State-Dependent Estimation of Delay Distributions in Fork-Join Networks

Nitzan Carmeli, Galit B. Yom-Tov
Technion—Israel Institute of Technology,
nitzany@campus.technion.ac.il, gality@technion.ac.il,

Onno J. Boxma
TU/e—Eindhoven University of Technology,
o.j.boxma@tue.nl

Problem Definition: Delay announcements have become an essential tool in service system operations: they influence customer behavior and network efficiency. Most current delay announcement methods are designed for relatively simple environments with a single service station or stations in tandem. However, complex service systems, such as healthcare systems, often have fork-join structures. Such systems usually suffer from long delays as a result of both resource scarcity and process synchronization, even when queues are fairly short. These systems may thus require more accurate delay estimation techniques than currently available.

Methodology/results: We analyze a network comprising a single-server queue followed by a two-station FJ structure using a recursive construction of the Laplace–Stieltjes transform of the joint delay distribution, conditioning on customers' movements in the network. Delay estimations are made at the time of arrival to the first station. Using data of an Emergency Department, we examine the accuracy and the robustness of the proposed approach, explore different model structures, and draw insights regarding the conditions under which the FJ structure should be explicitly modeled.

We provide evidence that the proposed methodology is better than other commonly used queueing theory estimators such as Last-to-Enter-Service (which is based on snapshot-principle arguments) and Queue-Length, and we replicate previous results showing that the most accurate estimations are obtained when using our model result as a feature in state-of-the-art machine learning estimation methods.

Managerial Implications: Our results allow management to implement individual, real-time, state-dependent delay announcements in complex FJ networks. We also provide rules of thumb with which one could decide whether to use a model with an explicit FJ structure or to reduce it to a simpler model requiring less computational effort.

Key words: Delay estimation; Fork-Join networks; Service systems; Healthcare operations; Queueing.

1. Introduction

Delay announcements have become an essential tool in service system operations, reducing the anxiety and uncertainty of waiting. They shorten perceived waiting time and increase customer satisfaction (Maister 1984, Munichor and Rafaeli 2007). More generally, it has been shown, both theoretically (Armony et al. 2009, Jouini et al. 2011, Pender et al. 2016) and empirically (Dong et al. 2019, Yu et al. 2016), that delay information influences customer behavior and, when wisely used, can improve system performance.

The creation of queues: scarce resources and synchronization requirements. Delay announcements rely on accurate delay estimations. Most methodologies for calculating delay estimations have been developed for single-station queues (Ibrahim and Whitt 2009a, 2009b, 2011a, 2011b) or for a predetermined route of queues in tandem (Ang et al. 2016, Gal et al. 2017), and capture delays stemming from a lack of resources. However, service networks additionally suffer from synchronization delays due to coordination needs in the network. Building methodologies to predict the combined effect of those two types of delay, which naturally emerge in fork-join (FJ) networks, is the essence of this paper. Terminology wise, from now on we shall refer to the *delay* in a station as the total sojourn time in the station.

An FJ structure models two (or more) activities that can be done in parallel by different resources, where both (or all) of these activities need to be completed in order to continue the process. Consider, for example, an emergency department (ED) patient who needs to have an X-ray taken and interpreted as well as a blood sample taken and tested at the lab before a physician can decide on the rest of the treatment. Those two processes, X-ray and blood test, can be (mostly) done in parallel; hence, their processing times may overlap. This example is illustrated in Figure 1 by the two yellow discs moving along the edges emanating from doctor attendance and leading to laboratory testing (Lab tests) and to X-Ray. Each colored disc represents a patient; discs of the same color represent the same patient. Thus, the two yellow discs represent a patient waiting for an X-ray while their blood specimens are being processed. FJ networks are prevalent in service systems, such as healthcare systems (as in the above example), the legal system (e.g., when a judge assigns several specialists to produce their opinion before ruling), and many other environments.

There are two main types of FJs—exchangeable and nonexchangeable. An exchangeable FJ is usually appropriate to describe assembly lines. For example, assume an assembly of two parts: A and B. In an exchangeable FJ, any type A part could be assembled (joined) with any type B part. In service systems, however, FJs are usually nonexchangeable, as “parts” in this case are usually two entities belonging to a customer. In our ED example, clearly, a patient’s blood test results must be matched with their own X-ray results and medical file.

Time-lagged information. It is often desirable to provide an estimation for the total sojourn time of a customer within a given service *network*. This usually entails estimation of delays in the different stations of the network, *at the time of arrival to the first station.* Many service systems, e.g., healthcare systems, are characterized by long sojourn times. In such systems, there may be substantial changes in the system state during a customer length of stay (LOS). Therefore, delay

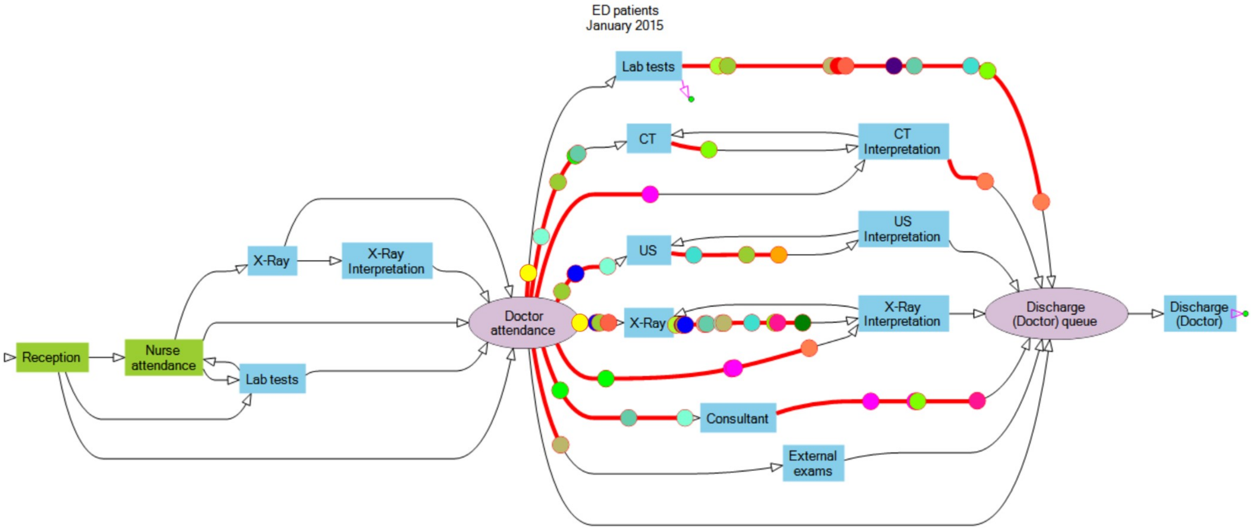

Figure 1 **A snapshot of patient flow in an Israeli ED. Nodes represent ED processes, and each colored disc moving along an edge represents a patient who completed the process at the edge origin, and is waiting for, or currently undergoing, the process at the edge destination. The time a disc spends on a specific edge is proportional to the patient sojourn time (waiting+service) in the edge destination activity. The red lines represent the relative progress of the patient in this activity, that is, the time already spent in this activity out of the total sojourn time.**
An animation of the data is available here: ED Flow.

estimation made when customers arrive at the first station for a service station down the customer path (i.e., estimation of a delay at some future point in time) may suffer from great inaccuracies stemming from time-lagged information (Dong et al. 2019, Pender et al. 2016)

What to announce? Most delay announcements, both in practice and in the queueing theory literature, are based on a single-point estimation, usually either the mean or the median. However, recent findings support the need to provide distributional information to customers, not just the mean/median. For example, Yom-Tov et al. (2017) showed that delay announcements given as a range (e.g., “your wait will be between 45 and 60 minutes”) influence customer patience differently than announcing expected wait or a lower bound, and that they help reduce customer abandonment. In addition, in a preliminary study of ED announcements, Efrat and Parush (2017) found that customers perceive such range-based delay announcements as more understandable and credible than those referring to a single point in time (“your wait will be approximately 45 minutes” or “your wait will be at least 45 minutes”).

In fact, the importance of estimating waiting-time distributions in order to provide delay announcements based on different distribution percentiles has already been discussed in Whitt

(1999) and Sun et al. (2012). In the latter, a quantile regression model was developed to provide delay estimations in an ED as a range. The authors claimed that, due to the great variability in patient waiting times (in this case, between triage and first encounter with a physician), it is extremely challenging to provide a single-point estimation (e.g., the mean waiting time), and that patients are more satisfied when they receive overestimations rather than underestimations.

1.1. Goals and main contributions

Motivated by real-life service systems, healthcare systems in particular, our ultimate goal is to provide accurate real-time estimations for delay distributions in service systems. Having said that, we note that service systems (usually conceptualized as queueing networks) in general can be extremely complex, and current research on delay estimations in such systems does not go beyond queues in tandem. Here, we do not attempt to provide a full solution to any given network, but rather take one important step in that direction by considering the interplay between delays driven by scarce resources and delays emerging from synchronization requirements. To this end, we consider a combination of tandem queues and an FJ structure, and we develop estimations for delay distributions in a system of one single-server queue which is followed by an FJ structure.

We note that an FJ structure could be only one part of a much more complex network. The computational effort required in order to produce delay estimators that account for the FJ structure(s) within a larger network may become too excessive as the network grows large. One of the methodologies to avoid this “curse of dimensionality” is to “fold” detailed structures, such as FJs, into simpler, less detailed building blocks (Senderovich et al. 2016). Thus, an additional goal is to provide modeling guidelines that consider the trade-off between estimation accuracy and computational complexity when modeling queues with FJ structures.

The main contributions of this paper are the following:

1. Developing delay estimators for service *networks* that include *FJ* structures, focusing on *individual, real-time, state-dependent* estimation as opposed to the average, steady-state sojourn time of all customers. We start by developing formulas for such delay estimators under the assumption that service times are exponential (§2). We then suggest extensions for service times having a coefficient of variation (CV) different than 1 (§EC.3).

2. Establishing rules of thumb under which the FJ structure should be explicitly modeled versus situations in which it could be simplified into structures whose delay estimations require less computational effort.

3. Implementing the method developed in 1. in a case study of an ED and showing that accounting for synchronization queues arising from an underlying FJ structure indeed significantly improves estimation accuracy.

4. Replicating previous results (Ang et al. 2016, Gal et al. 2017) and showing that enriching state-of-the-art machine learning techniques with queueing theory features (such as the result of our model) provides the most accurate estimations. This was done using a unique transaction-level data base of patients' paths within an Israeli ED.

1.2. Literature review

There is ample literature on delay estimations in service systems. An extensive survey of (mostly) steady-state results of exact delay expressions in queueing systems is given by Boxma and Daduna (1990). Another comprehensive survey, which adds coverage of real-time (state-dependent) results and the influence of delay information on customer behavior and system operations, is provided by Ibrahim (2018). Surveys on sojourn times in FJ networks are given in Boxma et al. (1994), Thomasian (2014).

Sojourn times in FJ networks: To the best of our knowledge, current results on sojourn times in FJ networks provide mostly approximations or bounds (Baccelli et al. 1989, Nelson and Tantawi 1988). Such approximations or bounds for the complete sojourn-time *distribution* are given, for example, in Ko and Serfozo (2004, 2008) and Rizk et al. (2015). Qiu et al. (2015) propose a phase-type representation of the sojourn-time distribution in an FJ network. In Zhang (1990), a closed-form solution for the joint Laplace–Stieltjes transform (LST) of steady-state waiting times in an FJ station with two queues is provided. Kim and Agrawala (1989) obtained the joint sojourn-time distribution in each single-server station of an FJ queue, starting with an empty system and computing it recursively at arrival epochs of new customers. The authors considered service durations having an Erlang distribution and suggested an approximation for the joint steady-state sojourn-time distribution by truncating the state space, assuming that the probability of having more than M customers in each of the queues approaches zero. We also suggest an *exact* recursive approach to derive state-dependent estimations of sojourn-time distributions in the FJ part of a network; however, we provide these estimations at the time of a customer arrival to a *preceding* service station, and provide an elegant, simple method for the recursive computation using matrix multiplication.

Snapshot-principle-based estimators in queueing networks: Reiman (1982) studied conventional heavy-traffic diffusion approximations for sojourn times in Jackson networks with K single-server stations. One of his main results is the *snapshot principle*: "It is as if the customer takes a snapshot of the network when he enters and all queues remain at the same value during the customer's sojourn through the network." Snapshot-principle-based estimators follow the assumption that changes in queue lengths are negligible during a customer visit to the system; this gave rise to Last-to-Enter-Service (LES) and Head-of-the-Line (HOL) estimators, which basically approximate the waiting time of an arriving customer by the waiting time of the last customer to enter service (LES), or the elapsed waiting time of the customer currently at the head of the queue (HOL). Huang et al. (2015), for example, used the snapshot principle to approximate queue length, virtual waiting time, and sojourn time in a single-server multiclass queue in which an ED physician treats both newly arrived patients and in-process patients. Nguyen (1993) studied heavy-traffic limits of FJ networks and showed that the total sojourn time of an arriving customer can be regarded as a function of the longest path. Using a large data set of patients' transactions within an Israeli ED, we compare our methodology's estimation results with those of snapshot-principle-based estimators. Noting that delays in EDs are usually long, and that the system state may change significantly during a patient LOS, we show that our methodology indeed substantially outperforms the snapshot-principle-based estimators.

Real-time, single-station, queueing theory-based estimations: Ibrahim and Whitt (2009a, 2009b, 2011a, 2011b) proposed a series of *real-time* waiting-time estimators for a single queue with abandonments, time-varying arrival rate and capacity, general service durations, and general inter-arrival and patience distributions. Their predictors can be categorized into two types: queue-length-based (QL) predictors, and delay-history-based predictors (or snapshot-principle-based predictors). Thiongane et al. (2016) expanded the delay-history-based predictors to a weighted average form that can cope with multiclass queues. Although these real-time predictors cover a wide range of service systems, they only account for single stations and provide only an estimation for the average waiting time. Nakibly (2002) focused on estimating waiting-time distributions, given the system state at the time of arrival, but this was again for single-station service systems, albeit considering skills-based routing, motivated by call centers.

