June 2013

WORKING PAPER SERIES

2013-MAN-02

A Markov model for measuring service levels in nonstationary G(t)/G(t)/s(t) + G(t) queues

Stefan CREEMERS
IESEG School of Management (LEM-CNRS)

Mieke DEFRAEYE
Research Center for Operations Management, KU Leuven

Inneke VAN NIEUWENHUYSE
Research Center for Operations Management, KU Leuven

IESEG School of Management
Lille Catholic University
3, rue de la Digue
F-59000 Lille
www.ieseg.fr
Tel: 33(0)3 20 54 58 92
Fax: 33(0)3 20 57 48 55

A Markov model for measuring service levels in nonstationary $G(t)/G(t)/s(t) + G(t)$ queues

Stefan Creemers*, Mieke Defraeye†, Inneke Van Nieuwenhuyse†

Abstract

We present a Markov model to approximate the queueing behavior at the $G(t)/G(t)/s(t) + G(t)$ queue with exhaustive discipline and abandonments. The performance measures of interest are: (1) the average number of customers in queue, (2) the variance of the number of customers in queue, (3) the average number of abandonments and (4) the virtual waiting time distribution of a customer when arriving at an arbitrary moment in time. We use acyclic phase-type distributions to approximate the general interarrival, service and abandonment time distributions. An efficient, iterative algorithm allows the accurate analysis of small- to medium-sized problem instances. The validity and accuracy of the model are assessed using a simulation study.

1 Introduction

Many service systems exhibit a cyclic demand for service. E.g., in call centers, emergency departments, banks and retail stores, the number of arrivals typically displays a daily, weekly or monthly recurring pattern. Figure 2, for instance, displays the daily fluctuations in arrival rate at the emergency department of a regional hospital in Belgium; other examples can be found in Green et al. [2006], Brown et al. [2005] and Dietz [2011], among others.

*IESEG School of Management (LEM-CNRS), Rue de la Digue 3, 59000 Lille, France s.creemers@ieseg.fr

†Research Center for Operations Management, Department of Decision Sciences and Information Management, KU Leuven, Naamsestraat 69, 3000 Leuven, Belgium firstname.lastname@kuleuven.be

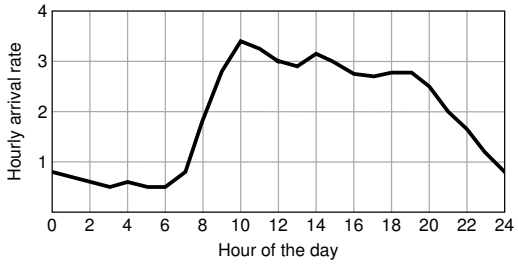


Figure 1: Hourly average arrival rates at the emergency department of a Belgian regional hospital

Apart from the time-varying nature of demand, additional complexities may arise because of (1) the presence of customer impatience, which causes customers to abandon before receiving service if their waiting time is too long and (2) the general distribution of service and abandonment times. The Poisson assumption that is common in the literature tends to be invalid in realistic settings; for instance, Brown et al. [2005] report a lognormal distribution and Castillo et al. [2009] report Erlang distributed service times in a call-center context. Moreover, many existing models in the literature implicitly assume a preemptive service discipline, such that service is interrupted and customers rejoin the queue when a server is scheduled to leave. An exhaustive service policy, where a customer's service is completed even if this requires the server to work past his scheduled time, is often more appropriate (especially in service systems with human customers and servers). This feature, however, is frequently overlooked in the literature [Ingolfsson et al., 2007, Chen and Henderson, 2001].

Performance analysis for systems with time-varying arrivals is highly important when making capacity decisions. Capacity planning models rely on a performance evaluation method as a subroutine to assess the solution quality of any given capacity vector. We refer to Green et al. [2007], Whitt [2007], Defraeye and Van Nieuwenhuyse [2011] for extensive reviews on capacity planning in time-varying systems.

This article presents a Markov model that approximates the transient and steady-state behavior of the $G(t)/G(t)/s(t) + G(t)$ queue with exhaustive discipline and time-varying arrival, service and abandonment rates. The model enables the evaluation of the following (time-varying) performance metrics: (1) the expected queue length, (2) the variance of the queue length,

(3) the expected number of abandonments and (4) the virtual waiting time distribution of a customer when arriving at an arbitrary moment in time. The suggested approach extends the randomization method of Ingolfsson et al. [2007] and Ingolfsson [2005], which targets $M(t)/M/s(t)$ queues with an exhaustive service policy (an outline on how to include customer impatience is provided, yet not implemented). To the best of our knowledge, this is the first analytical model that studies a queue with an exhaustive service policy, customer impatience and generally distributed (time-varying) arrival, service and abandonment rates. The approach is intended for small- to medium-sized systems that have human servers (e.g., banks, retail stores or small-scale call centers). For larger problem instances, the computational cost increases substantially. The model is validated and evaluated by means of a simulation study.

The remainder of the article is organized as follows: Section 2 starts with a brief overview of the literature on performance measurement in systems with time-varying arrivals. In Section 3, we present an in-depth description of the Markov model itself. Section 4 evaluates the accuracy and validity of the model by means of a computational experiment. In a final section (Section 5), we highlight the main conclusions and suggest directions for further research.

2 Related literature

Previous work has mainly focused on systems with time-varying arrival rates. In this section, we provide a brief overview of the (most frequently) used methods for performance analysis in such systems.

Stationary approximations are by far the most widely adopted approach. The arrival rate that is fed into the stationary model can be, for instance, the instantaneous arrival rate (as in the Pointwise Stationary Approximation or PSA [Green et al., 1991, Green and Kolesar, 1991, Whitt, 1991]) or the average arrival rate over a given interval (Stationary Independent Period-by-Period or SIPP [Green et al., 2001, Whitt, 1991]). However, time-varying systems typically display a time lag (or congestion lag): peaks in actual offered load lag the arrival rate peaks, with an amount that is proportional to the expected service time [Green and Kolesar, 1995, Thompson, 1993]. Accounting for this lag can greatly improve the accuracy of SIPP and PSA, particularly when service times are long (see the lagged variants of SIPP

and PSA [Green and Kolesar, 1997, 1995, Green et al., 2001]). The Modified Offered Load (MOL) approximation accounts for the congestion lag by relying on analytically tractable results for infinite server queues, which can be found in Eick et al. [1993a,b]. Further details on MOL can be found in Feldman et al. [2008], Jennings et al. [1996], Liu and Whitt [2009], Jagerman [1975], Massey and Whitt [1994, 1997] and Davis et al. [1995]. Though stationary approximations are straightforward and generally applicable, additional challenges may arise in complex systems, for which the stationary model itself is intractable. For instance, the applicability of MOL to the $M(t)/G/s(t)+G$ model necessarily relies on the availability and accuracy of approximations for the corresponding stationary $M/G/s+G$ model (see Whitt [2005] and Iravani and Balcioğlu [2008]). We refer to Green et al. [2007], Whitt [2007] and Defraeye and Van Nieuwenhuyse [2011] for further references on the stationary approximations available in the literature.

For the $M(t)/M/s(t)$ system, performance can be evaluated by numerically integrating the Chapman-Kolmogorov forward equations, a set of ordinary differential equations (ODEs) that describe the behavior of the system (see Gross et al. [2008] for general background; Ingolfsson et al. [2007] and Green and Soares [2007] provide a more thorough discussion). This can be achieved using an ODE-solver such as the Euler or Runge-Kutta ODE solver from the Matlab ODE Suite Shampine and Reichelt [1997]. Ingolfsson et al. [2007] show that this approach requires substantial computational effort and suggest using the randomization approach instead: this enables a drastic reduction in computational effort, at the cost of a slightly lower accuracy. The randomization (or uniformization) approach was originally developed for stationary systems [Jensen, 1953, Grassmann, 1977, Gross and Miller, 1984], but can be applied successfully to nonstationary queues [Ingolfsson, 2005, Ingolfsson et al., 2007]. In general, the numerical integration of ODEs as well as randomization require the use of exponential distributions in order to obtain accurate results. Furthermore, these approaches currently do not take into account abandonments (though Ingolfsson [2005] provides an outline on how to accommodate abandonments in the randomization approach).

Closure approximations [Rothkopf and Oren, 1979, Clark, 1981, Taaffe and Ong, 1987] approximate the set of forward differential equations by just two differential equations (one for the mean and one for the variance of the number in system at each time instant). However, as shown in Ingolfsson et al. [2007], the approach is cumbersome to implement and is dominated by other methods (such as MOL or randomization) in terms of both accuracy

and computation speed.

Discrete-time modeling (DTM) is used for performance evaluation of systems with general service time distributions [Chassioti and Worthington, 2004, Brahimi, 1990, Brahimi and Worthington, 1991, Wall and Worthington, 1994, 2007]. This approach approximates the general service process by means of a discrete process using a two-moment matching technique [Brahimi, 1990, Brahimi and Worthington, 1991]. Wall and Worthington [2007] report distinct advantages over stationary approximations such as MOL and PSA, particularly when temporal overloading is present. The complexity and computational effort of DTM, however, increase drastically with the number of servers; Wall and Worthington [2007] propose an approximation method to mitigate this effect. Note that the current DTM articles all study the $M(t)/G/s$ system (i.e., they assume a constant number of servers and no abandonments).

Deterministic fluid models (intended for systems that do not display stochasticity) can be used as approximations to derive time-dependent performance in stochastic systems. These methods rely on so-called "fluid scaling": the system is scaled up (e.g., by multiplying the arrival rates and the number of servers by the same factor) such that the stochastic randomness decreases in importance, relative to the system dynamics (see Helber and Henken [2010] for an example). Fluid approximations are particularly useful to assess performance in systems that are temporarily overloaded [Whitt, 2006a], but may fail to capture system dynamics accurately in underloaded systems [Aguir et al., 2004, Altman et al., 2001, Jiménez and Koole, 2004]. Liu and Whitt [2010] suggest an approach that works for overloaded as well as underloaded systems (separate models are applied in both situations). Additional literature on the use of fluid approximations for Markovian models, can be found in Mandelbaum et al. [1995, 1998, 1999a,b, 2002], Ridley et al. [2003] and Jiménez and Koole [2004]. For systems with general service and/or abandonment time distributions, we refer to the more recent work of Whitt [2006a] on $G(t)/GI/s + GI$ models (with state-dependent arrival rates), Liu and Whitt [2010, 2011b, 2012a,b] on the $G(t)/GI/s(t)+GI$ queue, Liu and Whitt [2011a] for a network of $G(t)/M(t)/s(t) + GI(t)$ queues and references therein. A key characteristic of fluid models is that arrivals and departures are considered as continuous flows, rather than discrete processes (an assumption that becomes more acceptable as the number of servers increases). Although Liu and Whitt [2010] report reasonably accurate results for a system with 20 servers, the assumption of fluid scaling renders these

approximations less applicable to small-scale settings where the discreteness of capacity is an essential characteristic of the system.

Finally, discrete-event simulation is frequently used (a comprehensive textbook can be found in [Law and Kelton, 2000]). The appeal of simulation lies in its inherent flexibility to evaluate the performance of virtually any given system. As such, simulation proves particularly useful in settings that are analytically intractable. On the downside, simulation tends to be rather time-consuming, both in terms of runtime and time required to build the model. Although simulation models are commonly dedicated and context-specific (e.g., [McGuire, 1994, García et al., 1995, Evans et al., 1996, Takakuwa and Shiozaki, 2004, Hung et al., 2007, Ahmed and Alkhamis, 2009] describe simulation applications in EDs with time-varying arrivals) efforts are made to develop generic simulation models (e.g., [Pitt, 1997, Sinreich and Marmor, 2004, Fletcher et al., 2007a,b, Gunal and Pidd, 2009]). In this article, we use discrete-event simulation to validate the Markov model.

