Setting staffing levels in systems with time-varying demand: 
the context of an emergency department 

Mieke Defraeye and Inneke Van Nieuwenhuyse
Setting staffing levels in systems with time-varying demand: the context of an emergency department

Mieke Defraeye and Inneke Van Nieuwenhuyse

Research Center for Operations Management, Department of Decision Sciences and Information Management, K.U.Leuven

Abstract. Many service systems are characterized by time-varying demand for service. For instance in an emergency department, patient arrival rates are usually not constant throughout the day. This arrival process is stochastic, but nonetheless predictable to some extent (a daily pattern can often be distinguished). However, this feature can severely complicate the process of determining appropriate staffing levels throughout the day.

In this paper, an overview of available methods for setting staffing levels under time-varying demand for service is given, with the applicability and appropriateness in the specific context of an emergency department as the main point of interest. An important goal of an emergency department is to strive for patient waiting times that are sufficiently low for all patients, independent of the arrival time, and therefore we particularly emphasize the importance of using performance measures related to waiting times.

1 Introduction

In many real life service systems, the demand for service is not constant over time, i.e. fluctuations are usually present on a daily, weekly, monthly or yearly basis. The question that is addressed here, is how personnel capacity planning can be used to deal with this time-varying nature of demand. Moreover, we focus on a healthcare setting and consequently, some characteristics specific to this context should be taken into account.

In general, variability in demand is to some extent predictable and can be dealt with in several ways, for instance by making appropriate capacity decisions, choosing a suitable appointment system or using waiting lists (Hall et al., 2006). However, these last two options are often not appropriate nor possible, especially in for instance an emergency department (ED), which results in a full tradeoff of capacity versus waiting times. This is why the emergency department of a hospital seems like an ideal starting point for an analysis of the time-varying nature of demand in a healthcare setting.
Figure 1 Example of a simplified ED system

An emergency department can be seen as a network in which patients arrive according to some time-varying arrival process and subsequently move through several process steps in order to receive treatment (an illustration of a simplified ED system is provided in Figure 1). Relevant characteristics of an ED include the following:

- **The routing of patients is typically not fixed**, i.e. the number and type of process steps tends to be quite different from patient to patient (Williams and Crouch, 2006). In Figure 1, an example path followed by two types of patients is shown.
- **The service process can be nonstationary**, It is not unlikely that physician examination times are influenced by the number of patients waiting for treatment. This means that doctors have an incentive to work faster during busy periods, in order to prevent the workload from increasing further. Moreover, multitasking is usually present (e.g. physicians interrupting one patient’s examination for some minutes to check lab results of another patient). This complicates modeling the ED process and can make obtaining real-life data on the service process problematic.
- **Resources are usually highly utilized**, i.e. overloading can be present. This means that the arrival rate is larger than the overall service capacity at some points in time, causing a buildup of patients waiting in queue. Green et al. (2007) state that this is a dominant feature that has a major impact on performance. Therefore, substantial overloading should not be neglected.
- **Patients are treated based on urgency**, The more severe the patient’s illness or injuries are, the sooner he should be treated. For this purpose, priorities are assigned to each patient by means of an urgency code that determines the sequence in which patients are treated (Moll (2010), George et al. (1992)).
- **Customer impatience is present**, which results in so-called *abandonments*, or equivalently *left without being seen* (Johnson et al. (2009), Pham et al. (2009)). If a patient’s waiting time becomes too long, he might choose to leave the ED without being seen by a doctor. As is mentioned in Green et al. (2007), the presence of abandonments can be advantageous because this often has a positive effect on the stability of the system, especially when some overloading is present in the system.
- **The demand for service is time-varying**, meaning that the number of patients arriving at the ED is not constant over time (Flottemesch et al. (2007), McCarthy et al. (2008), Green and Kolesar (1995), Green et al. (2006)). Fluctuations can occur according to a daily, weekly, monthly or yearly pattern (Hall et al., 2006). For instance, the number of patients arriving at

---

1 In Green et al. (2006), this problem is dealt with by using physician workload estimations that are available in the literature (see for instance Graff et al. (1993)).
the ED is usually higher during the day compared to at night, and during the weekend nightly arrival rates are often higher than on weekdays. Here, only variability on a daily basis is considered seen that these fluctuations are expected to be the most outspoken (Hall et al., 2006). The number of patients that is expected to arrive per hour at an ED usually does not change severely from day to day and thus can be predicted to some extent. However, random fluctuations around these hourly averages are usually present and it is this stochastic variability that complicates the process of determining staffing requirements. Certain disruptions in the demand process (e.g. because of a disaster) are less predictable; these are not considered.

As an illustration, the hourly arrival rates at the ED of the St.-Elisabeth Hospital in Turnhout are plotted in Figure 2a (plotted separately for weekdays and weekends). Figure 2b displays an additional example of time-varying arrival rates in a NY city emergency department (taken from Green et al., 2006). Clearly, both graphs display similar arrival rate patterns, e.g. a distinct peak can be distinguished around noon and nightly arrival rates are rather low.

![Figure 2 Examples of time-varying arrival rates](image)

Given the many complexities present in an ED setting, it is impossible to take them all into account. In our research, we wish to focus on the impact of time-varying arrivals (which may cause the system to be temporarily overloaded), and customer abandonments. A single-stage system will be considered, which is explained in further detail in Section 2.