Enriching machine learning methods with queueing theory-based features: Broadly speaking, there are two main types of delay estimators: queueing theory-based (QT) estimators (as specified above), and machine learning (ML) estimators. The latter are usually based on ample data logs with many features, while the QT estimators are usually based only on the system state (such as the number of customers in queue, the processing rates, or the number of service providers). Thus, it is not surprising that ML estimators often outperform QT estimators (e.g., Mourão et al., 2017, Sanit-in and Saikaew, 2019, Thiongane et al., 2020). In recent years, there is growing support for the superiority of estimators that are a *combination* of ML and QT estimators, that is, ML estimators that incorporate the results of QT estimators as explanatory variables or features. The value of this approach has been shown for wait time estimations in call centers (Senderovich et al. 2014, Senderovich et al. 2015), in emergency departments (EDs) (Ang et al. 2016, Benevento et al. 2019, Sun et al. 2012), and even for predicting bus travel times given a scheduled bus journey (Gal et al. 2017). Our work provides further support for this approach.

Exact analysis of two queues in tandem: From a performance analysis perspective, our work is most closely related to the literature using exact analysis on open- and closed-service networks of two tandem queues. Stadje (1996) derived a closed-form solution of the total sojourn time within a closed network of two single-server queues in tandem with exponential service durations, *given the number of customers in each queue at the time of arrival*. Boxma (1983), Daduna (1986), and Boxma and Daduna (2014) yielded expressions for the joint LST of the sojourn time in the two tandem queues. Boxma (1983) considered the case of one queue with exponential service times followed by a queue with general service times in a closed network. It was shown that reversing the order of the queues yields a different expression for the joint LST. Daduna (1986) generalized the service time distribution to an Erlang mixture, and Boxma and Daduna (2014) derived an expression for the joint LST of the sojourn times in both queues, using first general service times and then exponential service times (for both the closed and open network). We follow that approach and extend it to FJ queues, adding concurrency, which is prevalent in many service systems (e.g., healthcare systems), and thus providing an important additional layer to the analysis.

1.3. Organization

The rest of this paper is organized as follows. In §2, we present our basic model of a network with one single-server station followed by an FJ structure. In §3, we use an exact analysis approach to compute the LST of the sojourn time in each station, and the LST of the total sojourn time, given the real-time system state at the time of arrival to the first station. In §4, we explore three alternative models and discuss the settings under which each should be considered, accounting for the trade-off between computational complexity and estimation accuracy. We present a case study of an Israeli ED and evaluate our model's performance compared to other QT and ML estimators in §5. We use our data-based example to numerically evaluate the influence of ignoring the FJ structure (i.e., the synchronization queues) on the estimation accuracy, and to support our model's robustness to deviations from the model's assumptions. We conclude and discuss future research directions in §6.

2. The fork-join network model

We consider an open queueing network (see Figure 2) with two parts. The first consists of a single-server queue Q_0, and the second consists of two queues in an FJ structure. We refer to the queues in the second part as the upper FJ station (UFJ) and the lower FJ station (LFJ). We assume that service times in the first part are exponentially distributed with rate μ_0. This assumption can be relaxed; an analysis of general service times at the first station is given in §EC.2.

The FJ stations can be either single-, multiple-, or infinite-server queues having exponential service times with rates μ_1 and μ_2 for the upper and lower stations, respectively. We assume an FCFS service policy throughout the network. From here on, we shall assume both stations in the FJ part have a single server, but this is only to simplify notation and is not a constraint in any way.

We use the following notation to describe the system state at the time of arrival of a *tagged* customer (to whom we would like to provide the delay announcement): k is the number of customers in the first single-server station (including customers in queue and in service), h_1 is the number of customers in the UFJ station, and h_2 is the number of customers in the LFJ station. All customers first receive service in Q_0 and then simultaneously join UFJ and LFJ, requiring independent services at both stations. Let S_0 be the tagged customer's sojourn time in the first station, and S_1 (S_2) be their sojourn time in the UFJ (LFJ) station. Let S_{FJ} be the tagged customer's total sojourn time in the FJ part. See Figure 2 for illustration. Because of the FJ structure, the tagged customer leaves when their sojourn time at both the UFJ and LFJ stations ends: $S_{FJ} = \max\{S_1, S_2\}$. Since we are assuming an FCFS policy, and Q_0 has a single server, customers entering the system after the tagged customer will not influence S_0 or S_{FJ}. It can therefore be assumed that no customers enter the system after the tagged customer, whose total sojourn time can thus be viewed as the time it will take the system to drain.

Our goal is to determine $\Psi_{k,h_1,h_2}(w_0, w_{FJ}) \equiv \mathbb{E}_{k,h_1,h_2}\left[e^{-w_0 S_0 - w_{FJ} S_{FJ}}\right]$, the joint Laplace–Stieltjes transform of S_0 and S_{FJ}, given that, at the arrival epoch of the tagged customer in Q_0, there are k customers in the first station, h_1 customers in the UFJ station, and h_2 customers in the LFJ station. We proceed as follows: In the present section, we derive a recursion for $\Psi_{k,h_1,h_2}(w_0, w_{FJ})$ in k, expressing this LST in several terms $\Psi_{k-1,\cdot,\cdot}(w_0, w_{FJ})$. After k steps, we thus reach $\Psi_{0,\cdot,\cdot}(w_0, w_{FJ})$—and the latter terms turn out to be easy to obtain. In the next section, we solve that recursion.

We obtain such a recursion in k for $\Psi_{k,h_1,h_2}(w_0, w_{FJ})$ by conditioning on the remaining service time of the customer in service in Q_0 at the arrival epoch of the tagged customer in that queue. We

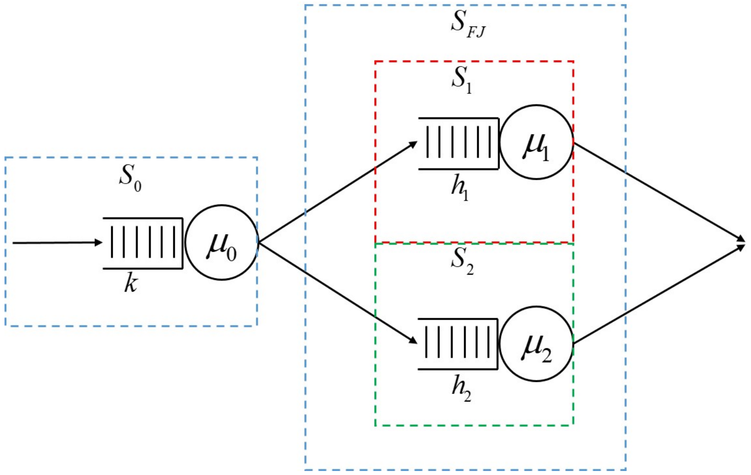

Figure 2 **Two queues in tandem: a single-server queue followed by a fork-join queue.**

restrict ourselves for the moment to the case where, at this arrival epoch, we have $k, h_1, h_2 > 0$. The baseline case $k = 0$ will be treated in the next section; there, we also show that the cases $h_1 = 0$ and $h_2 = 0$ follow from the same analysis.

If the tagged customer arrives and finds customers in all three stations, that is, $k, h_1, h_2 > 0$, then there are four cases to be distinguished for the events in UFJ and LFJ during the remaining service time t of the customer in service in Q_0.
Case 1: Out of the (h_1, h_2) customers at (UFJ, LFJ), $r_1 < h_1$ and $r_2 < h_2$ are served in $[0, t]$.
Case 2: Out of the (h_1, h_2) customers at (UFJ, LFJ), $r_1 < h_1$ and all h_2 are served in $[0, t]$.
Case 3: Out of the (h_1, h_2) customers at (UFJ, LFJ), all h_1 and $r_2 < h_2$ are served in $[0, t]$.
Case 4: All of the (h_1, h_2) customers at (UFJ, LFJ) are served in $[0, t]$.
Observe that the probability of r_1 services in $[0, t]$ equals $e^{-\mu_1 t}\frac{(\mu_1 t)^{r_1}}{r_1!}$ when $r_1 < h_1$, and that the probability of serving all h_1 customers in $[0, t]$ equals $\sum\limits_{r_1=h_1}^{\infty} e^{-\mu_1 t}\frac{(\mu_1 t)^{r_1}}{r_1!}$. Observe, furthermore, that the remaining service time of the customer in service in Q_0 has density $\mu_0 e^{-\mu_0 t}$. Hence, we obtain the following key recursion:

$$\Psi_{k,h_1,h_2}(w_0, w_{FJ}) \equiv \mathbb{E}_{k,h_1,h_2}\left[e^{-w_0 S_0 - w_{FJ} S_{FJ}}\right] = \int\limits_0^\infty e^{-w_0 t} X_{k,h_1,h_2} \mu_0 e^{-\mu_0 t}\,\mathrm{d}t, \tag{1}$$

where X_{k,h_1,h_2} is given by

$$\begin{aligned} X_{k,h_1,h_2} = &\sum_{r_1=0}^{h_1-1}\sum_{r_2=0}^{h_2-1} e^{-\mu_1 t}\frac{(\mu_1 t)^{r_1}}{r_1!} e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!}\Psi_{k-1,h_1-r_1+1,h_2-r_2+1}(w_0, w_{FJ}) \\ &+ \sum_{r_1=0}^{h_1-1}\sum_{r_2=h_2}^{\infty} e^{-\mu_1 t}\frac{(\mu_1 t)^{r_1}}{r_1!} e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!}\Psi_{k-1,h_1-r_1+1,1}(w_0, w_{FJ}) \end{aligned}$$

$$+\sum_{r_1=h_1}^{\infty}\sum_{r_2=0}^{h_2-1} e^{-\mu_1 t}\frac{(\mu_1 t)^{r_1}}{r_1!}e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!}\Psi_{k-1,1,h_2-r_2+1}(w_0,w_{FJ})$$
$$+\sum_{r_1=h_1}^{\infty}\sum_{r_2=h_2}^{\infty} e^{-\mu_1 t}\frac{(\mu_1 t)^{r_1}}{r_1!}e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!}\Psi_{k-1,1,1}(w_0,w_{FJ}). \tag{2}$$

The four terms on the right-hand side of (2) correspond to the four cases mentioned above. In the first term, $r_1 < h_1$ customers are served at UFJ, leaving $h_1 - r_1$ customers behind; the customer whose service in Q_0 ends at time t then joins UFJ, bringing its number of customers to $h_1 - r_1 + 1$. That customer also joins LFJ, making the total number of customers there $h_2 - r_2 + 1$. Because of the memoryless property of the exponential distribution, we can, for the LST of the remaining sojourn times of the tagged customer in the first part and the second part, proceed as if the tagged customer *now* arrives in Q_0, and finds $k-1$ customers in that queue and $(h_1 - r_1 + 1, h_2 - r_2 + 1)$ customers at (UFJ,LFJ)—without having to worry about the lengths of remaining service times at UFJ and LFJ; they are still exponentially distributed. That explains the occurrence of $\Psi_{k-1,h_1-r_1+1,h_2-r_2+1}(w_0,w_{FJ})$ in the first term on the right-hand side of (2). In the second term, all h_2 customers at LFJ have been served before time t, leaving LFJ empty; the arrival of the customer who was just served in Q_0 brings the number of customers at LFJ back to one, so that we get $\Psi_{k-1,h_1-r_1+1,1}(w_0,w_{FJ})$. The third and fourth terms are similarly explained.

3. Laplace–Stieltjes transform computation

Based on Equations (1) and (2), the joint LST can now be computed recursively. Each iteration corresponds to the number of remaining customers at the first station, until it has no customers except for the tagged customer. That is, in the first iteration, there are k customers ahead of the tagged customer in Q_0; in the next iteration, there will be $k-1$, and so on. The joint LST in each iteration is conditioned on the number of potential service completions in the FJ part by the time the current customer in service in Q_0 leaves the first station. In order to concisely express the recursion for the joint LST, we use the following notation:

$$a(r_1,r_2) \equiv \int_0^\infty e^{-(w_0+\mu_1+\mu_2)t}\frac{(\mu_1 t)^{r_1}}{r_1!}\frac{(\mu_2 t)^{r_2}}{r_2!}\mu_0 e^{-\mu_0 t}\,\mathrm{d}t,$$
$$b(r_1,h_2) \equiv \sum_{r_2=h_2}^{\infty} a(r_1,r_2) = \int_0^\infty e^{-(w_0+\mu_1+\mu_2)t}\sum_{r_2=h_2}^{\infty}\frac{(\mu_1 t)^{r_1}}{r_1!}\frac{(\mu_2 t)^{r_2}}{r_2!}\mu_0 e^{-\mu_0 t}\,\mathrm{d}t,$$
$$c(h_1,r_2) \equiv \sum_{r_1=h_1}^{\infty} a(r_1,r_2),$$

$$d(h_1,h_2) \equiv \sum_{r_1=h_1}^{\infty}\sum_{r_2=h_2}^{\infty} a(r_1,r_2).$$

Using the above notation, and a shorthand notation in which we omit w_0 and w_{FJ}, the recursive formula for $\Psi_{k,h_1,h_2}(w_0,w_{FJ}) \equiv \Psi_{k,h_1,h_2}$ given by Equations (1) and (2) (for the case where $k,h_1,h_2>0$) becomes

$$\begin{aligned}\Psi_{k,h_1,h_2} = &\sum_{r_1=0}^{h_1-1}\sum_{r_2=0}^{h_2-1}\Psi_{k-1,h_1-r_1+1,h_2-r_2+1}\cdot a(r_1,r_2) + \sum_{r_1=0}^{h_1-1}\Psi_{k-1,h_1-r_1+1,1}\cdot b(r_1,h_2)\\ &+\sum_{r_2=0}^{h_2-1}\Psi_{k-1,1,h_2-r_2+1}\cdot c(h_1,r_2) + \Psi_{k-1,1,1}\cdot d(h_1,h_2).\end{aligned} \tag{3}$$