3 Model

In this section we develop an approximation for the $G(t)/G(t)/s(t) + G(t)$ queue with exhaustive discipline and abandonments. Analogous to the DTM models discussed in the previous section, our model observes the state of the system at discrete moments in time. Unlike the DTM models, however, we do not rely on discrete distributions, but use continuous-time phase-type (PH) distributions to match the continuous system processes. Because each phase of a continuous-time PH distribution has an exponentially distributed visiting time, the system processes are approximated by mixtures of exponential distributions. A notable downside of DTM is that it requires to keep track of each server individually. In our approach, however, this is not the case. Due to the memoryless property of the exponential distribution, it suffices to keep track of the number of servers associated with a given phase of the service process.

In what follows, we first define the basic processes that govern the system (Section 3.1) and introduce the phase-type distributions that are used to model these basic processes (Section 3.2). Next, we define a counting process (Section 3.3) and a procedure to determine the probability that a given number of customers advances a phase (Section 3.4). In the last subsections, we present the model itself (Sections 3.5 and 3.7) and discuss the performance

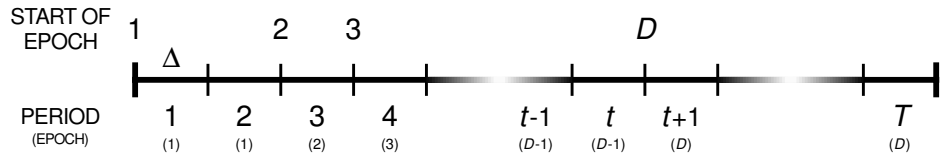


Figure 2: Division of time

measures (Section 3.6).

3.1 Basic Processes

We observe the state of the system at discrete, equidistant moments in time. The time between observation moments determines the granularity (and hence the precision) of the model and is denoted by Δ. Define $\mathbf{T} = \{1, \ldots, T\}$, the set of periods (where T is the last period; the period that marks the end of the time horizon). There are four basic processes: (1) the arrival process, (2) the service process, (3) the abandonment process and (4) the staffing process. At the start of any given period, these processes are allowed to change. If such a change takes place for at least one of the processes, the start of the period corresponds with the start of a so-called "epoch". Let $\mathbf{D}^{(\cdot)} = \left\{1, 2, \ldots, D^{(\cdot)}\right\}$ denote the set of epochs for a process $(\cdot)$, where $D^{(\cdot)}$ is the total number of epochs over the time horizon. For each process $(\cdot)$, define t_d, the period at which epoch d starts, where $t_1 = 0$ and $t_i < t_j \leq t_{D^{(\cdot)}} \leq T$ for all $i, j : i < j \leq D^{(\cdot)}$. Function $\phi_t^{(\cdot)} = i$ maps a period t onto an epoch i, where i is the ongoing epoch at the start of period t (i.e., there exists no epoch j for which $t_i < t_j \leq t$). Figure 2 further illustrates the division of time.

Each epoch of the arrival, service and abandonment process is characterized by a distribution $G_d^{(\cdot)}$ that has mean $\mu_d^{(\cdot)}$ and standard deviation $\sigma_d^{(\cdot)}$. As such:

- $\mu_d^{(\mathrm{I})}$ and $\sigma_d^{(\mathrm{I})}$ represent the mean and standard deviation of the interarrival times during an epoch $d : d \in \mathbf{D}^{(\mathrm{I})}$,
- $\mu_d^{(\mathrm{II})}$ and $\sigma_d^{(\mathrm{II})}$ represent the mean and standard deviation of the service times during an epoch $d : d \in \mathbf{D}^{(\mathrm{II})}$,

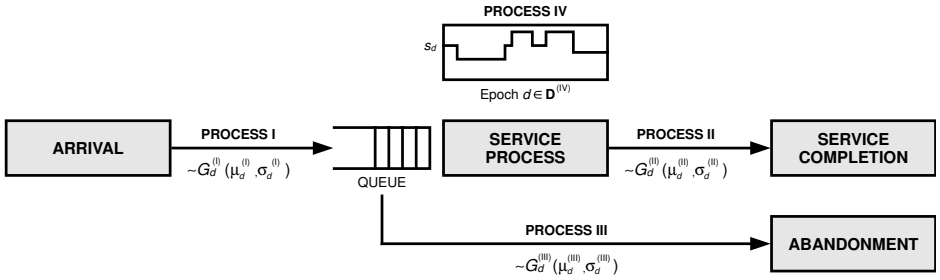


Figure 3: The $G(t)/G(t)/s(t)+G(t)$ queueing system

- $\mu_d^{(\mathrm{III})}$ and $\sigma_d^{(\mathrm{III})}$ represent the mean and standard deviation of the abandonment times during an epoch $d : d \in \mathbf{D}^{(\mathrm{III})}$.

Each epoch of the staffing process represents a so-called staffing interval (during which staffing remains unchanged) and is associated with a number of servers $s_d : d \in \mathbf{D}^{(\mathrm{IV})}$. In the remainder of this article, Roman numerals I, II, III and IV are used to label the arrival, service, abandonment and staffing process respectively. Figure 3 summarizes the single-stage multiserver service system with time-varying interarrival times, service times, abandonment times and staffing levels.

3.2 Phase-type distributions

We adopt continuous-time PH distributions to approximate the general interarrival, service and abandonment time distributions. Continuous-time PH distributions use exponentially-distributed building blocks to approximate any positive-valued continuous distribution with arbitrary precision (see Neuts [1981], Latouche [1999] and Osogami [2005] for further details on PH type distributions). More formally, a PH distribution is the distribution of time until absorption in a Markov chain with absorbing state 0 and state space $\{0, 1, \ldots, Z-1, Z\}$. It is fully characterized by parameters $\boldsymbol{\tau}$ and $\mathbf{Z}$. $\boldsymbol{\tau}$ is the vector of probabilities to start the process in any of the Z transient states (i.e., phases) and $\mathbf{Z}$ is the transient state transition matrix. The infinitesimal generator of the Markov chain representing the PH distribution is:

$$\mathbf{Q} = \begin{pmatrix} 0 & \mathbf{0} \\ \mathbf{t} & \mathbf{Z} \end{pmatrix},$$

where $\mathbf{0}$ is a zero matrix of appropriate dimension and $\mathbf{t} = -\mathbf{Ze}$ (with $\mathbf{e}$ a vector of ones of appropriate size).

Various techniques exist to approximate a given distribution by means of a PH distribution. In this article, we adopt a two-moment matching procedure that minimizes the required number of phases. Let C^2 denote the squared coefficient of variation of the distribution we want to approximate:

$$C^2 = \sigma^2 \mu^{-2}. \tag{1}$$

We distinguish three cases: (1) $C^2 = 1$, (2) $C^2 > 1$ and (3) $C^2 < 1$. In the first case, we approximate the distribution by means of an exponential distribution with rate parameter $\lambda = \mu^{-1}$. The parameters of the corresponding PH distribution are:

$$\begin{aligned} \boldsymbol{\tau} &= 1, \\ \mathbf{Z} &= (-\lambda). \end{aligned}$$

In the second case, we use a two-phase Coxian distribution where the rate parameter of the first phase is determined by means of a scaling factor κ:

$$\lambda_1 = \frac{1}{\mu\kappa}. \tag{2}$$

The expected value of the two-phase Coxian distribution is:

$$\mu = \lambda_1^{-1} + \beta\lambda_2^{-1}, \tag{3}$$

where λ_2 is the exponential rate parameter of the second phase and β is the probability of visiting the second phase. The variance of the two-phase Coxian distribution is:

$$\sigma^2 = \lambda_1^{-2} + 2\beta\lambda_2^{-2} - \beta^2\lambda_2^{-2}. \tag{4}$$

When deriving parameters λ_2 and β as a function of parameters μ, C^2 and κ, we obtain:

$$\lambda_2 = \frac{2(\kappa - 1)}{\mu(2\kappa - 1 - C^2)}, \tag{5}$$

$$\beta = \frac{2(\kappa - 1)^2}{1 + C^2 - 2\kappa}. \tag{6}$$

The parameters of the corresponding PH distribution are:

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{e}_1, \\ \mathbf{Z} &= \begin{pmatrix} -\lambda_1 & \beta\lambda_1 \\ 0 & -\lambda_2 \end{pmatrix}, \end{aligned}$$

where $\mathbf{e}_1$ is a single-entry vector of appropriate size that is populated with zeroes except for the first entry, which equals one. In the third case, we use a hypo-exponential distribution (a series of exponential distributions whose parameters are allowed to differ; a generalization of the Erlang distribution). The number of required phases equals:

$$Z = \lceil C^{-2} \rceil. \tag{7}$$

We assume that the first $Z-1$ phases of the hypo-exponential distribution are exponentially distributed with rate parameter λ_1. The last phase is exponentially distributed with rate parameter λ_2. The expected value and variance of the hypo-exponential distribution are:

$$\mu = (Z-1)\,\lambda_1^{-1} + \lambda_2^{-1}, \tag{8}$$
$$\sigma^2 = (Z-1)\,\lambda_1^{-2} + \lambda_2^{-2}. \tag{9}$$

When deriving parameters λ_1 and λ_2 as a function of parameters μ, C^2 and Z, we obtain:

$$\lambda_1 = \frac{(Z-1) - \sqrt{(Z-1)\,(ZC^2-1)}}{\mu\,(1-C^2)}, \tag{10}$$
$$\lambda_2 = \frac{1 + \sqrt{(Z-1)\,(ZC^2-1)}}{\mu\,(1-ZC^2+C^2)}. \tag{11}$$

The parameters of the corresponding PH distribution are:

$$\boldsymbol{\tau} = \mathbf{e}_1,$$
$$\mathbf{Z} = \begin{pmatrix} -\lambda_1 & \lambda_1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -\lambda_1 & \lambda_1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -\lambda_1 & \lambda_1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -\lambda_1 & \lambda_1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & -\lambda_2 \end{pmatrix}.$$

For the three cases, Z equals 1, 2 and $\lceil C^{-2} \rceil$ respectively. Figure 4 provides an overview of the PH distributions that are used in this article.

