50	8.15	10.89	5.95	6.39	9.82	6.99	7.96	8.59	8.20	8.09

Table A10. Effect of the parameters related to the utilisation of the stations on the accuracy of the methods for predicting the cycle time in the network with split and merge.

	Mean absolute percentage error $100\frac{\lvert CT-\bar{CT}\rvert}{\bar{CT}}$					
	$\lambda_{s,1}$			$\lambda_{s,2}$		
	1.25	1.75	2.5	1.25	1.75	2.5
SLQNA	5.20	3.38	3.16	3.36	3.63	4.17
MAPQNA	17.47	6.50	3.98	8.25	8.31	8.03
QNA	15.04	7.29	5.06	8.49	8.27	8.18