Note that, if $h_1=0$ and/or $h_2=0$, (3) remains valid when an empty sum is defined to be zero. Taking $k=0$ requires more attention; the case $k=0$ will be treated at the end of the section. By simple algebraic manipulations we get

$$\begin{aligned}a(r_1,r_2) &= \frac{\mu_0}{\mu_0+w_0}\int_0^\infty (\mu_0+w_0)e^{-(\mu_0+w_0)t}e^{-\mu_1 t}\frac{(\mu_1 t)^{r_1}}{r_1!}e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!}\,\mathrm{d}t \qquad (4)\\ &= \binom{r_1+r_2}{r_1}\frac{\mu_0}{\mu_0+w_0+\mu_1+\mu_2}\left(\frac{\mu_1}{\mu_0+w_0+\mu_1+\mu_2}\right)^{r_1}\left(\frac{\mu_2}{\mu_0+w_0+\mu_1+\mu_2}\right)^{r_2},\\ b(r_1,h_2) &= \frac{\mu_0}{\mu_0+w_0}\int_0^\infty (\mu_0+w_0)e^{-(\mu_0+w_0)t}e^{-\mu_1 t}\frac{(\mu_1 t)^{r_1}}{r_1!}\left(1-\sum_{r_2=0}^{h_2-1}e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!}\right)\mathrm{d}t,\\ c(h_1,r_2) &= \frac{\mu_0}{\mu_0+w_0}\int_0^\infty (\mu_0+w_0)e^{-(\mu_0+w_0)t}e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!}\left(1-\sum_{r_1=0}^{h_1-1}e^{-\mu_1 t}\frac{(\mu_1 t)^{r_1}}{r_1!}\right)\mathrm{d}t,\\ d(h_1,h_2) &= \frac{\mu_0}{\mu_0+w_0}\int_0^\infty (\mu_0+w_0)e^{-(\mu_0+w_0)t}\left(1-\sum_{r_1=0}^{h_1-1}e^{-\mu_1 t}\frac{(\mu_1 t)^{r_1}}{r_1!}\right)\left(1-\sum_{r_2=0}^{h_2-1}e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!}\right)\mathrm{d}t.\end{aligned}$$

These three integrals can also be evaluated in the same way as the integral expression for $a(r_1,r_2)$. Furthermore, note that if we define Y_1 to be a Poisson process with rate μ_1, Y_2 as a Poisson process with rate μ_2, and $\Omega \sim exp(\mu_0+w_0)$, then

$$\begin{aligned}a(r_1,r_2) &= \frac{\mu_0}{\mu_0+w_0}\mathbb{P}\left(Y_1(\Omega)=r_1,\ Y_2(\Omega)=r_2\right),\\ b(r_1,h_2) &= \frac{\mu_0}{\mu_0+w_0}\mathbb{P}\left(Y_1(\Omega)=r_1,\ Y_2(\Omega)\geq h_2\right),\\ c(h_1,r_2) &= \frac{\mu_0}{\mu_0+w_0}\mathbb{P}\left(Y_1(\Omega)\geq h_1,\ Y_2(\Omega)=r_2\right),\\ d(h_1,h_2) &= \frac{\mu_0}{\mu_0+w_0}\mathbb{P}\left(Y_1(\Omega)\geq h_1,\ Y_2(\Omega)\geq h_2\right).\end{aligned}$$

3.1. Boundary conditions

The recursive computation of the joint LST ends when there are no customers at the first station except for the tagged customer, that is, when $k=0$. In this case, the tagged customer's sojourn time at the first station will only be their service time, which is exponential with rate μ_0. We define $Z_{x,y}(w_{FJ})$ as the LST of the total sojourn time of the tagged customer in the FJ part, given that there are x customers at the UFJ station and y customers at the LFJ station (including the tagged customer), as follows:

$$Z_{x,y}(w_{FJ}) := E\left[e^{-w_{FJ}S_{FJ}} \mid h_1=x, h_2=y\right] = \int\limits_0^\infty e^{-w_{FJ}t}\,\mathrm{d}F_{S_{FJ}|x,y}(t). \tag{5}$$

The sojourn time of a tagged customer arriving at the UFJ (LFJ) station and finding $x-1$ $(y-1)$ customers there will be the sum of x (y) service durations, each exponentially distributed with rate μ_1 (μ_2). Therefore, $(S_1|h_1=x) \sim Gamma\,(x,\mu_1)$, $(S_2|h_2=y) \sim Gamma\,(y,\mu_2)$, and from the gamma-Poisson relationship, we get

$$\begin{aligned}
F_{S_{FJ}|x,y}(t) &= \mathbb{P}\left(S_{FJ} \le t \mid h_1=x, h_2=y\right) = \mathbb{P}\left(S_1, S_2 \le t \mid h_1=x, h_2=y\right)\\
&= \mathbb{P}\left(S_1 \le t \mid h_1=x, h_2=y\right)\mathbb{P}\left(S_2 \le t \mid h_1=x, h_2=y\right)\\
&= \mathbb{P}\left(S_1 \le t \mid h_1=x\right)\mathbb{P}\left(S_2 \le t \mid h_2=y\right)\\
&= \left(1-\sum_{m=0}^{x-1} e^{-\mu_1 t}\frac{(\mu_1 t)^m}{m!}\right)\left(1-\sum_{n=0}^{y-1} e^{-\mu_2 t}\frac{(\mu_2 t)^n}{n!}\right).
\end{aligned}$$

It follows that once $k=0$, the recursive formula is done, since it is only left to calculate $Z_{x,y}(w_{FJ})$ for all possible values of x and y (which is a straightforward Laplace–Stieltjes calculation of $F_{S_{FJ}|xy}(t)$) and the probabilities of having $h_1=x$ and $h_2=y$ by the time the tagged customer completes service at the first station.

We shall now review the $k=0$ case. The resulting recursive formula (again, we omit w_{FJ} for notational simplicity) becomes

$$\begin{aligned}
\Psi_{0,h_1,h_2} = &\sum_{r_1=0}^{h_1-1}\sum_{r_2=0}^{h_2-1} Z_{h_1-r_1+1,h_2-r_2+1}\cdot a(r_1,r_2) + \sum_{r_1=0}^{h_1-1} Z_{h_1-r_1+1,1}\cdot b(r_1,h_2)\\
&+\sum_{r_2=0}^{h_2-1} Z_{1,h_2-r_2+1}\cdot c(h_1,r_2) + Z_{1,1}\cdot d(h_1,h_2).
\end{aligned} \tag{6}$$

Here, the LST $Z_{x,y} \equiv Z_{x,y}(w_{FJ})$ is defined as in Equation (5), and $a(\cdot,\cdot)$, $b(\cdot,\cdot)$, $c(\cdot,\cdot)$, and $d(\cdot,\cdot)$ are defined as in Equation (4). In this formula, an empty sum ($h_1 = 0$ and/or $h_2 = 0$) is again defined to be zero. Note that the term with $h_1 = h_2 = 0$ becomes

$$\Psi_{0,0,0} = Z_{1,1} d(0,0) = \frac{\mu_0}{\mu_0 + w_0} Z_{1,1}. \tag{7}$$

Note furthermore that $(S_{FJ}|1,1)$ is the maximum of two exponential random variables with rates μ_1 and μ_2. Therefore,

$$F_{S_{FJ}|1,1}(t) = \mathbb{P}(S_1 \le t \mid h_1 = 1)\,\mathbb{P}(S_2 \le t \mid h_2 = 1) = \left(1 - e^{-\mu_1 t}\right)\left(1 - e^{-\mu_2 t}\right)$$

and

$$\begin{aligned} Z_{1,1} &= \int_0^\infty e^{-w_{FJ} t}\left[\mu_1 e^{-\mu_1 t}\left(1 - e^{-\mu_2 t}\right) + \mu_2 e^{-\mu_2 t}\left(1 - e^{-\mu_1 t}\right)\right] \mathrm{d}t \\ &= \frac{\mu_1}{\mu_1 + w_{FJ}} - \frac{\mu_1}{\mu_1 + \mu_2 + w_{FJ}} + \frac{\mu_2}{\mu_2 + w_{FJ}} - \frac{\mu_2}{\mu_1 + \mu_2 + w_{FJ}}. \end{aligned}$$

The recursive formula can be easily and efficiently calculated if written in compact matrix form. Details are provided in EC.1. Assigning $w_{FJ} = 0$ will yield the LST of the sojourn time in the first single-server station, assigning $w_0 = 0$ will yield the LST of the sojourn time in the FJ part, and assigning $w_{FJ} = w_0$ will yield the LST of the tagged customer's total sojourn time in the system. From the LST, we obtain the Fourier–Stieltjes transform and then use the numerical inversion of Witkovský (2016) to retrieve the sojourn-time distribution.

3.2. General service durations

An extension to general service durations in the first station is provided in EC.2. The case of completely general service durations in the FJ part is not a trivial extension of our model, and we leave it for future research. However, one can capture the essence of most distributions via its first two moments, or equivalently by its expected value and the coefficient of variation (CV). We therefore extend our model to also include Erlang and hyperexponential service durations in the FJ part of the network. The Erlang distribution can be used when service durations in one of the FJ stations (or in both of them) exhibit a CV that is less than 1, while the hyperexponential distribution can be used when the CV is greater than 1. Details are provided in EC.3.

4. Guidelines for modeling: complexity vs. accuracy

In §2, we presented an exact analysis approach to estimating delays in a network with one single-server station followed by an FJ part comprising an upper station (UFJ) and a lower station (LFJ). Naturally, this three-station queueing system may be one building block in a larger queueing network. Our exact iterative approach allows for estimating delay distributions. However, it may face the "curse of dimensionality" when the system grows large—when either the number of nodes or the queue lengths become large. In this section, we explore the trade-off between computational complexity and estimation accuracy, comparing our detailed FJ model (§2) with other relevant "black-box" models.

We explore three alternative models, all providing an estimated distribution for the delay in the FJ part of the network at the time of arrival to the first station. Below, we elaborate on each of the three models, including their computational complexity and the parameters to be considered to allow fair comparison between them. As in the previous section, we denote the number of customers seen upon arrival to the first station, the UFJ station, and the LFJ station by k, h_1, and h_2, respectively. We denote the equivalent service rates by μ_0, μ_1, and μ_2. For convenience, we shall also denote $\gamma_0 = \frac{k+1}{\mu_0}$, $\gamma_1 = \frac{h_1+1}{\mu_1}$, and $\gamma_2 = \frac{h_2+1}{\mu_2}$.

The first model will be called the *two-queue-lengths* (2QL) model. This model includes two queues in tandem; that is, we reduce the FJ part to a single station, the one with the highest workload. This means that we exclude the LFJ station if $\gamma_1 > \gamma_2$, and we exclude the UFJ station otherwise. Note that if the service rates at the two FJ stations are equal, then, in a specific time epoch, the station with the longer queue will have a higher load. The underlying assumption of this model is that during the tagged customer's stay at the first station, the same FJ station will remain the most loaded and the queue length at this FJ station will not change considerably (the snapshot principle). In this model, the sojourn time at the FJ part will be determined by the critical FJ station. Thus, if $\gamma_1 > \gamma_2$ $(\gamma_2 > \gamma_1)$ the customer's sojourn time in the FJ part will have an $Erlang\,(h_1 + 1, \mu_1)$ $(Erlang\,(h_2 + 1, \mu_2))$ distribution, which is computed in $O(1)$ time.

The second model is an extension of 2QL, which we will refer to as the *tandem-critical* model, as we again reduce the FJ part to the station with the highest workload. However, this time, we account for the queue evolution at this station during a customer's stay at the first station as a special case of the model presented and analyzed in Boxma and Daduna (2014). To calculate the computational complexity of this model, we can assume, for simplicity, that $h_1 = h_2 = h$. The computational complexity of calculating $a(\cdot)$, $b(\cdot)$, $c(\cdot)$, and $d(\cdot)$ in (4) can be regarded as $O(1)$. The number of such computations required in this model is $O\left(\sum_{i=1}^{k}(h+i)\right)$.

The third model we consider is a *delay-node* model. Here we propose to use a delay node within the FJ part. A delay node is an infinite-server station with a generally distributed service time. This approach was inspired by Yom-Tov and Mandelbaum (2014) and van Leeuwaarden et al. (2017), who used it for making staffing decisions in the complex ED environment. Those papers neglected the FJ structure inherent in ED services. Nevertheless, the delay-node approach can capture the dynamic structure so common in ED services. The resulting model will then have one single-server queue, followed by an FJ part consisting of one single- or multiple-server queue with exponentially distributed service durations and one infinite-server queue with general service durations. We consider two variations of the delay-node model, one in which the more loaded station in the FJ part becomes the delay node (denoted *delay-high*), and the other in which the less loaded station is the delay node (denoted *delay-low*).

This model can be reduced to one very similar to that presented in Boxma and Daduna (2014). With the delay node, in each of the recursion iterations, the joint LST is conditioned only on the number of potential service completions at one station with exponential service durations (and not on two such stations) in accordance with the recursion in Boxma and Daduna (2014). It thus has the same computational complexity as the tandem-critical model. However, the time each customer spends at the last station (an FJ with one delay node and one Markovian node) does not follow an Erlang distribution. Instead, it is the maximum of two independent random variables, one with a general distribution (corresponding to the sojourn time in the delay node) and one with an Erlang distribution. Formulas for this model are presented in EC.4.

Here, we assume that the UFJ station is the delay node, having (merely as an example) an exponentially distributed service time with rate $\frac{\mu_1}{h_1+1}$. This means that the time spent in the delay node is proportional to the workload in this node at the time of arrival, similar to the processor-sharing service policy. As we shall see, this simplification of the delay node allows for fair comparison between the different models and, in addition, accounts for the fact that a higher load at the UFJ station results in larger sojourn times.

In this section, we qualitatively examine the differences between the three models and the model presented in §2, which will be denoted below as the *Markovian fork-join (MFJ)* model. In §5.3, we evaluate the models' performance using a case study of an ED. Assuming that the true underlying system is, in fact, a Markovian single-server queue followed by a two-station Markovian FJ, the model that perfectly captures this system is our MFJ model. However, the MFJ model has a computational complexity of $O\left(\sum_{i=1}^{k}(h+i)^2\right)$ (assuming that $h_1 = h_2 = h$), which has approximately $(h+k)$ times more computations than the tandem-critical model and the delay-node model, and $(h+k)^2 k$ times more computations than the 2QL model. This raises a number of questions: Under which circumstances is it enough to use the simple 2QL model, which does not account for the system dynamics? Under which circumstances can we neglect the FJ structure and use the tandem-critical model? Finally, under which circumstances can we use the delay-node model?

We thus examine the difference in the theoretical density function of the sojourn time in the FJ part under different scenarios, where a *scenario* is a combination of the three queue lengths $(k,\, h_1,\, h_2)$, as seen by an arriving customer. For convenience, we shall assume from now on that all service rates are equal to one and that $\gamma_1 \geq \gamma_2$ in all scenarios. Figure 3 presents the probability density functions of our MFJ model (blue, solid), the delay-high model (purple, dotted), the delay-low model (yellow, dashed), the tandem-critical model (red, dashed-dotted), and the 2QL model (green, solid), for various scenarios.