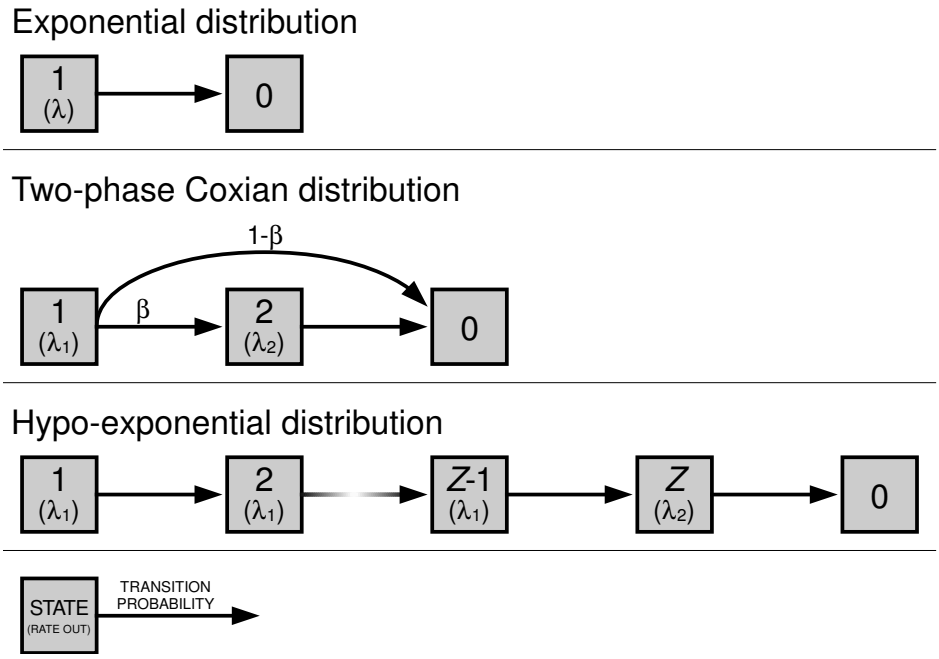


Figure 4: Overview of PH distributions

3.3 Counting process

We use a counting process to obtain $\Pr(x, v|u, d)$, the probability of having x arrivals during an interval t (of length Δ) for which $\phi_t^{(\mathrm{I})} = d$, and an arrival process at final phase v given that the arrival process starts in phase u and is modeled using a PH distribution with parameters $\boldsymbol{\tau}_d^{(\mathrm{I})}$ and $\mathbf{Z}_d^{(\mathrm{I})}$.

The counting process has continuous-time rate matrix [Ramaswami, 1988]:

$$\mathbf{Q}_d = \begin{pmatrix} \mathbf{L}_d & \mathbf{F}_d & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{L}_d & \mathbf{F}_d & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{L}_d & \mathbf{F}_d & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L}_d & \cdots \\ \cdots & \cdots & \cdots & \cdots & \ddots \end{pmatrix},$$

where $\mathbf{L}_d = \mathbf{Z}_d^{(\mathrm{I})}$ and $\mathbf{F}_d = \mathbf{t}_d^{(\mathrm{I})} \left(\boldsymbol{\tau}_d^{(\mathrm{I})}\right)^\top$. $\mathbf{C}_d$ holds the transition probabilities of the counting process during an interval of length Δ during epoch d:

$$\begin{aligned} \mathbf{C}_d &= e^{\Delta \mathbf{Q}_d}, && (12) \\ &= \sum_{i=0}^{\infty} \frac{\Delta^i}{i!} \mathbf{Q}_d^i, && (13) \\ &= e^{-\Delta \lambda_{d,\max}} \sum_{i=0}^{\infty} \frac{(\Delta \lambda_{d,\max})^i}{i!} \mathbf{P}_d^i, && (14) \end{aligned}$$

where $\lambda_{d,\max} = -\min(\mathrm{Diag}(\mathbf{Z}))$ and $\mathbf{P}_d$ is obtained as follows:

$$\mathbf{P}_d = \frac{\mathbf{Q}_d}{\lambda_{d,\max}} + \mathbf{I}, \qquad (15)$$

where $\mathbf{I}$ is an identity matrix of appropriate dimension.

The first block row of $\mathbf{C}_d$ holds the distribution of the number of arrivals (i.e., probabilities $\Pr(x, v|u, d)$). In order to obtain the first block row of $\mathbf{C}_d$, it suffices to compute the first block row of $\mathbf{P}_d^i$ for all $i \geq 0$; this can be done by means of a simple recursion.

3.4 Procedure to determine the probability of advancing a phase

The following procedure is used to determine the probability to advance a phase in the service or abandonment process. Let $\Pr(y|x, u, d)^{(\cdot)}$ denote the

probability that y customers successfully complete phase u of process $(\cdot)$ during an interval of length Δ, given that x customers are present in phase u at the start of the interval and the process is modeled using a PH distribution with parameters $\boldsymbol{\tau}_d^{(\cdot)}$ and $\mathbf{Z}_d^{(\cdot)}$.

In order to compute $\Pr(y|x,u,d)^{(\cdot)}$, we use a Markov process that has infinitesimal generator:

$$\mathbf{Q}_{d,u}^{(\cdot)} = \begin{pmatrix} -y\lambda_{d,u}^{(\cdot)} & y\lambda_{d,u}^{(\cdot)} & \cdots & 0 & 0 & 0 \\ -(y-1)\lambda_{d,u}^{(\cdot)} & (y-1)\lambda_{d,u}^{(\cdot)} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & -2\lambda_{d,u}^{(\cdot)} & 2\lambda_{d,u}^{(\cdot)} & 0 \\ 0 & 0 & \cdots & 0 & -\lambda_{d,u}^{(\cdot)} & \lambda_{d,u}^{(\cdot)} \\ 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix},$$

where $\lambda_{d,u}^{(\cdot)}$ is the exponential rate that corresponds to the u-th phase of a PH distribution with parameters $\boldsymbol{\tau}_d^{(\cdot)}$ and $\mathbf{Z}_d^{(\cdot)}$. $\mathbf{C}_{d,u}^{(\cdot)}$ holds the transition probabilities after an interval of length Δ during epoch d:

$$\begin{aligned} \mathbf{C}_{d,u}^{(\cdot)} &= e^{\Delta \mathbf{Q}_{d,u}^{(\cdot)}} && (16) \\ &= \sum_{i=0}^{\infty} \frac{\Delta^i}{i!} \left(\mathbf{Q}_{d,u}^{(\cdot)}\right)^i && (17) \\ &= e^{-\Delta\lambda_{d,u,\max}^{(\cdot)}} \sum_{i=0}^{\infty} \frac{(\Delta\lambda_{d,u,\max}^{(\cdot)})^i}{i!} \left(\mathbf{P}_{d,u}^{(\cdot)}\right)^i, && (18) \end{aligned}$$

where $\lambda_{d,u,\max}^{(\cdot)} = y\lambda_{d,u}^{(\cdot)}$ and where $\mathbf{P}_{d,u}^{(\cdot)}$ is obtained as follows:

$$\mathbf{P}_{d,u}^{(\cdot)} = \frac{\mathbf{Q}_{d,u}^{(\cdot)}}{\lambda_{d,u,\max}^{(\cdot)}} + \mathbf{I}. \qquad (19)$$

The first row of $\mathbf{C}_{d,u}^{(\cdot)}$ holds the distribution of the number of successes (i.e., probabilities $\Pr(y|x,u,d)^{(\cdot)}$). The first block row of $\mathbf{C}_{d,u}^{(\cdot)}$ can be obtained by computing the first row of $\left(\mathbf{P}_{d,u}^{(\cdot)}\right)^i$ for all $i \geq 0$; this can be done by means of a simple recursion.

3.5 Model building blocks

Let $(a, \mathbf{s}, \mathbf{b})_t$ denote the state of the system at the start of interval t (of length Δ), where: (1) a is the phase of the arrival process, (2) $\mathbf{s}$ is a vector that holds the number of customers in each service phase and (3) $\mathbf{b}$ is a vector that holds the number of customers in each abandonment phase. $\mathbf{S}$ and $\mathbf{B}$ are the sets of all possible vectors $\mathbf{s}$ and $\mathbf{b}$ respectively. In addition, define $\pi(a, \mathbf{s}, \mathbf{b})_t$, the probability to visit state $(a, \mathbf{s}, \mathbf{b})_t$. The maximum dimension of the state space at the start of any period depends on (1) $Z_{\max}^{(\mathrm{I})}$ is the maximum number of phases of the arrival process, (2) $Z_{\max}^{(\mathrm{II})}$ is the maximum number of phases of the service process, (3) $s_{\max}$ is the maximum number of servers, (4) $Z_{\max}^{(\mathrm{III})}$ is the maximum number of phases of the abandonment process and (5) $Q_{\max}$ is the maximum number of customers allowed in queue.

In order to determine the state of the system at the start of a period t, we propose a stepwise procedure. The following steps are executed in sequence:

1. Initialization.
2. Implement process changes (arrival, service, abandonment and staffing process).
3. Arrival of customers.
4. Service of customers.
5. Abandonment of customers.

In what follows, we discuss each of these steps.

3.5.1 Initialization

When making a transition from a state $(a, \mathbf{s}, \mathbf{b})_t$ towards a state $(a, \mathbf{s}, \mathbf{b})_{t+1}$, several state space manipulations take place (e.g., process changes, arrival, service and abandonment of customers). In order to process these state space manipulations, we use a temporary probability vector $\pi(\delta, a, \mathbf{s}, \mathbf{b})$ (where δ is a binary variable). $\pi(\delta, a, \mathbf{s}, \mathbf{b})$ represents the state of the system after manipulation, whereas $\pi(1-\delta, a, \mathbf{s}, \mathbf{b})$ represents the state of the system before manipulation takes place. Our method requires the state of the system to be stored only before and after each manipulation, which enables to save

memory. This is of critical importance, as is is infeasible to store the state space over the entire time horizon (even for small instances).

During the initialization step, we initialize this temporary probability vector. An outline of the initialization step is provided in Algorithm 1.

Algorithm 1 Initialization step at start of period t

```
Initialize binary variable: δ = 0
for a = 1 to Z^(I)_{φ_t} do
  for all (s ∈ S) ∧ (b ∈ B) do
    Initialize temporary probability vector: π(δ, a, s, b) = 0
    Initialize temporary probability vector: π(1 − δ, a, s, b) = π(a, s, b)_t
  end for
end for
```

3.5.2 Implementation of process changes

There are four basic processes and therefore four events can take place when implementing the process changes. First, a new arrival epoch may start at the start of period t. In this case, the arrival phase is reset. Departing from state $(1-\delta, a, \mathbf{s}, \mathbf{b})$, the following transition takes place:

$$(1-\delta, a, \mathbf{s}, \mathbf{b}) \rightarrow (\delta, 1, \mathbf{s}, \mathbf{b}).$$

If a new service epoch starts at the start of period t, the service process of all customers in service is reset. Departing from state $(1-\delta, a, \mathbf{s}, \mathbf{b})$, the following transition takes place:

$$(1-\delta, a, \mathbf{s}, \mathbf{b}) \rightarrow (\delta, a, n_{\mathbf{s}}\mathbf{e}_1, \mathbf{b}),$$

where $n_{\mathbf{s}}$ is the sum of all entries in vector $\mathbf{s}$:

$$n_{\mathbf{s}} = \mathrm{tr}(\mathbf{sI}), \tag{20}$$

where tr is the matrix trace operator. If a new abandonment epoch starts at the start of a period t, the abandonment process of all waiting customers is reset. Departing from state $(1-\delta, a, \mathbf{s}, \mathbf{b})$, the following transition takes place:

$$(1-\delta, a, \mathbf{s}, \mathbf{b}) \rightarrow (\delta, a, \mathbf{s}, n_{\mathbf{b}}\mathbf{e}_1),$$

with $n_{\mathbf{b}}$ the sum of all entries in vector $\mathbf{b}$:

$$n_{\mathbf{b}} = \mathrm{tr}\left(\mathbf{bI}\right). \tag{21}$$

Algorithm 2 summarizes how changes in the arrival, service and abandonment process are implemented.

Algorithm 2 Implementation of arrival, service and abandonment process changes at start of period t

if the arrival process changes at the start of period t **then**
  for $a = 1$ to $Z_{\phi_t}^{(\mathrm{I})}$ **do**
    for all $(\mathbf{s} \in \mathbf{S}) \wedge (\mathbf{b} \in \mathbf{B})$ **do**
      Implement change: $\pi\left(\delta, 1, \mathbf{s}, \mathbf{b}\right) \mathrel{+}= \pi\left(1-\delta, a, \mathbf{s}, \mathbf{b}\right)$
      Initialize temporary probability vector: $\pi\left(1-\delta, a, \mathbf{s}, \mathbf{b}\right) = 0$
    end for
  end for
  Update binary variable: $\delta = 1 - \delta$
end if
if the service process changes at the start of period t **then**
  for $a = 1$ to $Z_{\phi_t}^{(\mathrm{I})}$ **do**
    for all $(\mathbf{s} \in \mathbf{S}) \wedge (\mathbf{b} \in \mathbf{B})$ **do**
      Implement change: $\pi\left(\delta, a, n_{\mathbf{s}}\mathbf{e}_1, \mathbf{b}\right) \mathrel{+}= \pi\left(1-\delta, a, \mathbf{s}, \mathbf{b}\right)$
      Initialize temporary probability vector: $\pi\left(1-\delta, a, \mathbf{s}, \mathbf{b}\right) = 0$
    end for
  end for
  Update binary variable: $\delta = 1 - \delta$
end if
if the abandonment process changes at the start of period t **then**
  for $a = 1$ to $Z_{\phi_t}^{(\mathrm{I})}$ **do**
    for all $(\mathbf{s} \in \mathbf{S}) \wedge (\mathbf{b} \in \mathbf{B})$ **do**
      Implement change: $\pi\left(\delta, a, \mathbf{s}, n_{\mathbf{b}}\mathbf{e}_1\right) \mathrel{+}= \pi\left(1-\delta, a, \mathbf{s}, \mathbf{b}\right)$
      Initialize temporary probability vector: $\pi\left(1-\delta, a, \mathbf{s}, \mathbf{b}\right) = 0$
    end for
  end for
  Update binary variable: $\delta = 1 - \delta$
end if

If the staffing process changes, two options arise: (1) new servers become available or (2) the number of servers decreases. If new servers become

available, waiting customers are selected according to a first-come first-serve (FCFS) policy. We first select customers in the last phase of the abandonment process because it is likely that they have waited the longest (note that this is not necessarily the case). For each server that becomes available, the following state space manipulation is performed:

$$(1-\delta, a, \mathbf{s}, \mathbf{b}) \rightarrow \begin{cases} (\delta, a, \mathbf{s}+\mathbf{e}_1, \mathbf{b}-\mathbf{e}_u) & \text{if } n_{\mathbf{b}} > 0, \\ (\delta, a, \mathbf{s}, \mathbf{b}) & \text{otherwise,} \end{cases}$$

where: (1) $\mathbf{e}_u$ is a single-entry vector populated with zeroes, except for the entry at position u, (2) $u : \max_u (b_u > 0)$ and (3) b_u is the u-th entry of vector $\mathbf{b}$. Algorithm 3 summarizes the activation of a single server. In case of a decrease in capacity, we need to account for the exhaustive service policy: some servers may complete a customer's service, even if they are scheduled to leave. We adopt an approach that is similar to the technique used by Ingolfsson [2005]: since servers that work overtime no longer influence the performance of future customers, these are removed from the system (along with the customers they serve). Although in reality, these customers are still in the system, this modification is necessary to correctly calculate other performance measures (such as the distribution of the virtual waiting time, see Section 3.6). A decrease of x servers is accommodated by first removing all idle servers. If insufficient idle servers are available, $c_{(x,\mathbf{s},t)}$ active servers are removed:

$$c_{(x,\mathbf{s},t)} = \max\left(0, x - s_t + n_{\mathbf{s}}\right), \tag{22}$$

where $s_t - n_{\mathbf{s}}$ represents the number of idle servers. Given a distribution of customers $\mathbf{s}$ over the different phases of the service process, the probability to remove a server that is processing a customer who is in phase u of his service process equals:

$$\Pr\left(u|\mathbf{s}\right) = \frac{s_u}{n_{\mathbf{s}}}, \tag{23}$$

where s_u is the u-th entry of vector $\mathbf{s}$. For each active server that is removed, the following state space manipulation is performed (the transition probability is indicated above the arrow):

$$(1-\delta, a, \mathbf{s}, \mathbf{b}) \xrightarrow{\Pr(u|\mathbf{s})} (\delta, a, \mathbf{s}-e_u, \mathbf{b}),$$

Algorithm 4 summarizes how changes in the staffing process are implemented.

Algorithm 3 Activation of a single server

for $a = 1$ to $Z_{\phi_t}^{(\mathrm{I})}$ **do**
  for all $(\mathbf{s} \in \mathbf{S}) \wedge (\mathbf{b} \in \mathbf{B})$ **do**
    for $u = Z_{\phi_t}^{(\mathrm{III})}$ to 1 **do**
      if $b_u > 0$ **then**
        Implement change: $\pi(\delta, a, \mathbf{s} + \mathbf{e}_1, \mathbf{b} - \mathbf{e}_u) \mathrel{+}= \pi(1 - \delta, a, \mathbf{s}, \mathbf{b})$
        Customer has entered service, exit loop: $u = 1$
      end if
      Initialize temporary probability vector: $\pi(1 - \delta, a, \mathbf{s}, \mathbf{b}) = 0$
    end for
  end for
end for
Update binary variable: $\delta = 1 - \delta$

Algorithm 4 Implement staffing process change at start of period t

if x servers become available at the start of period t **then**
  for $i = 1$ to x **do**
    Activate a single server: Algorithm 3
  end for
else if x servers are removed at the start of period t **then**
  while $x > 0$ **do**
    for $a = 1$ to $Z_{\phi_t}^{(\mathrm{I})}$ **do**
      for all $(\mathbf{s} \in \mathbf{S}) \wedge (\mathbf{b} \in \mathbf{B})$ **do**
        if $c_{(x,\mathbf{s},t)} > 0$ **then**
          for $u = 1$ to $Z_{\phi_t}^{(\mathrm{II})}$ **do**
            Implement change: $\pi(\delta, a, \mathbf{s} - \mathbf{e}_u, \mathbf{b}) \mathrel{+}= \pi(1 - \delta, a, \mathbf{s}, \mathbf{b}) \Pr(u|\mathbf{s})$
          end for
          Initialize temporary probability vector: $\pi(1 - \delta, a, \mathbf{s}, \mathbf{b}) = 0$
        end if
      end for
    end for
    Update binary variable: $\delta = 1 - \delta$
    Decrement x: $x = x - 1$
  end while
end if

3.5.3 Arrival, service and abandonment of customers

From the counting process discussed in Section 3.3, we obtain probabilities $\Pr(x, v|u, d)$. Using these probabilities, we can determine the state of the system after arrivals have taken place. Because the size of the queue is limited to $Q_{\max}$ customers, we impose a reflecting boundary (i.e., whenever x customers arrive, with $x \geq Q_{max} - n_{\mathbf{b}}$, the resulting queue length equals $Q_{\max}$). More formally:

$$(1-\delta, u, \mathbf{s}, \mathbf{b}) \xrightarrow{\Pr\left(x,v|u,\phi_t^{(\mathrm{I})}\right)} \begin{cases} (\delta, v, \mathbf{s}, \mathbf{b} + x\mathbf{e}_1) & \text{if } Q_{\max} \geq n_{\mathbf{b}} + x, \\ (\delta, v, \mathbf{s}, \mathbf{b} + (Q_{\max} - n_{\mathbf{b}})\,\mathbf{e}_1) & \text{otherwise.} \end{cases}$$

Algorithm 5 provides an outline of the arrival step.

Algorithm 5 Arrival of customers during interval t

```
for a = 1 to Z_{φ_t}^{(I)} do
  for all (s ∈ S) ∧ (b ∈ B) do
    for all x = 0 to Q_max do
      if Q_max ≥ n_b + x then
        Arrival of x customers:
        π(δ, v, s, b + x e_1) += π(1 − δ, a, s, b) Pr(x, v|a, φ_t^{(I)})
      else
        Arrival of Q_max − n_b customers:
        π(δ, v, s, b + (Q_max − n_b) e_1) += π(1 − δ, a, s, b) Pr(x, v|a, φ_t^{(I)})
      end if
    end for
    Initialize temporary probability vector: π(1 − δ, a, s, b) = 0
  end for
end for
Update binary variable: δ = 1 − δ
```

Customers in service are only allowed to advance a single phase during an interval of length Δ. The probability of advancing a phase is obtained from the procedure discussed in Section 3.4. For each phase, a state space manipulation is performed and phases are processed in reverse order. Customers who are in the last phase of their service process complete service (note that

$Z_{\phi_t}^{(\mathrm{II})}$ is the last phase of the service process):

$$(1-\delta, a, \mathbf{s}, \mathbf{b}) \xrightarrow{\Pr(x|n_{\mathbf{s}},u,\phi_t)^{(\mathrm{II})}} \begin{cases} (\delta, a, \mathbf{s} - x\mathbf{e}_u, \mathbf{b}) & \text{if } s_u > 0 \wedge u = Z_{\phi_t}^{(\mathrm{II})}, \\ (\delta, a, \mathbf{s}, \mathbf{b}) & \text{otherwise.} \end{cases}$$

If the service process is not modeled using a two-phase Coxian distribution, customers who are not in the last phase of their service process advance a phase:

$$(1-\delta, a, \mathbf{s}, \mathbf{b}) \xrightarrow{\Pr(x|n_{\mathbf{s}},u,\phi_t)^{(\mathrm{II})}} \begin{cases} (\delta, a, \mathbf{s} - x\mathbf{e}_u + x\mathbf{e}_{u+1}, \mathbf{b}) & \text{if } s_u > 0 \wedge 1 \leq u < Z_{\phi_t}^{(\mathrm{II})}, \\ (\delta, a, \mathbf{s}, \mathbf{b}) & \text{otherwise.} \end{cases}$$

If the service process is modeled using a two-phase Coxian distribution, there is a probability that customers in the first phase complete service instead of advancing a phase. The probability of completing service equals $1 - \beta_{\phi_t}^{(\mathrm{II})}$. The probability that y out of x customers complete service is binomially distributed and equals:

$$\Pr(y|x, \phi_t)^{(\mathrm{II})} = \frac{x!}{y!\,(x-y)!} \left(1 - \beta_{\phi_t}^{(\mathrm{II})}\right)^{y} \left(\beta_{\phi_t}^{(\mathrm{II})}\right)^{x-y}. \tag{24}$$

The state space transitions are summarized as follows:

$$(1-\delta, a, \mathbf{s}, \mathbf{b}) \xrightarrow{\Pr(x|n_{\mathbf{s}},u,\phi_t)^{(\mathrm{II})}\Pr(y|x,\phi_t)^{(\mathrm{II})}} (\delta, a, \mathbf{s} - x\mathbf{e}_u + (x-y)\,\mathbf{e}_{u+1}, \mathbf{b}).$$

Algorithm 6 provides an outline of the service step.

With respect to the abandonment process, we adopt a logic that is similar to the one of the service process. Algorithm 7 provides an outline of the abandonment step.