We seek to establish guidelines, or conditions, under which one can use simpler models to estimate the sojourn-time distribution in the FJ part with fairly good accuracy. We start by exploring the effect of modeling one FJ station as a delay node. We note that when one FJ station has a much higher load than the other, the sojourn time in the FJ part will be mainly determined by the more loaded station. Hence, modeling the other station as a delay node, that is, using the delay-low model, will not have a substantial effect on the sojourn time in the FJ part. This means that using the delay-low model will result in almost the same density function as the one derived from the MFJ model. By contrast, using the delay-high model will result in a very different density function.

As expected, when the loads in both FJ stations are equal, there is no difference between the delay-high and delay-low models. We do observe two interesting phenomena. Consider the case where $h_1 = h_2 = 1$. Considering one station as a delay node, with an exponential service duration, should not affect the time that a single customer currently in service at the FJ stations spends there. However, it does affect the time that a tagged customer, entering the first single-server station, spends in the FJ part. To be concrete, the delay-node models will underestimate this tagged customer's time in the FJ part, and as the number of customers in front of the tagged customer at the first station increases, so does the underestimation. This is because each customer leaving the first station may find a queue at both FJ stations. According to the delay-node models, there is no queue at one station. Hence, each of these customers has a positive probability of being delayed in the queue that is not accounted for by the delay-node models, and this delay will accumulate until the tagged customer reaches the FJ station.

A second interesting phenomenon that we observe is the following: as the number of customers at the FJ stations increases, the density function resulting from the delay-node models has increasingly greater variability than the density function derived from the MFJ model. To explain this phenomenon, assume that a tagged customer arriving to the FJ part finds $h_1 = h_2 = h > 1$ customers at the FJ stations. According to the delay-node model, the time this tagged customer will spend in the FJ part is $max\left\{Erlang\left(h+1,1\right), Exp\left(\frac{1}{h+1}\right)\right\}$, while according to the MFJ model it is $max\left\{Erlang\left(h+1,1\right), Erlang\left(h+1,1\right)\right\}$. The delay-node models result in greater variability since $\mathbb{V}ar\left[Erlang\left(h+1,1\right)\right] = (h+1) < \mathbb{V}ar\left[Exp\left(\frac{1}{h+1}\right)\right] = (h+1)^2$, even though both have the same expected value, $h+1$.

We now seek to explore when one should use a model that accounts for the FJ structure, or equivalently, when one should ignore it and use the tandem-critical model. We observe that when both FJ stations are equally loaded, the tandem-critical model will underestimate sojourn times at those stations. This is due to the fact that $Erlang\left(n_1,\alpha\right) \preceq \max\{Erlang\left(n_1,\alpha\right), Erlang\left(n_2,\alpha\right)\}$,[1] and as n_2 increases, getting closer to n_1, the difference between the two distributions increases. However, if $n_2 \ll n_1$, as in the case where one FJ station is much more loaded than the other one, then $Erlang\left(n_1,\alpha\right) \approx \max\{Erlang\left(n_1,\alpha\right), Erlang\left(n_2,\alpha\right)\}$; hence, in such cases it is reasonable to use the simpler tandem-critical model instead of a model that accounts for the FJ structure. In short, when one FJ station is much more loaded than the other, it is very unlikely that the less loaded station will cause synchronization delays; hence, accounting for the FJ structure is less important.

Lastly, to explore the effect of the FJ queue evolution throughout the tagged customer's stay in the system, we examine the density function resulting from the 2QL model, which does not account for this evolution. From Figure 3, we conclude that it is reasonable to use the queue-length model when one of the FJ stations, or both of them, is much more loaded than the first single-server station, that is, when $\gamma_1 \gg \gamma_0$. In such cases, the time the tagged customer will spend in the first station is much shorter than in the FJ station; hence, by the time they leave the first station, the change in the loaded FJ station queue will be negligible, compared to the initial queue there.

To conclude, when one or both of the FJ stations is much more loaded than the first single-server station, the simple 2QL model is interchangeable with the tandem-critical model. When one FJ

[1] Here "$\preceq$" is in the sense of stochastic ordering, that is, $A \preceq B$ means that $\mathbb{P}\left(A > x\right) \leq \mathbb{P}\left(B > x\right), \quad \forall x \in (-\infty, \infty)$.

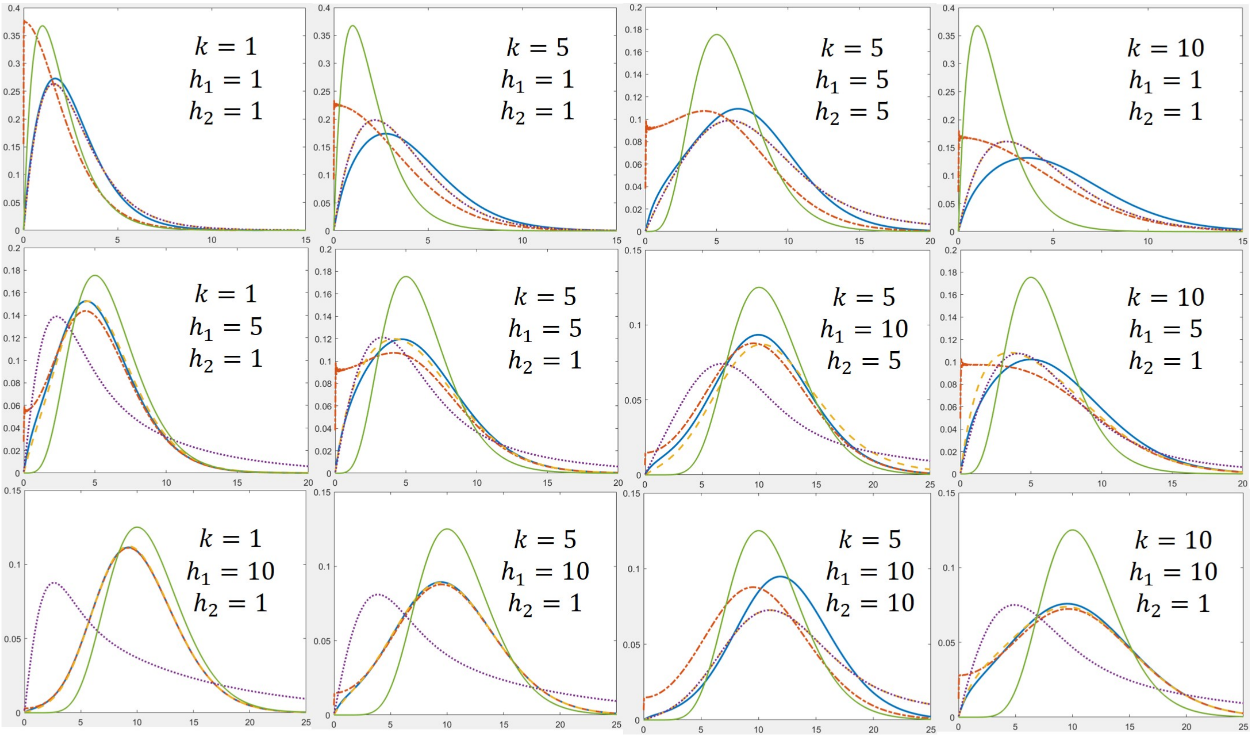

Figure 3 **Sojourn-time distribution in the FJ station for various scenarios according to the following models: MFJ (blue), delay-high (purple, dotted), delay-low (yellow, dashed), tandem-critical (red, dashed-dotted), and queue-length (green, solid).**

station is much more loaded than the other FJ station, the tandem-critical model and the delay-low model are interchangeable with the MFJ model.

5. "All models are wrong, but some are useful" (George Box)

As this well-known quote implies, any model, ours included, is constrained by its assumptions—assumptions that are rarely fully met. Using a case study (§5.1) and a simulation study, we shall demonstrate that, despite this fact, the proposed model is robust enough to be practical in real, complex, challenging service networks, such as those in healthcare systems. To this end, we first compare estimations based on our MFJ model to two commonly used queueing theory-based estimators and to a state-of-the-art machine learning method (§5.2). Our next step (§5.3) is to numerically evaluate the value of delay estimators that account for synchronization queues, which naturally arise in an FJ structure. To do so, we compare the estimation accuracy of our MFJ model with the accuracy of the simplified models presented in §4. Finally, in §5.4, we check for robustness by relaxing the FCFS model assumption.

5.1. Case study

We use two years of data (from April 2014 to August 2016) taken from an Israeli ED. The structure of the ED process was first established by in-depth interviews with various ED personnel—the ED manager, physicians, nurses, and administrators—and then verified by obtaining transaction-level data on all treatments provided throughout a patient's stay. On average, 138 patients arrive to the ED per weekday. The ED classifies them into two groups: non-severe independent *walking* patients (47%), and *acute* patients, which are assigned to a bed (53%). The average LOS in the ED for walking patients is approximately 5 hours; for acute patients, the average LOS is approximately 9.5 hours. As previously mentioned, Figure 1 presents a snapshot of real-time patient flow through the entire ED process. The following case study will focus on part of that network, including the following three stations: nurse attendance, doctor attendance, and lab testing.

Focusing on patients that went through these three stations (82.5% of the ED patients) and based on interviews with nurses and physicians, we assumed that the process in these stations can be modeled by a single- or few-server station which is followed a two-station FJ. For most patients at the nurse attendance station, the nurse will also take a blood sample to be tested. In some cases, the test results will arrive before the patient's arrival to the doctor attendance station, and in other cases, afterwards. Therefore, the laboratory testing and the doctor attendance are done in parallel and can be modeled as an FJ network (see Figure 4 for illustration). This description was also verified via the data we collected. However, we also note that unlike the FJ network assumptions, even if the doctor is available, they will sometimes wait for the lab tests to arrive before examining a patient.

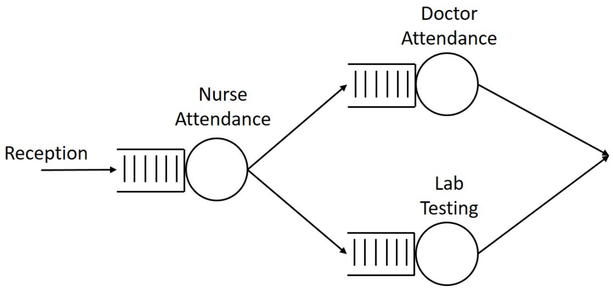

Figure 4 Modeling the beginning of an ED process as a single-server queue followed by a fork-join queue.

Our model requires six parameters: the queue length and the service rate at each of the three stations. We estimated the service rates using the data of the first two years (2014–2015) as the training set data (72% of the data observations). We then used the 2016 data as a test set, upon which we compared the different models. Separation of the data into two different time periods (as opposed to random separation) ensures that the observations in the two data sets are independent. More details on parameter estimations are provided in §EC.5.

Figure 5 presents the service rate at the three stations as a function of the queue length. It seems that, in general, there is a speed-up phenomenon in all three stations: as the queue length increases, the service rate increases. This may be due to service providers accelerating their work in order to mitigate the load in the system or simply due to extra staffing in loaded times. Incorporating such phenomena into our model is straightforward: we simply change service rates to be state-dependent.

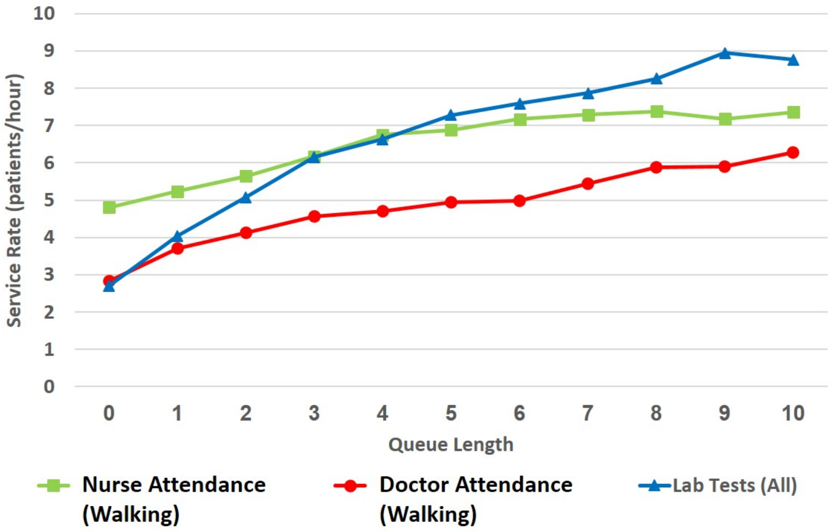

Figure 5 **Service rates as a function of queue lengths.**

5.2. Point estimations

In this section, we compare our primary model (see §2) results with actual patient sojourn times according to our ED data. A patient's sojourn time at the first station (S_0) is considered to be the time between administrative reception and nurse attendance. A patient's sojourn time at the second station is considered to be the time between nurse attendance and doctor attendance. For most of the blood tests (70% of the cases), the hospital records the time of a blood test order. This time is usually only a few minutes after the nurse attendance time, which suggests that blood tests are taken during the nurse attendance. Hence, for all cases, even when there was no record of a blood test order (about 30% of the cases), we assumed that the laboratory testing phase starts at nurse attendance and ends when the final blood test results arrive. Consequently, the sojourn time in the FJ part (S_{FJ}) is defined as the maximum of the doctor attendance sojourn time and the laboratory testing sojourn time. The total sojourn time was then calculated as the summation of the sojourn time at the first station and the sojourn time in the FJ part.

Our data included $112,727$ observations. We excluded all observations with zero sojourn time in at least one of the stations, (approximately 4.8% of all observations) or with a sojourn time exceeding 5 hours in at least one of the stations (0.5% of the observations). This resulted in $106,662$ observations. As stated above, we divided the data into two time periods (after verifying that there were no trends): all observations from 2014–2015 were used as a training set, and all observations from 2016 were used as a test set.

As in §4, we shall define a *scenario* as a combination of the three queue lengths $(k,\ h_1,\ h_2)$, as seen by an arriving patient. We estimated the three service rates for each scenario as specified in §EC.5. Note that these service rates are dependent not on the queue length in any specific station, but rather on the queue lengths in all three stations. We then tested different models on all observations in the test set for which the three queue lengths, as seen at the time of arrival, fit a scenario that was observed at least 30 times in the training set. This resulted in a total of 191 scenarios for walking patients, 277 scenarios for acute patients, and 5,843 (12,365) observations of walking (acute) patients in the test set. These observations included all patients who went through all three stations (that is, it does not include patients without lab tests).