After the abandonment step, probabilities $\pi(a, \mathbf{s}, \mathbf{b})_{t+1}$ are readily available:

$$\pi(a, \mathbf{s}, \mathbf{b})_{t+1} = \pi(1-\delta, a, \mathbf{s}, \mathbf{b}). \tag{25}$$

3.6 Performance measures

Let $\mathbf{W} \subseteq \mathbf{T}$ denote the set of performance intervals and define $\varphi_w^{(\cdot)} = i$, the function that maps a performance interval w onto an epoch i, where i is the ongoing epoch of process $(\cdot)$ at the start of performance interval w. The performance measures of interest are: (1) the expected queue length,

Algorithm 6 Service of customers during interval t

for $u = Z_{\phi_t}^{(\mathrm{II})}$ to 1 **do**
  for $a = 1$ to $Z_{\phi_t}^{(\mathrm{I})}$ **do**
    for all $(\mathbf{s} \in \mathbf{S}) \wedge (\mathbf{b} \in \mathbf{B})$ **do**
      for all $x = 0$ to s_u **do**
        if $u = Z_{\phi_t}^{(\mathrm{II})}$ **then**
          x customers complete service:
          $\pi\left(\delta, a, \mathbf{s} - x\mathbf{e}_u, \mathbf{b}\right) \mathrel{+}= \pi\left(1-\delta, a, \mathbf{s}, \mathbf{b}\right) \Pr\left(x|s_u, u, \phi_t\right)^{(\mathrm{II})}$
        else
          if Two-phase Coxian distribution is used **then**
            for all $y = 0$ to x **do**
              y customers complete service, $x - y$ customers advance:
              $\pi\left(\delta, a, \mathbf{s} - x\mathbf{e}_u + (x-y)\,\mathbf{e}_{u+1}, \mathbf{b}\right) \mathrel{+}=$
              $\pi\left(1-\delta, a, \mathbf{s}, \mathbf{b}\right) \Pr\left(x|s_u, u, \phi_t\right)^{(\mathrm{II})} \Pr\left(y|x, \phi_t\right)^{(\mathrm{II})}$
            end for
          else
            x customers advance a phase:
            $\pi\left(\delta, a, \mathbf{s} - x\mathbf{e}_u + x\mathbf{e}_{u+1}, \mathbf{b}\right) \mathrel{+}= \pi\left(1-\delta, a, \mathbf{s}, \mathbf{b}\right) \Pr\left(x|s_u, u, \phi_t\right)^{(\mathrm{II})}$
          end if
        end if
      end for
      Initialize temporary probability vector: $\pi\left(1-\delta, a, \mathbf{s}, \mathbf{a}\right) = 0$
    end for
  end for
  Update binary variable: $\delta = 1 - \delta$
end for
while Servers are idle **do**
  Activate a single server: algorithm 3
end while

Algorithm 7 Abandonment of customers during interval t

for $u = Z_{\phi_t}^{(\mathrm{III})}$ to 1 **do**
  for $a = 1$ to $Z_{\phi_t}^{(\mathrm{I})}$ **do**
    for all $(\mathbf{s} \in \mathbf{S}) \wedge (\mathbf{b} \in \mathbf{B})$ **do**
      for all $x = 0$ to b_u **do**
        if $u = Z_{\phi_t}^{(\mathrm{III})}$ **then**
          x customers abandon:
          $\pi\left(\delta, a, \mathbf{s}, \mathbf{b} - x\mathbf{e}_u\right) \mathrel{+}= \pi\left(1-\delta, a, \mathbf{s}, \mathbf{b}\right) \Pr\left(x|b_u, u, \phi_t\right)^{(\mathrm{III})}$
        else
          if Two-phase Coxian distribution is used **then**
            for all $y = 0$ to x **do**
              y customers abandon, $x - y$ customers advance:
              $\pi\left(\delta, a, \mathbf{s}, \mathbf{b} - x\mathbf{e}_u + (x-y)\,\mathbf{e}_{u+1}\right) \mathrel{+}=$
              $\pi\left(1-\delta, a, \mathbf{s}, \mathbf{b}\right) \Pr\left(x|b_u, u, \phi_t\right)^{(\mathrm{III})} \Pr\left(y|x, \phi_t\right)^{(\mathrm{III})}$
            end for
          else
            x customers advance a phase:
            $\pi\left(\delta, a, \mathbf{s}, \mathbf{b} - x\mathbf{e}_u + x\mathbf{e}_{u+1}\right) \mathrel{+}= \pi\left(1-\delta, a, \mathbf{s}, \mathbf{b}\right) \Pr\left(x|b_u, u, \phi_t\right)^{(\mathrm{III})}$
          end if
        end if
      end for
      Initialize temporary probability vector: $\pi\left(1-\delta, a, \mathbf{s}, \mathbf{a}\right) = 0$
    end for
  end for
  Update binary variable: $\delta = 1 - \delta$
end for

(2) the expected queue length at the start of performance interval w, (3) the variance of the expected queue length, (4) the variance of the expected queue length at the start of performance interval w, (5) the expected number of abandonments during performance interval w and (6) the waiting time distribution of a virtual customer that arrives at the start of performance interval w. The virtual waiting time at time t is defined as the time a virtual customer would have to spend in queue if he were to arrive at time t (cf. Gross et al. [2008] and Campello and Ingolfsson [2011]). The expected queue length is approximated by:

$$\mathcal{Q} = \sum_{t=1}^{T} \sum_{a=1}^{Z_{\phi_t}^{(\mathrm{I})}} \sum_{\mathbf{s}\in\mathbf{S}} \sum_{\mathbf{b}\in\mathbf{B}} \pi\left(a,\mathbf{s},\mathbf{b}\right)_t n_{\mathbf{b}}. \tag{26}$$

The expected queue length at the start of performance interval w equals:

$$\mathcal{Q}_w = \sum_{a=1}^{Z_{\varphi_w}^{(\mathrm{I})}} \sum_{\mathbf{s}\in\mathbf{S}} \sum_{\mathbf{b}\in\mathbf{B}} \pi\left(a,\mathbf{s},\mathbf{b}\right)_w n_{\mathbf{b}}. \tag{27}$$

The variance of the queue length is approximated by:

$$\mathcal{V} = \sum_{t=1}^{T} \sum_{a=1}^{Z_{\phi_t}^{(\mathrm{I})}} \sum_{\mathbf{s}\in\mathbf{S}} \sum_{\mathbf{b}\in\mathbf{B}} \pi\left(a,\mathbf{s},\mathbf{b}\right)_t \left(n_{\mathbf{b}} - \mathcal{Q}_t\right)^2. \tag{28}$$

The variance of the expected queue length at performance interval w equals:

$$\mathcal{V}_w = \sum_{a=1}^{Z_{\varphi_w}^{(\mathrm{I})}} \sum_{\mathbf{s}\in\mathbf{S}} \sum_{\mathbf{b}\in\mathbf{B}} \pi\left(a,\mathbf{s},\mathbf{b}\right)_w \left(n_{\mathbf{b}} - \mathcal{Q}_w\right)^2. \tag{29}$$

Let $\mathcal{A}_w$ denote the expected number of abandonments during performance interval w. $\mathcal{A}_w$ is computed during the abandonment step; see Algorithm 8 for details (which is an adaptation of Algorithm 7).

Define $\Pr\left(\mathcal{W}_w = h\right)$, the probability that a virtual customer who arrives at the start of performance interval w receives service during interval $w + h$ (i.e., the virtual customer receives service after waiting h intervals of length Δ). In order to obtain $\Pr\left(\mathcal{W}_w = h\right)$, we use a death process and stop the arrival process at the start of performance interval w. The first period during

Algorithm 8 Expected number of abandonments during performance interval w

for $u = Z_{\varphi_w}^{(\mathrm{III})}$ to 1 **do**
  for $a = 1$ to $Z_{\varphi_w}^{(\mathrm{I})}$ **do**
    for all $(\mathbf{s} \in \mathbf{S}) \wedge (\mathbf{b} \in \mathbf{B})$ **do**
      for all $x = 0$ to b_u **do**
        if $u = Z_{\varphi_w}^{(\mathrm{III})}$ **then**
          x customers abandon:
          $\pi(\delta, a, \mathbf{s}, \mathbf{b} - x\mathbf{e}_u) \mathrel{+}= \pi(1-\delta, a, \mathbf{s}, \mathbf{b}) \Pr(x|b_u, u, \varphi_w)^{(\mathrm{III})}$
          $\mathcal{A}_w \mathrel{+}= x\pi(1-\delta, a, \mathbf{s}, \mathbf{b}) \Pr(x|n_{\mathbf{b}}, u, \varphi_w)^{(\mathrm{III})}$
        else
          if Two-phase Coxian distribution is used **then**
            for all $y = 0$ to x **do**
              y customers abandon, $x - y$ customers advance:
              $\pi(\delta, a, \mathbf{s}, \mathbf{b} - x\mathbf{e}_u + (x-y)\mathbf{e}_{u+1}) \mathrel{+}=$
              $\pi(1-\delta, a, \mathbf{s}, \mathbf{b}) \Pr(x|b_u, u, \varphi_w)^{(\mathrm{III})} \Pr(y|x, \varphi_w)^{(\mathrm{III})}$
              $\mathcal{A}_w \mathrel{+}= y\pi(1-\delta, a, \mathbf{s}, \mathbf{b}) \Pr(x|n_{\mathbf{b}}, u, \varphi_w)^{(\mathrm{III})} \Pr(y|x, \varphi_w)^{(\mathrm{III})}$
            end for
          else
            x customers advance a phase:
            $\pi(\delta, a, \mathbf{s}, \mathbf{b} - x\mathbf{e}_u + x\mathbf{e}_{u+1}) \mathrel{+}= \pi(1-\delta, a, \mathbf{s}, \mathbf{b}) \Pr(x|b_u, u, \varphi_w)^{(\mathrm{III})}$
          end if
        end if
      end for
      Initialize temporary probability vector: $\pi(1-\delta, a, \mathbf{s}, \mathbf{a}) = 0$
    end for
  end for
  Update binary variable: $\delta = 1 - \delta$
end for

which a server becomes idle, defines the waiting time of the virtual customer. More formally, the virtual waiting time equals $h\Delta$ where h is the first integer for which $\mathcal{N}_{w+h} < s_{w+h}$ and where $\mathcal{N}_t$ denotes the number of customers in system at time t, if the arrival process is stopped at the start of performance interval w. Note that $\mathcal{N}_t$ does not include customers serviced by servers working overtime. Algorithm 9 is an adaptation of Algorithm 3 that allows us to determine the interval during which a server becomes idle. The death process is outlined in Algorithm 11 (see next section).

3.7 Model summary

Our model enables both the transient and the (periodic) steady-state analysis of the $G(t)/G(t)/s(t) + G(t)$ queue. Steady-state, however, will usually not be achieved at the end of the time horizon, hence the model has to run for multiple consecutive "cycles" (each with a length equal to the time horizon T). Let $c_{\max}$ denote the number of cycles after which steady-state results are obtained. In addition, define ε_c, the relative difference in queue lengths for cycles $(c-1)$ and c:

$$\varepsilon_c = \sum_{t=1}^{T} \left| 1 - \frac{\mathcal{Q}_{t,c}}{\mathcal{Q}_{t,c-1}} \right|. \tag{30}$$

If ε_c is smaller than the (user-specified) parameter $\varepsilon_{\max}$, cycle c is the last cycle and steady-state results have been obtained. In other words, $c_{\max}$ is the smallest integer for which $\varepsilon_{c_{\max}} < \varepsilon_{\max}$, where $\varepsilon_{\max}$ is the predefined maximum allowed deviation. In the case of a transient analysis, the user can specify the number of cycles that needs to be processed.

In summary, Algorithm 10 models the system over T periods and $c_{\max}$ cycles. Algorithms 1–7 and Equation 25 allow to compute the vector of state space probabilities at the start of period $t+1$ when departing from the vector of state space probabilities at the start of period t. Performance measures are obtained using Equations 28–29 and Algorithms 8, 9 and 11, where Algorithm 11 models the death process that is required to calculate the waiting time distribution of a virtual customer that arrives at the start of performance interval w. Algorithm 11 is similar to Algorithm 10, however it does not allow arrivals to take place.