We compare our method with three other estimation methods: a variation of the Last-to-Enter-Service (LES) estimator, the 2QL estimator, and a machine learning, Gradient Boosting Regressor (GBR) (Friedman 2001) with different sets of features.

The LES estimator is easy to implement, fairly robust, and hence widely used (Gal et al. 2017, Huang et al. 2015); it is based on the snapshot principle, which has been proven to also hold for FJ networks (Nguyen 1993). Consider a tagged patient arriving at the administrative reception. The LES estimator is based on the principle that this tagged patient's waiting time at any of the following three stations (nurse attendance, doctor attendance, and lab testing) can be estimated by the waiting time of the last patient to enter service at that station, prior to the tagged patient's reception time. Our data includes a single time stamp for every activity done in the ED. For example, we do not know exactly when a nurse attendance service started and ended, but rather we have a single time stamp indicating that this service occurred. This time stamp is associated with the fulfilment of the patient electronic medical file by the nurse; therefore, it is reasonable to assume that it occurred at (or close to) the end of the service. Thus, to estimate sojourn times we take a variation of the LES and use the sojourn time of the last patient to *complete* service at each of the three stations. This estimator will be denoted Last to Complete Service (LCS). In order to estimate the sojourn time of a tagged patient in the FJ part, we first compute the patient's LCS estimators for the doctor attendance and lab testing stations separately, and then take the maximal value of the two. This is based on the snapshot principle for FJ networks proved by Nguyen (1993). The total time of the LCS estimator will be the summation of the LCS estimators for the first station and for the FJ station.

The 2QL estimator is simply an extension of the commonly used QL estimator (Ibrahim and Whitt 2009a) for two queues in tandem. As in our model, its only parameters are the queue length observed at the time of arrival and the total service rate at the station. As described in §4, the 2QL model in fact reduces the FJ part to a single station, represented by the more loaded station at the time of the customer arrival to the first station.

The GBR is a tree ensemble model. It starts with a regression tree of a fixed size as the base learner. At each iteration, a new estimator is added to the model, obtaining a new model (tree), that attempts to improve the former one. The added estimator is trained to fit the residual of the last model (i.e., the difference between the model's predicted values and the actual values). When using the mean squared error (MSE) loss function, these residuals correspond to the negative gradients of the loss function, and therefore they define the "steepest descent" direction.

A summary of the results is provided in Table 1. The columns correspond to the FJ part (doctor attendance and lab testing) and the total sojourn time, for both walking and acute patients. The top two rows present statistics of the average sojourn time and its standard deviation. The rest of the rows present the MSE of the different models (in minutes). Note that our model (denoted by MFJ) and the 2QL model provide estimations for the entire distribution, while the others only provide a single-point estimation; hence, for "fair" comparison, we present here the MSE with respect to the expected sojourn times. We consider two variations of the GBR model. In GBR (PA), the explanatory variables are individual patient attributes set at the time of the customer arrival: gender, age, means of arrival (independently, by ambulance, etc.), and a set of variables indicating arrival time (hour, day of week, month). In GBR (PA+MFJ), we add to the patient's attributes an additional explanatory variable which is the result of our MFJ model.

It is clear that our model outperforms the LCS estimator and the 2QL estimator, with over 40% improvement in the MSE compared to the LCS estimator, and approximately 5%–7% improvement

	Walking		Acute	
	FJ Station	Total	FJ Station	Total
Avg. sojourn time	59.58	87.27	59.64	87.78
Sojourn time STD	36.7	42.08	39.16	43.79
LCS	2409	2945	2833	3354
2QL	1522^*	1796^*	1803^*	2056^*
MFJ	1419^*	1695^*	1664^*	1944^*
GBR (PA)	1300^*	1676	1516^*	1848^*
GBR (PA+MFJ)	1278^*	1558^*	1507^*	1802^*

Table 1 Mean squared error (in minutes) of various models in estimating sojourn times in the FJ part and the total sojourn time. Values marked with $*$ are those for which the difference from the MSE value above is statistically significant according to the Diebold–Mariano test (Diebold and Mariano 2002).

.

compared to the 2QL estimator. One may argue that the somewhat small improvement in the MSE compared to the 2QL estimator does not justify the additional computational complexity required for the MFJ estimator; however, the two estimators might result in very different distributions. In §4 we discussed the implications of using the 2QL estimator to predict distributions and under what conditions one could use it instead of the MFJ model. In the next section (5.3), we examine the accuracy of the two methods in estimating delay distributions in the FJ part.

We observe that the GBR (PA) model reduces the MSE by approximately 9% in the FJ station and by only 1%–5% in estimating the total sojourn time, compared to our MFJ model. Adding the mean delay time according to our MFJ model as an additional explanatory variable further reduces the MSE; the reduction is statistically significant according to the Diebold–Mariano test for predictive accuracy comparison (Diebold and Mariano 2002).[2] This result replicates previous results (e.g., Ang et al., 2016, Gal et al., 2017), showing that the integration of queueing theory-based estimators as features of machine learning techniques achieves the best estimation accuracy.

It is important to note that the superiority of the ML method is not surprising. In Chocron et al. (2021), it was shown that for a simple single-station queue with FCFS priority policy, QT-based estimators can be compatible with ML methods; however, for more complex systems, ML estimators outperform QT-based estimators. Be that as it may, QT-based estimators, ours included,

[2] Similar results were obtained using linear regression and an XGBoost algorithm (Chen and Guestrin 2016). The results reported here are the best achieved within the set of learning models that we tried.

do not require a learning phase. That is, they can be immediately implemented in any service system. By contrast, ML methods require a learning phase that is unique to every service system. One cannot "copy" an ML model from one system to another without having to train the model again. This requires the preliminary work of collecting enough data, cleaning it, and calculating relevant features, all of which usually demand a lot of time and effort. In our case study, these ML techniques yield only a minor improvement, which may not justify the additional effort required in order to apply these models in other service systems.

5.3. The value of an FJ model

In §4, we explored, qualitatively, the influence of different model structures on the resulting delay distributions. Here, we use our case study to compare the estimation accuracy of the different models discussed in §4. We show that if the underlying system dynamic indeed includes an FJ structure, and therefore synchronization queues, then using simplified models that do not account for that structure can lead to less accurate delay estimations.

We compare the models' accuracy in providing a single-point estimation, namely the distribution mean, for the delay in the FJ part (doctor attendance and laboratory testing). In addition, we put special emphasis on examining the models' accuracy in capturing the tail of the distribution, and in particular, their ability to avoid underestimation. We argue that avoiding underestimation of delays is more important than avoiding overestimation, due to the asymmetry in people's reactions to estimation errors. For example, Efrat-Treister et al. (2020) showed that when actual waiting times exceed the announced waiting times, ED patients react negatively and tend to be more aggressive (a reaction that did not happen with overestimations).

Table 2 presents the MSE of the single-point estimation, as well as the models' accuracy in capturing the 75th and the 90th percentiles. To this end, we calculated the percentage of patients in our test set whose sojourn time in the FJ part was less than or equal to the appropriate values of the 75th and the 90th percentiles, according to each of the models. The closer these percentages of patients are to 75% and 90%, the better the models capture the distribution tail, and percentages higher than the percentiles (overestimations) are better than lower (underestimations).

We observe that the MFJ model provides the best estimation for the mean sojourn time in the FJ part, for both walking and acute patients; the MSE reduction is statistically significant according to the Diebold–Mariano test (Diebold and Mariano 2002), with respect to the model providing the least MSE value among all the other models. For acute patients (68% of our test population), the MFJ model also provides the best performance in capturing the tail of the sojourn-time distribution.

	Walking			Acute		
	Mean (MSE)	75th %	90th %	Mean (MSE)	75th %	90th %
2QL	1522	0.64	0.8	1803	0.72	0.85
tandem-critical	1616	0.60	0.79	1720	0.73	0.87
delay-high	1473	0.76	0.93	1946	0.82	0.93
delay-low	1460	0.73	0.88	1853	0.81	0.91
MFJ	1419	0.69	0.85	1664	0.79	0.89

Table 2 Comparison of different model structures in estimating the mean sojourn times in the FJ part and in estimating the tail of the sojourn-time distribution.

.

Note that the two models which neglect the FJ structure (2QL and tandem-critical), and thus overlook the effect of synchronization queues, substantially underestimate the sojourn time at the FJ part for both walking and acute patients. This is yet another support for the need to account for FJ structures when present.

Another observation is that for walking patients, the delay-node models outperform the MFJ model. This may be due to the fact that the ED patients who wait for the two services (doctor attendance and laboratory testing) only partially represent the actual queue and load in those stations; the lab processes patients' samples from the entire hospital, and the doctor may (and usually does) attend each ED patient more than once. This raises the problem of missing data, and indeed, one way to handle this is to ignore the queue information and treat such stations as delay nodes. This idea was implemented successfully for ED staffing problems by Yom-Tov and Mandelbaum (2014). Our hypothesis is that the missing data problem is less relevant for acute patients due to their clinical condition which sets them in a higher priority. For example, the doctor may prioritize newly arrived acute patients over patients who were already seen. Thus, while the delay-node models capture the tail of the FJ sojourn-time distribution better when it comes to walking patients, when considering acute patients, the missing data effect diminishes and therefore an exact model, such as the MFJ model, outperforms the delay-node models.

5.4. Priority policies

One of the main model assumptions is that each station operates according to an FCFS priority policy. However, in healthcare systems, particularly in EDs, patients are assigned different priorities according to their medical condition. Since our data does not include priority levels, we used simulation to examine the influence of having a priority policy on the accuracy of our estimations.

We assumed that arrivals follow a non-homogeneous Poisson process, where the time-dependent arrival rate was taken as the average hourly arrival rate of walking patients according to our data (April 2014 to August 2016). We ran the simulation for a period of two years (17,520 hours) and then removed the first and last weeks (any patient arriving at $t < 168$ hours or $t > 17,352$ hours), to allow for a warm-up period and to account for all patients who may have passed lower-priority patients. We took all scenarios in which $k, h_1, h_2 \in \{0, 1, ..., 5\}$, as these are the relevant queue lengths according to our case study ED data. We assumed that all priority classes have the same service rates (which are equal to the average service rates of walking patients, according to our data: $\mu_0 = 5.5, \mu_1 = 4.2, \mu_2 = 6$). We also assumed a non-preemptive static priority policy; we believe this is quite reasonable when focusing on walking patients. The percentage of arriving patients from each priority level was set according to the parameters presented in Huang et al. (2015). They focused on triage levels 3 to 5, which include the less severe patients (5 being the least severe) and are compatible with walking patients. Level 3 patients constituted 10% of the arriving population, and level 4 and 5 patients constituted 40% and 50%, respectively.

We compared our MFJ model (which does not account for priorities) with LCS estimators (see §5.2) with and without priorities. The ordinary LCS estimator does not take into account the underlying priority policy that is operating in the simulation. It estimates the sojourn time of an arriving tagged customer as the sojourn time of the last customer who left the service station prior to the tagged customer's arrival. By contrast, the LCS estimator with priorities estimates the tagged customer's sojourn time as the sojourn time of the last customer who left the service station *from the same priority level as the tagged customer* (in the spirit of Thiongane et al. 2016).

Table 3 presents the MSE of the three models (in minutes) compared to the simulation results. This includes the total MSE as well as the MSE for each priority class (low, medium, and high). First, we observe that in all cases, except for high-priority customers in the first station, our MFJ model outperforms the LCS model with priorities, reducing MSE by 30%–38%. We believe this is due to the fact that our model accounts for the system's evolution over time, whereas the LCS estimator, with or without priorities, cannot capture the changes in queues during a customer's sojourn time in the system. The LCS estimator is a snapshot-principle-based estimator. The two main assumptions of the snapshot principle are that changes in queue lengths are negligible during a customer stay in the system, and that the system is operating under an FCFS policy. The LCS estimator with priorities relaxes the assumption of the FCFS policy, while our MFJ estimator relaxes the assumption of a "steady state" during a customer stay in the system. We observe that

	First station				FJ station			
	Low	Medium	High	Total	Low	Medium	High	Total
Avg. sojourn time	32.58	20.13	17.01	26.08	71.48	46.32	38.57	58.20
LCS	1763	699	567	1220	8487	4129	3201	6229
LCS with priorities	1831	516	340	1165	8749	2444	1573	5526
MFJ	1244	320	345	787	6134	1526	1037	3793

Table 3 **Mean squared error (in minutes) of different models for estimating sojourn times in the first station and the FJ station, for low-priority (16,625 observations), medium-priority (13,210 observations), and high-priority customers (3,254 observations).**

in this time-varying system, where we have a combined effect of resource queues and synchronization queues, the latter relaxation allows greater improvement. The same results were obtained when repeating the simulation with high-priority customers, who comprised 50% of arriving customers (low priority: 10%, medium priority: 40%). According to Bassamboo and Ibrahim (2021), the LES estimator performs better in moderately to highly congested systems. The traffic intensity ρ (average daily arrival rate divided by average service rate) in our simulation, which is based on our case study ED data, is fairly low at all three stations (up to 0.7), which could explain the poor results of the LCS estimator compared to our MFJ estimator.

We also observe that considering priorities greatly improves the LCS results for medium- and high-priority classes. However, for low-priority customers, we see no improvement in the results, but rather an increase in the MSE by 3%. We argue that this is due to the fact that substantial changes in the three queue lengths may occur during the sojourn time of low-priority customers. First, the sojourn time of low-priority customers in the system is much higher than that of customers from higher priority classes, as observed in Table 3. Secondly, during the waiting time of low-priority customers, higher-priority customers may enter the system and prolong the queue in front of them. In addition, low-priority customers will only be served, one by one, when there are no higher-priority customers in the system, until the next higher-priority customer arrives. This process creates batches of customers from the same priority class in the departure process of a service station. Thus, there could be a considerable time difference between the arrival epoch of a new low-priority customer and the last service completion of a low-priority customer. Hence, for low-priority customers, the priority LCS estimator may be based on lagged information.

6. Discussion

Our research goal was to develop real-time, state-dependent estimations for delay distributions in service networks that include FJ structures. The motivation came from healthcare systems that are known to be complex queueing networks. We developed a recursive formula to estimate sojourn times in a network with one single-server queue followed by an FJ part with two stations. We derived an estimation for the sojourn time in the first (single-server) part, the sojourn time in the second (FJ) part, and the total sojourn time; all three estimations are calculated at the time of a customer's arrival to the system and are based on the state of the system in that arrival epoch.

We examined our methodology in a case study of an Israeli ED using unique transaction-level data. This case study revealed that, although essentially none of our model assumptions hold precisely, our methodology provides fairly accurate estimations for sojourn-time distributions and outperforms other commonly used QT-based estimators, such as LCS (a small variation of LES) and 2QL (an extension of QL to two queues in tandem). Furthermore, we showed that although machine learning estimation methods may outperform our model estimations, incorporating our model result as an additional feature for these methods provides the most accurate estimations. We also showed via simulation that even if a priority policy is in place, our methodology, which ignores priority, still provides better results than other estimators that *do* account for it.

We start with a Markovian model but then relax some of the distributional assumptions, providing a natural extension of our model to include general service times in the single-server station, and Erlang or hyperexponential service times in the FJ stations. In addition, we explore qualitatively different models that either neglect or simplify the FJ structure. While having lower computational complexity, these models may fail to capture delays that are due to synchronization queues emerging from the FJ structure. We highlight this trade-off and provide initial modeling guidelines with regard to the most appropriate model under different scenarios.

The simplified models are one resort to the "curse of dimensionality," which is common with the exact analysis approach taken here. That is, computations may become cumbersome if there are more stations in the FJ part or if queue sizes grow large. Another solution, which is usually appropriate in large systems that operate in heavy traffic, is to develop delay estimations that are based on fluid or diffusion approximations. We leave this approach for future research.

References

Ang E, Kwasnick S, Bayati M, Plambeck EL, Aratow M (2016) Accurate emergency department wait time prediction. *Manufacturing & Service Operations Management* 18(1):141–156.

Armony M, Shimkin N, Whitt W (2009) The impact of delay announcements in many-server queues with abandonment. *Operations Research* 57(1):66–81.

Baccelli F, Makowski AM, Shwartz A (1989) The fork-join queue and related systems with synchronization constraints: Stochastic ordering and computable bounds. *Advances in Applied Probability* 21(3):629–660.

Bassamboo A, Ibrahim R (2021) A general framework to compare announcement accuracy: Static vs. LES-based announcement. *Management Science* 67(7):4191–4208.

Benevento E, Aloini D, Squicciarini N, Dulmin R, Mininno V (2019) Queue-based features for dynamic waiting time prediction in emergency department. *Measuring Business Excellence* 23(4):458–471.

Boxma O (1983) The cyclic queue with one general and one exponential server. *Advances in Applied Probability* 15(4):857–873.

Boxma O, Daduna H (2014) The cyclic queue and the tandem queue. *Queueing Systems* 77(3):275–295.

Boxma OJ, Daduna H (1990) Sojourn times in queueing networks. Takagi H, ed., *Stochastic Analysis of Computer and Communication Systems*, 401–450 (North-Holland Publishing Company, Amsterdam).

Boxma OJ, Koole G, Liu Z (1994) Queueing-theoretic solution methods for models of parallel and distributed systems. *In: Performance Evaluation of Parallel and Distributed Systems—Solution Methods*, Proceedings of the Third QMIPS Workshop, Part 1 (CWI Tract 105, Amsterdam).

Chen T, Guestrin C (2016) XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794 (ACM).

Chocron E, Feigin PD, Cohen I (2021) Queueing theory or queue mining for delay prediction in multi-class service systems. Working paper.

Daduna H (1986) Two-stage cyclic queues with nonexponential servers: Steady-state and cyclic time. *Operations Research* 34(3):455–459.

Diebold FX, Mariano RS (2002) Comparing predictive accuracy. *Journal of Business & Economic Statistics* 20(1):134–144.

Dong J, Yom-Tov E, Yom-Tov GB (2019) The impact of delay announcements on hospital network coordination and waiting times. *Management Science* 65(5):1969–1994.

Efrat M, Parush A (2017) Personalized information to patients in the emergency department: User centered design and testing, *Production and Operations Management Society (POMS) Conference.*

Efrat-Treister D, Moriah H, Rafaeli A (2020) The effect of waiting on aggressive tendencies toward emergency department staff: Providing information can help but may also backfire. *PloS one* 15(1):e0227729.

Friedman JH (2001) Greedy function approximation: A gradient boosting machine. *Annals of Statistics* 29(5):1189–1232.

Gal A, Mandelbaum A, Schnitzler F, Senderovich A, Weidlich M (2017) Traveling time prediction in scheduled transportation with journey segments. *Information Systems* 64:266–280.

Huang J, Carmeli B, Mandelbaum A (2015) Control of patient flow in emergency departments, or multiclass queues with deadlines and feedback. *Operations Research* 63(4):892–908.

Ibrahim R (2018) Sharing delay information in service systems: A literature survey. *Queueing Systems* 89(1–2):49–79.

Ibrahim R, Whitt W (2009a) Real-time delay estimation based on delay history. *Manufacturing & Service Operations Management* 11(3):397–415.

Ibrahim R, Whitt W (2009b) Real-time delay estimation in overloaded multiserver queues with abandonments. *Management Science* 55(10):1729–1742.

Ibrahim R, Whitt W (2011a) Real-time delay estimation based on delay history in many-server service systems with time-varying arrivals. *Production and Operations Management* 20(5):654–667.

Ibrahim R, Whitt W (2011b) Wait-time predictors for customer service systems with time-varying demand and capacity. *Operations Research* 59(5):1106–1118.

Jouini O, Aksin Z, Dallery Y (2011) Call centers with delay information: Models and insights. *Manufacturing & Service Operations Management* 13(4):534–548.

Kerner Y (2008) The conditional distribution of the residual service time in the Mn/G/1 queue. *Stochastic Models* 24(3):364–375.

Kim C, Agrawala AK (1989) Analysis of the fork-join queue. *IEEE Transactions on Computers* 38(2):250–255.

Ko SS, Serfozo RF (2004) Response times in M/M/s fork-join networks. *Advances in Applied Probability* 36(3):854–871.

Ko SS, Serfozo RF (2008) Sojourn times in G/M/1 fork-join networks. *Naval Research Logistics (NRL)* 55(5):432–443.

Maister DH (1984) The psychology of waiting lines. Czepiel JA, Solomon MR, Surprenant CF, eds., *The Service Encounter*, 113–123 (Lexington Books, Lexington, MA).

Mandelbaum A, Yechiali U (1983) Optimal entering rules for a customer with wait option at an M/G/1 queue. *Management Science* 29(2):174–187.

Mourõo R, Carvalho R, Carvalho R, Ramos GN (2017) Predicting waiting time overflow on bank teller queues. *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 842–847 (IEEE).

Munichor N, Rafaeli A (2007) Numbers or apologies? Customer reactions to telephone waiting time fillers. *Journal of Applied Psychology* 92(2):511–518.

Nakibly E (2002) *Predicting waiting times in telephone service systems*. Master's thesis, Technion—Israel Institute of Technology.

Nelson R, Tantawi AN (1988) Approximate analysis of fork/join synchronization in parallel queues. *IEEE Transactions on Computers* 37(6):739–743.

Nguyen V (1993) Processing networks with parallel and sequential tasks: Heavy traffic analysis and Brownian limits. *The Annals of Applied Probability* 3(1):28–55.

Pender J, Rand RH, Wesson E (2016) Managing information in queues: the impact of giving delayed information to customers. Working paper. arXiv preprint arXiv:1610.01972.

Qiu Z, Pérez JF, Harrison PG (2015) Beyond the mean in fork-join queues: Efficient approximation for response-time tails. *Performance Evaluation* 91:99–116.

Reiman MI (1982) The heavy traffic diffusion approximation for sojourn times in Jackson networks. *Applied Probability–Computer Science: The Interface*, 409–421 (Springer).

Rizk A, Poloczek F, Ciucu F (2015) Computable bounds in fork-join queueing systems. *ACM SIGMETRICS Performance Evaluation Review*, 43(1):335–346.

Sanit-in Y, Saikaew KR (2019) Prediction of waiting time in one stop service. *International Journal of Machine Learning and Computing* 9(3):322–327.

Senderovich A, Rogge-Solti A, Gal A, Mendling J, Mandelbaum A, Kadish S, Bunnell CA (2016) Data-driven performance analysis of scheduled processes. *International Conference on Business Process Management*, 35–52 (Springer).

Senderovich A, Weidlich M, Gal A, Mandelbaum A (2014) Queue mining–Predicting delays in service processes. *International Conference on Advanced Information Systems Engineering*, 42–57 (Springer).

Senderovich A, Weidlich M, Gal A, Mandelbaum A (2015) Queue mining for delay prediction in multi-class service processes. *Information Systems* 53:278–295.

Stadje W (1996) Non-stationary waiting times in a closed exponential tandem queue. *Queueing Systems* 22(1–2):65–77.

Sun Y, Teow KL, Heng BH, Ooi CK, Tay SY (2012) Real-time prediction of waiting time in the emergency department, using quantile regression. *Annals of Emergency Medicine* 60(3):299–308.

Thiongane M, Chan W, L'Ecuyer P (2016) New history-based delay predictors for service systems. *Winter Simulation Conference*.

Thiongane M, Chan W, L'Ecuyer P (2020) Delay predictors in multi-skill call centers: An empirical comparison with real data. *ICORES*, 100–108.

Thomasian A (2014) Analysis of fork/join and related queueing systems. *ACM Computing Surveys (CSUR)* 47(2):1–71.

van Leeuwaarden JSH, Mathijsen BWJ, Sloothaak F, Yom-Tov GB (2017) The restricted Erlang-R queue: Finite-size effects in service systems with returning customers, working paper.

Whitt W (1999) Predicting queueing delays. *Management Science* 45(6):870–888.

Witkovský V (2016) Numerical inversion of a characteristic function: An alternative tool to form the probability distribution of output quantity in linear measurement models. *Acta IMEKO* 5(3):32–44.

Yom-Tov GB, Mandelbaum A (2014) Erlang-R: A time-varying queue with reentrant customers, in support of health-care staffing. *Manufacturing & Service Operations Management* 16(2):283–299.

Yom-Tov GB, Rafaeli A, Westphal M (2017) An empirical study of customer patience and abandonment in online customer service, working paper.

Yu Q, Allon G, Bassamboo A (2016) How do delay announcements shape customer behavior? An empirical study. *Management Science* 63(1):1–20.

Zhang Z (1990) Analytical results for waiting time and system size distributions in two parallel queueing systems. *SIAM Journal on Applied Mathematics* 50(4):1176–1193.

Technical notes and model extensions

EC.1. Recursive formula calculation

The recursive formula presented in §2, Equation 3, can be easily and efficiently calculated if written in compact matrix form. We define the following two matrices. The first is $\bar{\Psi}_{k-1,h_1+1,h_2+1}$, given by

$$\bar{\Psi}_{k-1,h_1+1,h_2+1} = \begin{pmatrix} \Psi_{k-1,h_1+1,h_2+1} & \Psi_{k-1,h_1+1,h_2} & \dots & \Psi_{k-1,h_1+1,1} \\ \Psi_{k-1,h_1,h_2+1} & \Psi_{k-1,h_1,h_2} & \dots & \Psi_{k-1,h_1,1} \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_{k-1,1,h_2+1} & \Psi_{k-1,1,h_2} & \dots & \Psi_{k-1,1,1} \end{pmatrix},$$

and its dimensions are $(h_1+1)\times(h_2+1)$. The second matrix, Φ_{h_1,h_2}, is defined as

$$\Phi_{h_1,h_2} = \begin{pmatrix} a(0,0) & a(1,0) & \dots & a(h_1-1,0) & c(h_1,0) \\ a(0,1) & a(1,1) & \dots & a(h_1-1,1) & c(h_1,1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a(0,h_2-1) & a(1,h_2-1) & \dots & a(h_1-1,h_2-1) & c(h_1,h_2-1) \\ b(0,h_2) & b(1,h_2) & \dots & b(h_1-1,h_2) & d(h_1,h_2) \end{pmatrix}$$

using the notation in §3, and its dimensions are $(h_2+1)\times(h_1+1)$. We now observe that

$$\Psi_{k,h_1,h_2} = trace\left(\bar{\Psi}_{k-1,h_1+1,h_2+1}\cdot\Phi_{h_1,h_2}\right).$$

If there are k customers at the first station, h_1 customers at the UFJ station, and h_2 at the LFJ station, by the time the tagged customer completes service at the first station there could be up to h_1+k customers at the UFJ station and up to h_2+k customers at the LFJ station. Thus, to recursively compute Ψ_{k,h_1,h_2}, we start with the $(h_1+k)\times(h_2+k)$ matrix:

$$\bar{\Psi}_{0,h_1+k,h_2+k} = \begin{pmatrix} \Psi_{0,h_1+k,h_2+k} & \Psi_{0,h_1+k,h_2+k-1} & \dots & \Psi_{0,h_1+k,1} \\ \Psi_{0,h_1+k-1,h_2+k} & \Psi_{0,h_1+k-1,h_2+k-1} & \dots & \Psi_{0,h_1+k-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_{0,1,h_2+k} & \Psi_{0,1,h_2+k-1} & \dots & \Psi_{0,1,1} \end{pmatrix};$$

each of its elements can be easily computed using Equation (6). As a basis for all iterations, it will be useful to define the following four $(h_1+k-1)\times(h_2+k-1)$ matrices:

$$A_{h_1+k,h_2+k} = \begin{pmatrix} a(0,0) & a(0,1) & \dots & a(0,h_2+k-2) \\ a(1,0) & a(1,1) & \dots & a(1,h_2+k-2) \\ \vdots & \vdots & \ddots & \vdots \\ a(h_1+k-2,0) & a(h_1+k-2,1) & \dots & a(h_1+k-2,h_2+k-2) \end{pmatrix},$$

$$B_{h_1+k,h_2+k} = \begin{pmatrix} b(0,1) & b(0,2) & \dots & b(0,h_2+k-1) \\ b(1,1) & b(1,2) & \dots & b(1,h_2+k-1) \\ \vdots & \vdots & \ddots & \vdots \\ b(h_1+k-2,1) & b(h_1+k-2,2) & \dots & b(h_1+k-2,h_2+k-1) \end{pmatrix},$$

$$C_{h_1+k,h_2+k} = \begin{pmatrix} c(1,0) & c(1,1) & \dots & c(1,h_2+k-2) \\ c(2,0) & c(2,1) & \dots & c(2,h_2+k-2) \\ \vdots & \vdots & \ddots & \vdots \\ c(h_1+k-1,0) & c(h_1+k-1,1) & \dots & c(h_1+k-1,h_2+k-2) \end{pmatrix},$$

$$D_{h_1+k,h_2+k} = \begin{pmatrix} d(1,1) & d(1,2) & \dots & d(1,h_2+k-1) \\ d(2,1) & d(2,2) & \dots & d(2,h_2+k-1) \\ \vdots & \vdots & \ddots & \vdots \\ d(h_1+k-1,1) & d(h_1+k-1,2) & \dots & d(h_1+k-1,h_2+k-1) \end{pmatrix}.$$

The elements of these matrices are computed by Equation (4).