Algorithm 9 Waiting time distribution of a virtual customer that arrives at the start of performance interval w

for $a = 1$ to $Z_{\varphi_w}^{(\mathrm{I})}$ **do**
  for all $(\mathbf{s} \in \mathbf{S}) \wedge (\mathbf{b} \in \mathbf{B})$ **do**
    for $u = Z_{\varphi_w}^{(\mathrm{III})}$ to 1 **do**
      if $b_u > 0$ **then**
        Implement change: $\pi(\delta, a, \mathbf{s} + \mathbf{e}_1, \mathbf{b} - \mathbf{e}_u) \mathrel{+}= \pi(1 - \delta, a, \mathbf{s}, \mathbf{b})$
        Customer has entered service, exit loop: $u = 1$
      else
        Update virtual waiting time distribution:
        $\Pr(\mathcal{W}_w = w_t) \mathrel{+}= \pi(1 - \delta, a, \mathbf{s}, \mathbf{b})$
      end if
      Initialize temporary probability vector: $\pi(1 - \delta, a, \mathbf{s}, \mathbf{b}) = 0$
    end for
  end for
end for
Update binary variable: $\delta = 1 - \delta$

4 Results

We use a simulation study to assess the validity and accuracy of the model over a set of 162 problem instances. Both the Markov model and the simulation model are implemented in Visual Studio C++. All tests are performed on a Intel I7 3.40 GHz computer, with 8 GB RAM.

In what follows, we first describe the computational experiment (Section 4.1) and discuss the main drivers of model accuracy and computation speeds (Section 4.2). Next, we validate the model and elaborate further on the trade-off between accuracy and computation times (Section 4.3).

4.1 Experimental setting

Table 1 provides an overview of the parameter settings that are used to construct the test set. The parameters give rise to 162 problem instances that are representative of small- to medium-sized systems. Each instance covers a one-day time horizon (i.e., 1440 minutes) which is divided into smaller periods of length Δ. In the experiment, Δ ranges from 0.0625 to 1 minute. The arrival rate is piecewise constant over 10-minute intervals and the staffing

Algorithm 10 Model summary

for $a = 1$ to $Z_{\phi_1}^{(\mathrm{I})}$ **do**
  for all $(\mathbf{s} \in \mathbf{S}) \wedge (\mathbf{b} \in \mathbf{B})$ **do**
    Initialize vector: $\pi\left(a, \mathbf{s}, \mathbf{b}\right)_1 = 0$
  end for
end for
Initialize vector: $\pi\left(1, \mathbf{0}, \mathbf{0}\right)_1 = 1$
Initialize cycle: $c = 1$
while $c < c_{\max}$ **do**
  Determine whether $c = c_{\max}$ using Equation 30
  Initialize period: $t = 1$
  while $t < T$ **do**
    Perform initialization: Algorithm 1
    Implement process changes: Algorithms 2, 3 and 4
    Arrival of customers: Algorithm 5
    Service of customers: Algorithms 3 and 6
    if $c = c_{\max}$ and t is the start of performance interval w **then**
      Abandonment of customers: Algorithm 8
      Compute $\Pr\left(\mathcal{W}_w = w_t\right)$: Algorithm 11
    else
      Abandonment of customers: Algorithm 7
    end if
    Compute $\pi\left(a, \mathbf{s}, \mathbf{b}\right)_{t+1}$ using Equation 25
    Increment period: $t = t + 1$
  end while
  Increment cycle: $c = c + 1$
end while

Algorithm 11 Computation of the virtual waiting time distribution at performance interval w

Initialize period: $t = w$
while $t < T$ **do**
  Implement process changes: Algorithms 2, 4 and 9
  Service of customers: Algorithms 6 and 9
  Abandonment of customers: Algorithm 7
  Increment period: $t = t + 1$
  Increment virtual waiting time: $h = h + 1$
  if $h \geq W_{\max}$ **then**
    Maximum waiting time reached, exit loop
  end if
end while
while $c < \infty$ **do**
  Initialize period: $t = 1$
  while $t < T$ **do**
    Implement process changes: Algorithms 2, 4 and 9
    Service of customers: Algorithms 6 and 9
    Abandonment of customers: Algorithm 7
    Increment period: $t = t + 1$
    Increment virtual waiting time: $h = h + 1$
    if $h \geq W_{\max}$ **then**
      Maximum waiting time reached, exit loop
    end if
  end while
  Increment cycle: $c = c + 1$
end while

interval has a length of 30 minutes.

The time-varying arrival rate $\lambda_t^{(\mathrm{I})}$ is modeled as a discretized sine function with cycle equal to T. Let $\mathrm{RA}^{(\mathrm{I})} \equiv A/\bar{\lambda}^{(\mathrm{I})}$ denote the relative amplitude, with A the absolute amplitude and $\bar{\lambda}^{(\mathrm{I})}$ the average arrival rate over the time horizon. More formally:

$$\lambda_t^{(\mathrm{I})} = \frac{\bar{\lambda}^{(\mathrm{I})}}{2}\left(2 + \mathrm{RA}^{(\mathrm{I})} \sin\left(\frac{2\pi t}{T}\right) + \mathrm{RA}^{(\mathrm{I})} \sin\left(\frac{2\pi\,(t+1)}{T}\right)\right). \tag{31}$$

Note that $\bar{\lambda}^{(\mathrm{I})}$ is determined uniquely by the average capacity $\bar{c}$, the average service rate $\bar{\lambda}^{(\mathrm{II})}$ and the average traffic intensity $\bar{\rho} \equiv \bar{\lambda}^{(\mathrm{I})}/\left(\bar{c}\bar{\lambda}^{(\mathrm{II})}\right)$. Given the parameter settings in Table 1, it follows that $\bar{\lambda}^{(\mathrm{I})}$ ranges between 1 and 57 customers per hour. To limit the size of the test set, we assume that all processes have the same C^2 (i.e., 0.5, 1 or 2) and that the distributional parameters of the service and the abandonment processes remain constant throughout the day. We emphasize that these assumptions are not a limitation of the suggested model (that can handle different C^2 values for the arrival, service and abandonment processes, as well as time-dependence in the process parameters).

The staffing process is modeled as a discretized sine function with relative amplitude $\mathrm{RA}^{(\mathrm{IV})}$. As such:

$$c_t = \frac{\bar{c}}{2}\left(2 + \mathrm{RA}^{(\mathrm{IV})} \sin\left(\frac{2\pi t}{T}\right) + \mathrm{RA}^{(\mathrm{IV})} \sin\left(\frac{2\pi\,(t+1)}{T}\right)\right). \tag{32}$$

Note that the capacity function is not shifted compared to the arrival rate function (which could be done to account for the commonly observed congestion lag).

Parameter	Values
Time horizon T (in min)	1440
Period length Δ (in min)	$\{0.0625, 0.125, 0.25, 0.5, 1\}$
Epoch length (arrival process, in min)	10
Epoch length (staffing process, in min)	30
Performance interval length (in min)	Δ
Relative amplitude $\mathrm{RA}^{(\mathrm{I})}$	0.5
Average service rate $\bar{\lambda}^{(\mathrm{II})}$ (customers/hour)	$\{1, 2, 6\}$
Average abandonment rate $\bar{\lambda}^{(\mathrm{III})}$	$\{0.5\bar{\lambda}^{(\mathrm{II})}, \bar{\lambda}^{(\mathrm{II})}\}$
Average capacity $\bar{c}$	$\{2, 5, 10\}$
Relative amplitude $\mathrm{RA}^{(\mathrm{IV})}$	0.5
Average traffic intensity $\bar{\rho} \equiv \bar{\lambda}^{(\mathrm{I})} / \left(\bar{c}\bar{\lambda}^{(\mathrm{II})}\right)$	$\{0.5, 0.75, 0.95\}$
Squared coefficient of variation C^2	$\{0.5, 1, 2\}$
Maximum waiting time $W_{\max}$ (in min)	240
Maximum allowed deviation $\varepsilon_{\max}$	0.0001

Table 1: Parameter settings used in the computational experiment

In order to validate the model, we use the expected queue length. Let $\mathcal{Q}_t^{\mathrm{SIM}}$ denote the queue length at the start of interval t. $\mathcal{Q}_t^{\mathrm{SIM}}$ is obtained by means of an accurate simulation model (this can be considered as the "true" value). The relative error (RE) at the start of period t can be expressed as:

$$\mathrm{RE}_t = \frac{\left|\mathcal{Q}_t^{\mathrm{SIM}} - \mathcal{Q}_t\right|}{\mathcal{Q}_t^{\mathrm{SIM}}}. \tag{33}$$

To obtain an aggregate performance metric over the time horizon, RE_t is weighted with the queue length. As such, the weighted relative error (WRE) for a given problem instance is defined as follows:

$$\mathrm{WRE} = \sum_{t=1}^{T} \left(\frac{\mathcal{Q}_t^{\mathrm{SIM}}}{\sum_{t=1}^{T} \mathcal{Q}_t^{\mathrm{SIM}}} \mathrm{RE}_t \right), \tag{34}$$

$$= \frac{\sum_{t=1}^{T} \left|\mathcal{Q}_t^{\mathrm{SIM}} - \mathcal{Q}_t\right|}{\sum_{t=1}^{T} \mathcal{Q}_t^{\mathrm{SIM}}}. \tag{35}$$

4.2 Drivers of accuracy and computation speed

We distinguish three main drivers of accuracy and computation speed:

1. The length of Δ.

2. The size of the state space.

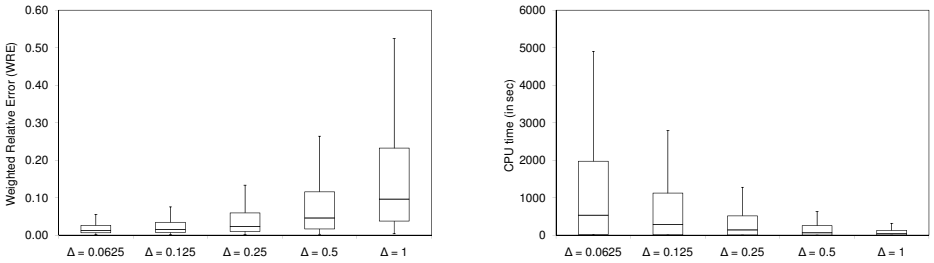
3. The approximations used in the model.

The choice of Δ determines the frequency at which the system is observed. Evidently, larger values of Δ lead to shorter computation times. Accurate results, however, can only be obtained if Δ is sufficiently small. During an interval of length Δ, events aggregate. The more events aggregate (i.e., the larger the event frequency), the less accurate the results. Therefore, Δ should be chosen such that the number of aggregated events remains small.

The size of the state space only impacts the computation time. The state space grows exponentially with the required number of phases in the arrival, service and abandonment processes and grows linearly with the maximum capacity and the maximum queue length; the latter can be controlled by the user.

The presented model is an approximation because of three reasons. Firstly, the general arrival, service and abandonment processes are approximated by means of PH distributions. Secondly, as discussed in Section 3.5.2, we assume that customers in the last phase of the abandonment process have waited the longest. This assumption significantly reduces the required computational effort. Thirdly, we assume that any phase in the arrival, service and abandonment process takes at least one interval to complete. Consequently, distributions other than the exponential distribution require lower values of Δ to maintain accuracy. Clearly, the error that is induced by this last assumption tends to zero as Δ approaches zero.

We would like to point out that computation speed also depends on the number of performance intervals that was specified. Because the performance measures are calculated at each performance interval, an increase in the number of performance intervals will also increase the required computation time. This especially is true for the calculation of the virtual waiting time distribution as it involves the evaluation of a computationally intensive death process. Note that the computation times reported in this study include the computation of all aforementioned performance measures.



(a) Weighted relative error (b) CPU time(in sec)

Figure 5: Weighted relative error and CPU times of test set

4.3 Model validation and results

Figure 5(a) presents a box-and-whisker diagram of the WRE for different values of Δ. It is clear that the proposed method yields highly accurate results, provided that Δ is sufficiently small. Figure 5(b) shows the required CPU times in terms of Δ. We observe a clear trade-off between accuracy and computational effort. In the remainder of this section, we further analyze this trade-off.

The lower quantiles of Figure 5(a) show that even for high values of Δ, the model can yield accurate results. As expected, for any value of Δ, the model is most accurate if $C^2 = 1$. This is illustrated in Table 2 and Figure 6. If C^2 does not equal unity, the PH distributions adopt exponential distributions with a mean that is smaller than the mean of the approximated distribution. In other words, the event frequency increases. For these settings, a lower value for Δ may be required to achieve sufficient accuracy. The performance is worst for the instances with $C^2 = 0.5$. These are modeled using a hypo-exponential distribution (see Section 4). For $C^2 = 0.5$, a series of two identical exponential distributions is used, with a mean that is half the mean of the approximated distribution. As such, the event frequency is doubled. The accuracy tends to be better for $C^2 > 1$, thanks to the use of the two-phase Coxian distribution. This distribution increases the event frequency, but to a lesser extent than the hypo-exponential distribution.

Likewise, Table 2 shows that the CPU times increase drastically for non-exponential settings. This is no surprise, as the state space grows exponentially with the number of phases.

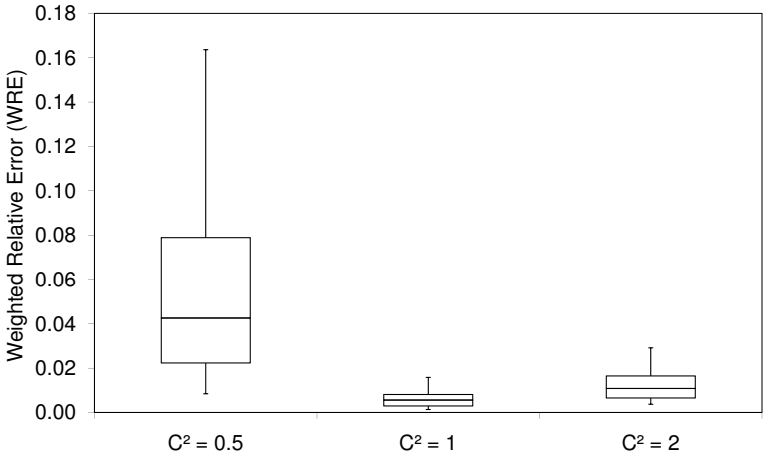


Figure 6: Weighted relative error, as a function of C^2 (for $\Delta = 0.0625$)

Δ		Weighted relative error (WRE) $C^2 = 0.5$	$C^2 = 1$	$C^2 = 2$	CPU time (in sec) $C^2 = 0.5$	$C^2 = 1$	$C^2 = 2$
	Min	0.008	0.001	0.004	332	3	386
0.0625	Avg	0.056	0.007	0.012	3048	9	3525
	Max	0.233	0.020	0.030	9504	19	11618
	Min	0.010	0.002	0.003	230	2	209
0.125	Avg	0.069	0.012	0.012	1725	5	2029
	Max	0.275	0.037	0.032	4930	10	5855
	Min	0.013	0.004	0.002	112	1	116
0.25	Avg	0.095	0.023	0.018	845	2	1009
	Max	0.352	0.071	0.072	2184	4	2667
	Min	0.016	0.008	0.002	44	0	52
0.5	Avg	0.144	0.044	0.035	366	1	433
	Max	0.480	0.136	0.146	1016	3	1212
	Min	0.028	0.016	0.003	28	0	26
1	Avg	0.228	0.084	0.075	208	1	243
	Max	0.656	0.252	0.277	501	1	624

Table 2: WRE and CPU time (in sec), as a function of C^2 (for all considered Δ values)

Figure 7 plots the trade-off between accuracy and computation time, for different values of the average utilization (Figure 7(a)), the average service rate (Figure 7(b)), the average capacity (Figure 7(c)) and the average aban-

donment rate (Figure 7(d)). In each plot, every observation point represents the combination of WRE and CPU time for a given value of Δ, averaged over all instances characterized by a given parameter setting.

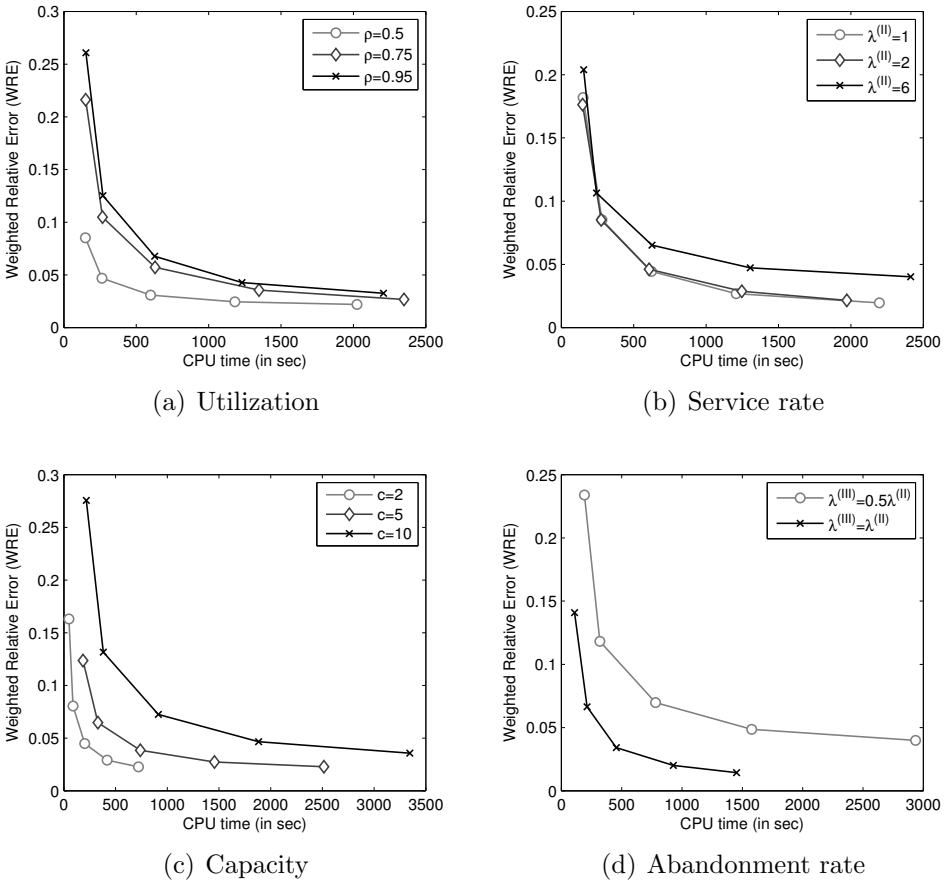
Figure 7(a) shows that smaller levels of utilization require less computational effort in order to maintain the same level of accuracy. The same holds for service rates and capacity, as is clear from Figures 7(b) and 7(c). For all three cases, a decrease in utilization/service rate/capacity results in a decrease of event frequency. In addition, a decrease in capacity also results in a decrease of the size of the state space. Therefore, a large value of Δ may suffice to obtain reasonable accuracy in systems that have low utilization/service rates/capacity.

From Figure 7(d), it is clear that smaller abandonment rates require more computational effort in order to maintain the same level of accuracy. This is somewhat surprising as small abandonment rates decrease the event frequency. They, however, also increase the utilization. Therefore, smaller values of Δ may be required in systems that have low abandonment rates.

We can conclude that the trade-off between accuracy and computation time is mainly influenced by (1) the event frequency, (2) the C^2 values of the arrival, service and abandonment processes and (3) the size of the state space. As a result, the model is most appropriate in settings with low service rate, low utilization, low capacity or high abandonment rates.

5 Conclusions and directions for further research

In this article, we have presented a model that approximates the transient and steady-state behavior of a $G(t)/G(t)/s(t) + G(t)$ queueing system with an exhaustive service policy. The model yields the following (time-varying) performance measures: (1) the expected queue length, (2) the variance of the queue length, (3) the expected number of abandonments and (4) the virtual waiting time distribution of a customer when arriving at an arbitrary moment in time. The analysis does not require heavy traffic conditions (a condition that is common in existing work). Computational experiments showed that results are highly accurate and that computational effort remains limited, especially in small- to medium-sized systems. Problem instances with (1) a low service rate, (2) a low average capacity, (3) a low utilization or (4) a



(a) Utilization (b) Service rate

(c) Capacity (d) Abandonment rate

Figure 7: Trade-off between accuracy and computation time

high abandonment rate typically required less computation time to achieve a given level of accuracy. Other problems can be solved as well, albeit at a higher computational cost.

Existing models are often incapable of accurately capturing the (time-varying) behavior of small- to medium-scaled systems. Our model on the other hand, excels in this. Banks, retail stores and emergency departments are just a few of the example systems that may benefit from our model. Our approach could, for instance, be used to evaluate the performance of alternative personnel schedules or to determine the minimal required staffing levels. We intend to further explore our method's applicability within the context of capacity planning in future research.

Acknowledgements: This research was supported by the Research Foundation-Flanders (FWO) (grant no G.0547.09).

References

L.V. Green, J. Soares, J.F. Giglio, R.A. Green, Using Queueing Theory to Increase the Effectiveness of Emergency Department Provider Staffing, Academic Emergency Medicine 13(1) (2006) 61–68.

L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, L. Zhao, Statistical analysis of a telephone call center: A queueing perspective, Journal of the American Statistical Association 100(469) (2005) 36–50.

D.C. Dietz, Practical scheduling for call center operations, Omega 39 (2011) 550–557.

I. Castillo, T. Joro, Y.Y. Li, Workforce scheduling with multiple objectives, European Journal of Operational Research 196(1) (2009) 162–170.

A. Ingolfsson, E. Akhmetshina, S. Budge, Y. Li, A survey and experimental comparison of service level approximation methods for non-stationary $M(t)/M/s(t)$ queueing systems with exhaustive discipline, INFORMS Journal on Computing 19(2) (2007) 201–214.

B.K.P. Chen, S.G. Henderson, Two Issues in Setting Call Centre Staffing Levels, Annals of Operations Research 108(1-4) (2001) 175–192.

L.V. Green, P.J. Kolesar, W. Whitt, Coping with Time-Varying Demand When Setting Staffing Requirements for a Service System, Production and Operations Management 16(1) (2007) 13–39.

W. Whitt, What you should know about queueing models to set staffing requirements in service systems, Naval Research Logistics 54(5) (2007) 476–484.

M. Defraeye, I. Van Nieuwenhuyse, Setting staffing levels in an emergency department: opportunities and limitations of stationary queuing models, Review of Business and Economics 56(1) (2011) 73–100.

A. Ingolfsson, Modeling the $M(t)/M/s(t)$ queue with an exhaustive discipline, Working paper, University of Alberta, Canada (2005). Available online on http://www.business.ualberta.ca/aingolfsson/publications.htm

L.V. Green, P.J. Kolesar, A. Svoronos, Some Effects of Nonstationarity on Multiserver Markovian Queueing Systems, Operations Research 39(3) (1991) 502–511.