We shall also use the following notation: let X be a matrix; then $X[l:m,n:p]$ is the resulting matrix by taking all the elements of X in rows l through m, and columns n through p.

In the first iteration, we compute the matrix $\bar{\Psi}_{1,h_1+k-1,h_2+k-1}$. Each of its elements,

$$\Psi_{1,m,n},\ \forall 1\le m\le h_1+k-1,\ 1\le n\le h_2+k-1,$$

will be calculated as follows:

$$\Psi_{1,m,n} = trace\Big(\bar{\Psi}_{0,m+1,n+1}\cdot\Phi_{m,n}\Big),$$

where

$$\bar{\Psi}_{0,m+1,n+1} = \bar{\Psi}_{0,h_1+k,h_2+k}\Big[h_1+k-m:h_1+k\ ,\ h_2+k-n:h_2+k\Big],$$

and $\Phi_{m,n}$ is composed of the matrices A, B, C, and D, in the following manner:

$$\Phi_{m,n} = \begin{bmatrix} A[1:m,1:n]^T & C[m,1:n]^T \\ B[1:m,n]^T & D[m,n] \end{bmatrix}.$$

In the second iteration, we calculate $\bar{\Psi}_{2,h_1+k-2,h_2+k-2}$. Each of its elements,

$$\Psi_{2,m,n},\ \forall 1 \le m \le h_1+k-2,\ 1 \le n \le h_2+k-2,$$

can be calculated as follows:

$$\Psi_{2,m,n} = trace\Big(\bar{\Psi}_{1,m+1,n+1} \cdot \Phi_{m,n}\Big).$$

We continue this process, until we reach the kth iteration, in which the desired LST will be

$$\Psi_{k,h_1,h_2}(w_0,w_{FJ}) \equiv \Psi_{k,h_1,h_2} = trace\Big(\bar{\Psi}_{k-1,h_1+1,h_2+1} \cdot \Phi_{h_1,h_2}\Big).$$

EC.2. General service times at the first station

In this appendix, we provide a formulation for estimating sojourn times in the FJ network when service time at the first station has a general distribution.

Assume that a tagged customer arrives to the first station and finds k customers there. Let R_k be the remaining service duration of the customer being served at the first station at the time of that tagged customer's arrival (i.e., given that there were k customers in the first queue). Expressions for the expected value of R_k and for its LST are given in Mandelbaum and Yechiali (1983) and Kerner (2008).

Let $B(\cdot)$ be the cumulative distribution function of service duration at the first station (i.e., $S_0|k=0 \sim B$). As before, we define $\Psi_{k,h_1,h_2}(w_0,w_{FJ})$ as the joint Laplace–Stieltjes transform of S_0 and S_{FJ} given k customers at the first station, h_1 customers at the UFJ station, and h_2 customers at the LFJ station.

$$\Psi_{k,h_1,h_2}(w_0,w_{FJ}) \equiv \mathbb{E}_{k,h_1,h_2}\left(e^{-w_0S_0-w_{FJ}S_{FJ}}\right) = \int_0^\infty e^{-w_0t} X_{k,h_1,h_2}\, \mathrm{d}\mathbb{P}\left(R_k < t\right)$$

$$= \int_0^\infty e^{-w_0t} \Bigg[\sum_{r_1=0}^{h_1-1}\sum_{r_2=0}^{h_2-1} e^{-\mu_1 t}\frac{(\mu_1 t)^{r_1}}{r_1!} e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!} \Psi_{k-1,h_1-r_1+1,h_2-r_2+1}(w_0,w_{FJ})$$

$$+ \sum_{r_1=0}^{h_1-1}\sum_{r_2=h_2}^{\infty} e^{-\mu_1 t}\frac{(\mu_1 t)^{r_1}}{r_1!} e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!} \Psi_{k-1,h_1-r_1+1,1}(w_0,w_{FJ})$$

$$+\sum_{r_1=h_1}^{\infty}\sum_{r_2=0}^{h_2-1}e^{-\mu_1 t}\frac{(\mu_1 t)^{r_1}}{r_1!}e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!}\Psi_{k-1,1,h_2-r_2+1}(w_0,w_{FJ})$$
$$\left.+\sum_{r_1=h_1}^{\infty}\sum_{r_2=h_2}^{\infty}e^{-\mu_1 t}\frac{(\mu_1 t)^{r_1}}{r_1!}e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!}\Psi_{k-1,1,1}(w_0,w_{FJ})\right]\mathrm{d}\mathbb{P}\left(R_k<t\right).$$

Here, $\Psi_{l,m,n}\left(w_0,w_{FJ}\right)$ is the LST of the joint distribution of the tagged customer's sojourn time at the first station and at the FJ part, given that *a new service is starting in the first station* and that in this epoch the tagged customer is behind l customers at the first station, and sees m customers at the UFJ station, and n customers at the LFJ station. We now denote

$$a(r_1,r_2,w_0)=\int_0^{\infty}e^{-(w_0+\mu_1+\mu_2)t}\frac{(\mu_1 t)^{r_1}}{r_1!}\frac{(\mu_2 t)^{r_2}}{r_2!}\,\mathrm{d}B(t),$$
$$b(r_1,h_2,w_0)=\int_0^{\infty}e^{-(w_0+\mu_1+\mu_2)t}\sum_{r_2=h_2}^{\infty}\frac{(\mu_1 t)^{r_1}}{r_1!}\frac{(\mu_2 t)^{r_2}}{r_2!}\,\mathrm{d}B(t),$$
$$c(h_1,r_2,w_0)=\sum_{r_1=h_1}^{\infty}a(r_1,r_2,w_0),$$
$$d(h_1,h_2,w_0)=\sum_{r_1=h_1}^{\infty}b(r_1,h_2,w_0).$$

Let X be a Poisson process with rate μ_1, and let Y be a Poisson process with rate μ_2. The above expressions are reduced to

$$a(r_1,r_2,w_0)=\int_0^{\infty}e^{-w_0 t}\mathbb{P}\left(X(t)=r_1,\,Y(t)=r_2\right)\mathrm{d}B(t),$$
$$b(r_1,h_2,w_0)=\int_0^{\infty}e^{-w_0 t}\mathbb{P}\left(X(t)=r_1,\,Y(t)\geq h_2\right)\mathrm{d}B(t),$$
$$c(h_1,r_2,w_0)=\int_0^{\infty}e^{-w_0 t}\mathbb{P}\left(X(t)\geq h_1,\,Y(t)=r_2\right)\mathrm{d}B(t),$$
$$d(h_1,h_2,w_0)=\int_0^{\infty}e^{-w_0 t}\mathbb{P}\left(X(t)\geq h_1,\,Y(t)\geq h_2\right)\mathrm{d}B(t).$$

With these new notations, the recursive formula for calculating $\Psi_{k,h_1,h_2}(w_0,w_{FJ})$ presented in Equation (3) will not change, nor does the rest of the analysis. There will be only minor differences in the formulas of the boundary conditions. Let $\beta(\cdot)$ be the LST of the specified service duration (which can now be general). Thus, the boundary conditions will now be: $\Psi_{0,0,0}(w_0,w_{FJ}) = \beta(w_0)Z_{1,1}(w_{FJ})$. In the same manner, we get

$$\Psi_{k,0,0}(w_0,w_{FJ}) = \Psi_{k-1,1,1}(w_0,w_{FJ})\beta(w_0).$$

We shall also define (using a shorthand notation in which we omit w_0):

$$a^*(k,r_1,r_2) = \int_0^\infty e^{-w_0 t}\mathbb{P}\left(X(t)=r_1,\, Y(t)=r_2\right)\mathrm{d}\mathbb{P}\left(R_k<t\right),$$

$$b^*(k,r_1,h_2) = \int_0^\infty e^{-w_0 t}\mathbb{P}\left(X(t)=r_1,\, Y(t)\geq h_2\right)\mathrm{d}\mathbb{P}\left(R_k<t\right),$$

$$c^*(k,h_1,r_2) = \int_0^\infty e^{-w_0 t}\mathbb{P}\left(X(t)\geq h_1,\, Y(t)=r_2\right)\mathrm{d}\mathbb{P}\left(R_k<t\right),$$

$$d^*(k,h_1,h_2) = \int_0^\infty e^{-w_0 t}\mathbb{P}\left(X(t)\geq h_1,\, Y(t)\geq h_2\right)\mathrm{d}\mathbb{P}\left(R_k<t\right).$$

Hence, define

$$\Phi^*_{k,m,n} = \begin{pmatrix} a^*(k,0,0) & a^*(k,1,0) & \dots & a^*(k,m-1,0) & c^*(k,m,0) \\ a^*(k,0,1) & a^*(k,1,1) & \dots & a^*(k,m-1,1) & c^*(k,m,1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a^*(k,0,n-1) & a^*(k,1,n-1) & \dots & a^*(k,m-1,n-1) & c^*(k,m,n-1) \\ b^*(k,0,n) & b^*(k,1,n) & \dots & b^*(k,m-1,n) & d^*(k,m,n) \end{pmatrix}.$$

As before, for $1 \leq l \leq k-1$,

$$\Psi_{l,m,n} = trace\left(\bar{\Psi}_{l-1,m+1,n+1}\cdot\Phi_{m,n}\right),$$

and

$$\Psi_{k,h_1,h_2}(w_0,w_{FJ}) \equiv \mathbb{E}_{k,h_1,h_2}\left(e^{-w_0 S_0 - w_{FJ}S_{FJ}}\right) = trace\left(\bar{\Psi}_{k-1,h_1+1,h_2+1}\cdot\Phi^*_{k,h_1,h_2}\right).$$

EC.3. General service durations in the FJ stations

In §2, we assumed that service durations in all three stations of the model are exponential. In Appendix EC.2, we partially relaxed this assumption, considering general service durations in the first station. Here, we provide further relaxation of this assumption by allowing Erlang and hyperexponential service durations in the FJ stations. We note that the Erlang distribution can be used when the true service durations (according to data) have a coefficient of variation (CV) that is less than 1, while the hyperexponential service duration is useful when the CV is greater than 1. We start by analyzing the case of service durations with an Erlang distribution. For simplicity, we assume that service durations in the UFJ station have an $Erlang\left(m,\mu_1\right)$ distribution, while the LFJ station service durations are exponential with rate μ_2, and the first station's service durations are also exponential with rate μ_0 (the case where both FJ stations have an Erlang distribution immediately follows and hence is omitted here).

Assume that a tagged customer arrives to the first station and finds k customers at the first station, h_1 customers at the UFJ, and h_2 customers at the LFJ, where the customer in service in the UFJ is in the $j\in\{1,...,m\}$ service phase (we believe that it is reasonable to assume that once a customer's service starts, it is known in which service phase they are at all times). From the memoryless property of the exponential distribution, this means that the customer in service in the UFJ station will need to complete $m-j+1$ exponential phases in order to leave this station. Thus, at the arrival time of the tagged customer to the first station, the total number of *service phases* to be conducted, that is, that are already in the buffer of the UFJ server, is $\left(h_1-1\right)m+\left(m-j+1\right)=h_1m-j+1$. We define

$$\Psi_{k,h_1m-j+1,h_2}(w_0,w_{FJ})\equiv\mathbb{E}_{k,h_1m-j+1,h_2}\left[e^{-w_0S_0-w_{FJ}S_{FJ}}\right]=\int\limits_0^\infty e^{-w_0t}X_{k,h_1m-j+1,h_2}\mu_0e^{-\mu_0t}\,\mathrm{d}t, \tag{EC.1}$$

where X_{k,h_1m-j+1,h_2} is given by

$$\begin{aligned}X_{k,h_1m-j+1,h_2}=&\sum_{r_1=0}^{h_1m-j}\sum_{r_2=0}^{h_2-1}e^{-\mu_1t}\frac{\left(\mu_1t\right)^{r_1}}{r_1!}e^{-\mu_2t}\frac{\left(\mu_2t\right)^{r_2}}{r_2!}\Psi_{k-1,h_1m+(m-j+1)-r_1,h_2-r_2+1}(w_0,w_{FJ})\\&+\sum_{r_1=0}^{h_1m-j}\sum_{r_2=h_2}^{\infty}e^{-\mu_1t}\frac{\left(\mu_1t\right)^{r_1}}{r_1!}e^{-\mu_2t}\frac{\left(\mu_2t\right)^{r_2}}{r_2!}\Psi_{k-1,h_1m+(m-j+1)-r_1,1}(w_0,w_{FJ})\\&+\sum_{r_1=h_1m-j+1}^{\infty}\sum_{r_2=0}^{h_2-1}e^{-\mu_1t}\frac{\left(\mu_1t\right)^{r_1}}{r_1!}e^{-\mu_2t}\frac{\left(\mu_2t\right)^{r_2}}{r_2!}\Psi_{k-1,m,h_2-r_2+1}(w_0,w_{FJ})\\&+\sum_{r_1=h_1m-j+1}^{\infty}\sum_{r_2=h_2}^{\infty}e^{-\mu_1t}\frac{\left(\mu_1t\right)^{r_1}}{r_1!}e^{-\mu_2t}\frac{\left(\mu_2t\right)^{r_2}}{r_2!}\Psi_{k-1,m,1}(w_0,w_{FJ}).\end{aligned} \tag{EC.2}$$