L.V. Green, P.J. Kolesar, The Pointwise Stationary Approximation for Queues with Nonstationary Arrivals, Management Science 37(1) (1991) 84–97.

W. Whitt, The pointwise stationary approximation for $M_t/M_t/s$, Management Science 37(3) (1991) 307-314.

L.V. Green, P.J. Kolesar, J. Soares, Improving the SIPP Approach for Staffing Service Systems That Have Cyclic Demands, Operations Research 49(4) (2001) 549–564.

L.V. Green, P.J. Kolesar, On the Accuracy of the Simple Peak Hour Approximation for Markovian Queues. Management Science 41(8) (1995) 1353–1370.

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

L.V. Green, P.J. Kolesar, The Lagged PSA for Estimating Peak Congestion in Multiserver Markovian Queues with Periodic Arrival Rates, Management Science 43(1) (1997) 80–87.

S.G. Eick, W.A. Massey, W. Whitt, The Physics of the $Mt/G/\infty$ Queue, Operations Research 41(4) (1993a) 731–742.

S.G. Eick, W.A. Massey, W. Whitt, $Mt/G/\infty$ Queues with Sinusoidal Arrival Rates, Management Science 39(2) (1993b) 241–252.

Z. Feldman, A. Mandelbaum, W.A. Massey, W. Whitt, Staffing of Time-Varying Queues to Achieve Time-Stable Performance, Management Science 54(2) (2008) 324–338.

O.B. Jennings, A. Mandelbaum, W.A. Massey, W. Whitt, Server Staffing to Meet Time-Varying Demand, Management Science 42(10) (1996) 1383–1394.

Y. Liu, W. Whitt, Stabilizing customer abandonment in many-server queues with time-varying arrivals, Working paper, Columbia University, New York, NY (2009). Available online at: http://www.columbia.edu/ ww2040/recent.html

D.L. Jagerman, Nonstationary blocking in telephone traffic, Bell Syst. Tech. 54 (1975) 625-661.

W.A. Massey, W. Whitt, An Analysis of the Modified Offered-Load Approximation for the Nonstationary Erlang Loss Model, The Annals of Applied Probability 4(4) (1994) 1145–1160.

W.A. Massey, W. Whitt, Peak congestion in multi-server service systems with slowly varying arrival rates, Queueing Systems 25(1) (1997) 157–172.

J.L. Davis, W.A. Massey, W. Whitt, Sensitivity to the Service-Time Distribution in the Nonstationary Erlang Loss Model, Management Science 41(6) (1995) 1107–1116.

W. Whitt, Engineering Solution of a Basic Call-Center Model, Management Science 51(2) (2005) 221–235.

F. Iravani, B. Balcioğlu, Approximations for the $M/GI/N + GI$ type call center, Queueing Systems 58(2) (2008) 137–153.

D. Gross, J.F. Shortle, J.M. Thompson, C.M. Harris, Fundamentals of Queueing Theory, 4th Edition, Wiley Series in Probability and Statistics, Wiley-Blackwell, 2008.

F. Campello, A. Ingolfsson, Exact Necessary Staffing Requirements based on Stochastic Comparisons with Infinite-Server Models, Working paper, University of Alberta, Canada (2011).

L.V. Green, J. Soares, Computing time-dependent waiting time probabilities in $M(t)/M/s(t)$ queueing systems, Manufacturing & Service Operations Management 9(1) (2007) 54-61.

L.F. Shampine, M.W. Reichelt, The MATLAB ODE Suite, SIAM Journal on Scientific Computing 18(1) (1997) 1–22.

A. Jensen, Markov Chains as an Aid in the Study of Markov Processes, Skand. Aktuarietidskrift 3 (1953) 87–91.

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

D. Gross, D.R. Miller, The randomization technique as a modeling tool and solution procedure for transient Markov processes, Operations Research 32(2) (1984) 343–361.

M.H. Rothkopf, S.S. Oren, A Closure Approximation for the Nonstationary $M/M/s$ Queue, Management Science 25(6) (1979) 522–534.

G.M. Clark, Use of Polya distributions in approximate solutions to nonstationary $M/M/s$ queues, Commun. ACM 24(4) (1981) 206–217.

M. Taaffe, K. Ong, Approximating nonstationary Ph(t)/Ph(t)/l/c queueing systems, Annals of Operations Research 8(1) (1987) 103–116.

E. Chassioti, D.J. Worthington, A New Model for Call Centre Queue Management, The Journal of the Operational Research Society 55(12) (2004) 1352–1357.

M. Brahimi, Approximating multi-server queues with inhomogeneous arrival rates and continuous service time distributions, PhD Dissertation, University of Lancaster, Lancaster, UK (1990).

M. Brahimi, D.J. Worthington, The finite capacity multi-server queue with inhomogeneous arrival rate and discrete service time distribution and its application to continuous service time problems, European Journal of Operational Research 50(3) (1991) 310–324.

A.D. Wall, D.J. Worthington, Using Discrete Distributions to Approximate General Service Time Distributions in Queueing Models, The Journal of the Operational Research Society 45(12) (1994) 1398–1404.

A.D. Wall, D.J. Worthington, Time-dependent analysis of virtual waiting time behaviour in discrete time queues, European Journal of Operational Research 178(2) (2007) 482–499.

S. Helber, K. Henken, Profit-oriented shift scheduling of inbound contact centers with skills-based routing, impatient customers, and retrials, OR Spectrum 32(1/4) (2010) 109–134.

W. Whitt, Fluid Models for Multiserver Queues with Abandonments, Operations Research 54(1) (2006a) 37–54.

S. Aguir, F. Karaesmen, O.Z. Akskin, F. Chauvet, The impact of retrials on call center performance, OR Spectrum 26(3) (2004) 353–376.

E. Altman, T. Jiménez, G. Koole, On the comparison of queueing systems with their fluid limits, Probability in the Engineering and Informational Sciences 15 (2001) 165–178.

T. Jiménez, G. Koole, Scaling and comparison of fluid limits of queues applied to call centers with time varying parameters, OR Spectrum 26(3) (2004) 413–422.

Y. Liu, W. Whitt, A Fluid Approximation for the $GI(t)/GI/s(t)+GI$ Queue, Working paper, Columbia University, New York (2010). Available online at: http://www.columbia.edu/ ww2040/allpapers.html

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

A. Mandelbaum, W.A. Massey, M.I. Reiman, R. Rider, Time varying multiserver queues with abandonments and retrials, Proceedings of the 16th International Teletraffic Conference 3 (1999a) 355–364.

A. Mandelbaum, W.A. Massey, M. I. Reiman, A. Stolyar, Waiting time asymptotics for time varying multiserver queues with abandonment and retrials, Proc. 37th Allerton Conf. Monticello, IL (1999b) 1095–1104.

A. Mandelbaum, W.A. Massey, M.I. Reiman, A. Stolyar, B. Rider, Queue lengths and waiting times for multiserver queues with abandonment and retrials, Telecommunication Systems 21(2-4) (2002) 149–171.

A. Mandelbaum, W.A. Massey, M. Reiman, Strong approximations for Markovian service networks, Queueing Systems 30(1) (1998) 149–201.

A.D. Ridley, M.C. Fu, W.A. Massey, Customer relations management: call center operations: Fluid approximations for a priority call center with time-varying arrivals, Proceedings of the 35th Conference on Winter Simulation, New Orleans, LA, 2 (2003) 1817–1823.

Y. Liu, W. Whitt, Large-Time Asymptotics for the $G_t/M_t/s_t + GI_t$ Many-Server Fluid Queue with Abandonment, Queueing systems 67(2) (2011b) 145–182.

Y. Liu, W. Whitt, The $G_t/GI/s_t + GI$ many-server fluid queue, Queueing Systems 71(4) (2012a) 405–444.

Y. Liu, W. Whitt, A many-server fluid limit for the $G_t/GI/s_t + GI$ queueing model experiencing periods of overloading, OR Letters 40 (2012b) 307–312.

Y. Liu, W. Whitt, A Network of Time-Varying Many-Server Fluid Queues with Customer Abandonment, Operations Research 59(4) (2011a) 835–846.

A.M. Law, W.D. Kelton, Simulation modeling and analysis, McGraw-Hill series in industrial engineering and management science, McGraw-Hill, Boston, 2000.

F. McGuire, Using simulation to reduce length of stay in emergency departments, In Proceedings of the 26th conference on Winter simulation (WSC '94), M.S. Manivannan, J.D. Tew (Eds.). Society for Computer Simulation International, San Diego, CA, USA (1994) 861–867.

M.L García, M.A. Centeno, C. Rivera, N. DeCario, Reducing time in an emergency room via a fast-track, In Proceedings of the 27th conference on Winter simulation (WSC '95), C. Alexopoulos, K. Kang (Eds.). IEEE Computer Society, Washington, 1995, 1048–1053.

G.W. Evans, T.B. Gor, E. Unger, A simulation model for evaluating personnel schedules in a hospital emergency department, In Proceedings of the 28th conference on Winter simulation (WSC '96), J.M. Charnes, D.J. Morrice, D.T. Brunner, J.J. Swain (Eds.), IEEE Computer Society, Washington, 1996, 1205–1209.

S. Takakuwa, H. Shiozaki, Functional analysis for operating emergency department of a general hospital, In Proceedings of the 36th conference on Winter simulation(WSC '04). Winter Simulation Conference (2004) 2003–2011.

G.R. Hung, S.R. Whitehouse, C.B. O'Neill, A.P. Gray, N. Kissoon, Computer Modeling of Patient Flow in a Pediatric Emergency Department Using Discrete Event Simulation, Pediatric Emergency Care 23(1) (2007) 5–10.

Ahmed, M.A., T.M. Alkhamis. 2009. Simulation optimization for an emergency department healthcare unit in Kuwait, European Journal of Operational Research 198(3) (2009) 936–942.

M. Pitt, A generalised simulation system to support strategic resource planning in healthcare, In Proceedings of the 29th conference on Winter simulation (WSC '97), S. Andradottir, K.J. Healy, D.H. Withers, B.L. Nelson (Eds.). IEEE Computer Society, Washington, 1997, 1155–1162.

D. Sinreich, Y.N. Marmor, A simple and intuitive simulation tool for analyzing emergency department operations, In Proceedings of the 36th conference on Winter simulation(WSC '04). Winter Simulation Conference (2004) 1994–2002.

A. Fletcher, D. Halsall, S. Huxham, D. Worthington, The DH Accident and Emergency Department model: a national generic model used locally, Journal of the Operational Research Society 58 (2007a) 1554–1562.

A. Fletcher, D.J. Worthington, What is a "generic" hospital model? Working Paper, Department of Management Science, Lancaster University, UK (2007b).

M.M. Gunal, M. Pidd, Understanding target-driven action in emergency department performance using simulation, Emergency medicine journal 26(10) (2009) 724–727.

M.F. Neuts, Matrix-geometric solutions in stochasticmodels, Johns Hopkins University Press, Baltimore, 1981.

G. Latouche, V. Ramaswami, Introduction to Matrix Analytic Methods in Stochastic Modeling. ASA-SIAM Series on Statistics and Applied Probability, Philadelphia, 1999.

T. Osogami, Analysis of multiserver systems via dimensionality reduction of Markov chains, PhD thesis, School of Computer Science, Carnegie Mellon University (2005).

V. Ramaswami, A stable recursion for the steady state vector in Markov chains of $M/G/1$ type, Stochastic Models, 4(1) (1988) 183–189.