Using the following notation (and a shorthand notation in which we omit the w_0, w_{FJ}), we get

$$\begin{aligned}
a(r_1,r_2) &\equiv \int_0^\infty e^{-(w_0+\mu_1+\mu_2)t}\frac{(\mu_1 t)^{r_1}}{r_1!}\frac{(\mu_2 t)^{r_2}}{r_2!}\mu_0 e^{-\mu_0 t}\,\mathrm{d}t,\\
b(r_1,h_2) &\equiv \sum_{r_2=h_2}^{\infty} a(r_1,r_2) = \int_0^\infty e^{-(w_0+\mu_1+\mu_2)t}\sum_{r_2=h_2}^{\infty}\frac{(\mu_1 t)^{r_1}}{r_1!}\frac{(\mu_2 t)^{r_2}}{r_2!}\mu_0 e^{-\mu_0 t}\,\mathrm{d}t,\\
c(h_1 m-j+1,r_2) &\equiv \sum_{r_1=h_1 m-j+1}^{\infty} a(r_1,r_2),\\
d(h_1 m-j+1,h_2) &\equiv \sum_{r_1=h_1 m-j+1}^{\infty}\sum_{r_2=h_2}^{\infty} a(r_1,r_2),
\end{aligned}$$

and the recursive formula for $\Psi_{k,h_1 m-j+1,h_2}$, given by Equations (EC.1) and (EC.2) (for the case where $k,h_1,h_2>0$), becomes

$$\begin{aligned}
\Psi_{k,h_1 m-j+1,h_2} = &\sum_{r_1=0}^{h_1 m-j}\sum_{r_2=0}^{h_2-1}\Psi_{k-1,h_1 m+(m-j+1)-r_1,h_2-r_2+1}\cdot a(r_1,r_2)\\
&+\sum_{r_1=0}^{h_1 m-j}\Psi_{k-1,h_1 m+(m-j+1)-r_1,1}\cdot b(r_1,h_2)\\
&+\sum_{r_2=0}^{h_2-1}\Psi_{k-1,m,h_2-r_2+1}\cdot c(h_1 m-j+1,r_2)+\Psi_{k-1,m,1}\cdot d(h_1 m-j+1,h_2).
\end{aligned} \tag{EC.3}$$

We now consider the case of hyperexponential service durations in the UFJ. As before, the case where both FJ stations have a hyperexponential service durations, or where one FJ station has Erlang service durations and the other has hyperexponential service durations, are an immediate extension, and hence are omitted here. Assume that in the UFJ station, each customer service is $Exp\left(\mu_1^1\right)$ with probability p_1, or $Exp\left(\mu_1^2\right)$ with probability $p_2=1-p_1$. We also assume that once a customer starts to receive service in the UFJ station, we know the type of their service duration (that is, whether it is with rate μ_1^1 or with rate μ_1^2). We shall denote customers having an $Exp\left(\mu_1^i\right)$ service duration as *type* i customers, where $i=1,2$.

A tagged customer entering the first single-server station observes the three queue lengths and the *type* of the customer in service in the UFJ station. Let $N_i(t)$ be the number of service completions at the UFJ station given that the customer in service in the UFJ station is of type $i=1,2$. Let A_i denote the event: the (r_1+1)th customer at the UFJ station is of type $i=1,2$, and let A_i^{Curr}

denote the event: the current customer in service in the first station will be of type $i = 1,2$ on arrival to service at the UFJ station. We can now define

$$\Psi_{k,h_1,h_2,i}(w_0,w_{FJ}) \equiv \mathbb{E}_{k,h_1,h_2,i}\left[e^{-w_0 S_0 - w_{FJ} S_{FJ}}\right] = \int\limits_0^\infty e^{-w_0 t} X_{k,h_1,h_2,i} \mu_0 e^{-\mu_0 t}\,\mathrm{d}t, \qquad \text{(EC.4)}$$

where $X_{k,h_1,h_2,i}$ is given by

$$\begin{aligned} X_{k,h_1,h_2,i} &= \sum_{r_1=0}^{h_1-1}\sum_{r_2=0}^{h_2-1} e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!}\sum_{j=1}^{2}\mathbb{P}\left(N_i(t)=r_1|A_j\right) p_j \Psi_{k-1,h_1-r_1+1,h_2-r_2+1,j}(w_0,w_{FJ}) \\ &+ \sum_{r_1=0}^{h_1-1}\sum_{r_2=h_2}^{\infty} e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!}\sum_{j=1}^{2}\mathbb{P}\left(N_i(t)=r_1|A_j\right) p_j \Psi_{k-1,h_1-r_1+1,1,j}(w_0,w_{FJ}) \\ &+ \sum_{r_2=0}^{h_2-1} e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!}\sum_{j=1}^{2}\mathbb{P}\left(N_i(t)\geq h_1|A_j^{Curr}\right) p_j \Psi_{k-1,1,h_2-r_2+1,j}(w_0,w_{FJ}) \\ &+ \sum_{r_2=h_2}^{\infty} e^{-\mu_2 t}\frac{(\mu_2 t)^{r_2}}{r_2!}\sum_{j=1}^{2}\mathbb{P}\left(N_i(t)\geq h_1|A_j^{Curr}\right) p_j \Psi_{k-1,1,1,j}(w_0,w_{FJ}). \end{aligned}$$

It is left to compute $\mathbb{P}\left(N_i(t)=r_1|A_j\right)$ and $\mathbb{P}\left(N_i(t)\geq h_1|A_j^{Curr}\right)$, where $i,j=1,2$. To do so, we define the events: $C_{l,i}^{n}$ as there are l customers of type i among the first n customers at the UFJ station, $i=1,2$, $l=1,...,n$, and $n=1,...,h_1-1$. For simplicity, we denote $i^c := 3-i$. Therefore, for $r_1>0$,

$$\begin{aligned} \mathbb{P}\left(N_i(t)=r_1|A_j\right) &= \sum_{l=1}^{r_1}\mathbb{P}\left(N_i(t)=r_1|C_{l,i}^{r_1},A_j\right)\mathbb{P}\left(C_{l,i^c}^{r_1}|A_j\right) \\ &= \sum_{l=1}^{r_1}\mathbb{P}\left(N_i(t)=r_1|C_{l,i}^{r_1},A_j\right)\binom{r_1-1}{l-1}p_i^{l-1}\left(1-p_i\right)^{r_1-l}, \end{aligned}$$

while for $l<r_1$,

$$\mathbb{P}\left(N_i(t)=r_1|C_{l,i}^{r_1},A_j\right) = \int\limits_{s=0}^{t}\int\limits_{u=0}^{t-s}\frac{\left(\mu_1^i\right)^l s^{l-1}e^{-\mu_1^i s}}{(l-1)!}\frac{\left(\mu_1^{i^c}\right)^{r_1-l}u^{r_1-l-1}e^{-\mu_1^{i^c}u}}{(r_1-l-1)!}e^{-\mu_1^j(t-(s+u))}\,\mathrm{d}u\,\mathrm{d}s.$$

This formulation is derived from conditioning on the case where the service of the l type i customers takes exactly s units of time; the service of the (r_1-l) type i customers takes exactly u time units, and the service time of the (r_1+1)th type j customer exceeds $t-(s+u)$ time units. In the case where $r_1>0$ and $l=r_1$, we get

$$\mathbb{P}\left(N_i(t)=r_1|C_{l,i}^{r_1},A_j\right) = \int\limits_{s=0}^{t}\frac{\left(\mu_1^i\right)^{r_1}s^{r_1-1}e^{-\mu_1^i s}}{(r_1-1)!}e^{-\mu_1^j(t-s)}\,\mathrm{d}s,$$

while in the case where $r_1 = 0$, we simply get

$$\mathbb{P}\left(N_i(t) = 0 | A_j\right) = e^{-\mu_1^i t}.$$

In the same manner,

$$\begin{aligned}\mathbb{P}\left(N_i(t) \geq h_1 | A_j^{Curr}\right) &= \sum_{l=1}^{h_1} \mathbb{P}\left(N_i(t) \geq h_1 | C_{l,i}^{h_1}, A_j^{Curr}\right) \mathbb{P}\left(C_{l,i}^{h_1} | A_j^{Curr}\right) \\ &= \sum_{l=1}^{h_1} \mathbb{P}\left(N_i(t) \geq h_1 | C_{l,i}^{h_1}\right) \binom{h_1 - 1}{l - 1} p_i^{l-1} \left(1 - p_i\right)^{h_1 - l},\end{aligned}$$

while, for $l < h_1$,

$$\mathbb{P}\left(N_i(t) \geq h_1 | C_{l,i}^{h_1}\right) = \int_{s=0}^{t} \int_{u=0}^{t-s} \frac{\left(\mu_1^i\right)^l s^{l-1} e^{-\mu_1^i s}}{(l-1)!} \frac{\left(\mu_1^{i^c}\right)^{h_1-l} u^{h_1-l-1} e^{-\mu_1^{i^c} u}}{(h_1-l-1)!} \mathrm{d}u \, \mathrm{d}s,$$

and for $l = h_1$,

$$\mathbb{P}\left(N_i(t) \geq h_1 | C_{l,i}^{h_1}\right) = \int_{s=0}^{t} \frac{\left(\mu_1^i\right)^{h_1} s^{h_1-1} e^{-\mu_1^i s}}{(h_1-1)!} \mathrm{d}s.$$

EC.4. Fork-join with delay node

We consider a model with one single-server queue (with either exponential or general service time distribution), followed by an FJ part that consists of one single- or multiple-server queue with exponentially distributed service durations and one infinite-server queue with some general service durations. We denote the CDF of this distribution by $G(\cdot)$. Assuming that the delay node (that is, the infinite-server queue) is the UFJ station, we get, as in the model of Boxma and Daduna (2014), that the joint LST of the sojourn times in the first station and in the FJ part is given by the following recursive formula:

$$\Psi_{k,h_2}(w_0, w_{FJ}) = \int_0^\infty e^{-w_0 t} \left\{ \sum_{l=0}^{h_2-1} e^{-\mu_2 t} \frac{(\mu_2 t)^l}{l!} \Psi_{k-1,h_2-l+1}(w_0, w_{FJ}) + \sum_{l=h_2}^{\infty} e^{-\mu_2 t} \frac{(\mu_2 t)^l}{l!} \Psi_{k-1,1}(w_0, w_{FJ}) \right\} \mathrm{d}B(t).$$

However,

$$\Psi_{0,h_2}(w_0, w_{FJ}) = \int_0^\infty e^{-w_0 t} \left\{ \sum_{l=0}^{h_2-1} e^{-\mu_2 t} \frac{(\mu_2 t)^l}{l!} Z_{h_2-l+1}(w_{FJ}) + \sum_{l=h_2}^{\infty} e^{-\mu_2 t} \frac{(\mu_2 t)^l}{l!} Z_1(w_{FJ}) \right\} \mathrm{d}B(t),$$

and

$$\Psi_{0,0}(w_0, w_{FJ}) = \beta(w_0) Z_1(w_{FJ}).$$

Here $\beta(\cdot)$ is the LST of the general service times in the first station, and $Z_x(w_{FJ})$ is the LST of the maximum of two independent random variables, one with general distribution (corresponding to the sojourn time in the delay node) and one with Erlang distribution (corresponding to the sojourn time of a customer entering a Markovian multiple/single-server queue with x customers already there). Hence

$$Z_x(w_{FJ}) \equiv \mathbb{E}_x\left(e^{-w_{FJ}S_{FJ}}\right) = \int\limits_0^\infty e^{-w_{FJ}t}\,\mathrm{d}F_{S_{FJ}|x}(t),$$

where

$$\begin{aligned} F_{S_{FJ}|x}(t) &= \mathbb{P}\left(S_{FJ} \le t | h_2 = x\right) = \mathbb{P}\left(S_1 \le t,\ S_2 \le t | h_2 = x\right) \\ &= \mathbb{P}\left(S_2 \le t | h_2 = x\right)\mathbb{P}\left(S_1 \le t\right) = \left(1 - \sum_{m=0}^{x-1} e^{-\mu_2 t}\frac{(\mu_2 t)^m}{m!}\right) G(t). \end{aligned}$$

EC.5. Service time estimation

In order to check the influence of queue length on service rates, we first need to evaluate service durations. Our data includes a single timestamp for each patient's activity, which is assumed to be the activity completion time. We do not have data on activity starting times; hence, estimations for activity durations were needed. In the ED examined, walking and acute patients are treated by different doctors and nurses. Thus, in order to calculate service rates at the nurse attendance station and at the doctor attendance station, we separated the population into walking patients and acute patients. We did not make this separation when estimating service rates at the lab testing station as it provides services to all types of patients.

To estimate service rates, we essentially estimated the service duration for each patient and then inverted the average service duration to get the desired service rate. We explain our service duration estimation methodology for nurse attendance. Note that all patients' first activity in the ED is administrative reception and the second is nurse attendance. Let Patient x be our tagged customer, and let Patient y be the last patient to go through nurse attendance prior to Patient x. We distinguish between two cases; see Figure EC.1 for illustration. In Case 1, Patient x's reception timestamp was before Patient y's nurse attendance timestamp. In this case, when Patient x arrived to the station, the nurse was busy; hence, Patient x's service duration (at the nurse attendance station) will be "Patient x's nurse attendance timestamp" minus "Patient y's nurse attendance timestamp." Otherwise, Case 2 applies, in which the nurse is assumed to be idle. In this case, Patient x's service duration (at the nurse attendance station) will be "Patient x's nurse attendance timestamp" minus "Patient x's reception timestamp."

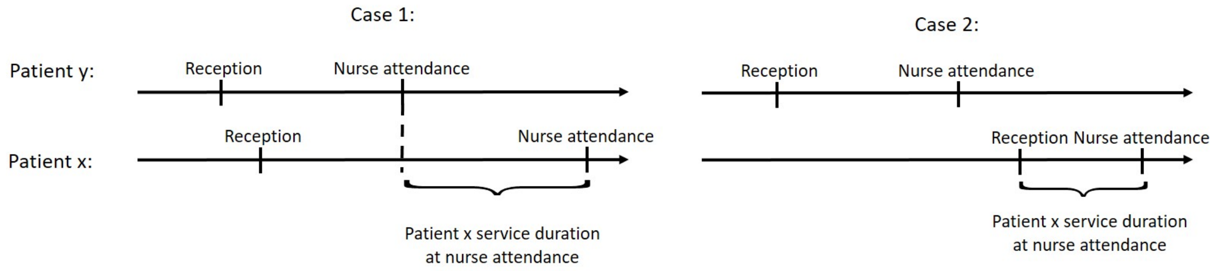

Figure EC.1 Estimating service duration, an example.