## 2 Problem description

In the PhD research, the primary focus lies on $M(t)/G/s(t)+G$ queues, i.e. single-stage multiserver queues in which patients arrive based on a nonstationary Poisson arrival process, with nonexponential service and abandonment times and a time-varying number of servers. Let $\mu$ denote the service rate and $\lambda(t)$ the Poisson arrival rate at time $t$ ($\bar{\lambda}$ represents the average arrival rate over the considered time horizon $T$). The main goal is then to determine an appropriate staffing requirement function $s(t)$, which defines the number of servers to be assigned at each time instant $t$ (with $0 \leq t \leq T$). In doing so, the desired system performance (e.g. targets w.r.t. the length of waiting times) must be taken into consideration. Moreover, the number of servers is usually only allowed to change at fixed points in time. The planning period over which the number of servers is assumed to remain unchanged will be referred to as the staffing interval.

### A nonstationary arrival process

Whereas arrival rates are assumed to be constant in most queueing models, we focus on a nonstationary Poisson arrival process, in which arrival rates vary over time. Although regular steady state results are
no longer valid in this situation, a periodic steady state is assumed. Roughly spoken, periodic steady
with period \( T \) implies that the system performance at time \( t \) does not differ much from the performance
at time \( t + T \), for all \( t \) (a more formal definition can be found for instance in Heyman and Whitt
(1984)). Similar to the well-known stationary results, a prerequisite for system to be in a periodic
steady state is that \( \lambda < \bar{\mu} \) (the only difference lies in the use of averages for the arrival rate and the
number of servers), meaning that the average traffic intensity \( \bar{\rho} = \frac{\lambda}{\bar{\mu}} \) does not exceed one (Hall et al.,
2006).

With respect to the cycle length, a policy with daily cycles is often adopted (so \( T \) equals 24
hours), in which 1 peak occurs per cycle, as was illustrated in Figure 2. For modeling convenience
however, arrival rates are often approximated by a sinusoid (e.g. see Green et al. (1991), Ingolfsson
et al. (2010), Green et al. (2001), Whitt (1991), Liu and Whitt (2009)). In addition, arrival rates are
frequently modeled as a piecewise constant function, because real-life data are often solely available
on an aggregated basis (e.g. per hour) and some methods to determine staffing requirements assume that
arrival rate is constant over the staffing interval anyway (Ingolfsson et al., 2007).

The question rises to what extent the nonstationarity in the arrival process impacts the system’s
performance. In Green et al. (1991), the effects of nonstationarity are explored for sinusoidal arrival
patterns in \( M(t)/M/s \) systems. The concept of nonstationarity is determined by the relative amplitude on
the one hand \( RA = \frac{A}{\bar{\lambda}} \) where \( A \) represents the amplitude of the sine and \( \bar{\lambda} \) refers to the average arrival
rate over the time horizon) and the frequency of events on the other hand. Increasing event frequencies
correspond to an augmentation in the arrival and service rates whilst keeping the traffic intensity fixed
and are assumed to result in a more stationary process. System performance is measured by numerical
integration of the ordinary differential equations for the \( M(t)/M/s \) model and afterwards averaging the
obtained delay probability, expected waiting time and expected queue length measures over the day.
The results are somewhat predictable, as daily expected delay, daily delay probability and daily
expected queue size increase as the system becomes more nonstationary. However, nonstationarity in
terms of the relative amplitude has a larger impact on the daily expected delay as the average traffic
intensity \( \bar{\rho} = \frac{\lambda}{\bar{\mu}} \) increases.

The distribution of the service and abandonment process
Regarding the distribution of the service process, Bolotin (1994) and Brown et al. (2005) state that a
Markovian assumption is often not appropriate, because non-exponential service time distributions are
quite common in practice. Moreover, ignoring non-exponentiality appears to have a larger impact for
the abandonment process than for the service process (Whitt, 2005). For these reasons, no restrictions
will be placed on the service time and time-to-abandon distributions. Moreover, we assume that the
service process distribution does not change over time, so nonstationarity in the service process is not
considered.

An exhaustive service policy
Many staffing methods in literature implicitly assume a \textit{pre-emptive} service discipline, where service is
interrupted and customers rejoin the queue when a server is scheduled to leave. In an emergency
department however, an \textit{exhaustive} service policy is more appropriate. Instead of rejoining the queue,
customer service is completed before the server leaves, even if this means that a server has to work
beyond his scheduled shift. This has important implications for the model and consequently, omitting
this feature might lead to inferior performance (Ingolfsson et al., 2007). Recently, more attention is
given to this system characteristic, because of its importance in many service systems (e.g. in Liu and
Whitt (2009) an additional constraint is added to account for this feature). An extension of this
exhaustive service policy is suggested in Ingolfsson (2005): if a service completion occurs at a time
instant close to the scheduled server departure, no new customer is accepted.
Performance measures
When deciding on staffing levels, one first needs to specify what defines a “good” result. An appropriate performance measure has to be chosen and a target must be set w.r.t. that performance measure. The performance target should be interpreted as a goal rather than a strict constraint (Green et al., 2001). There are numerous possibilities concerning the choice of appropriate performance measures (an overview of some ED-specific performance measures is provided in Welch et al. (2006)), but here we limit ourselves to a brief description of performance measures that are either interesting in an ED context or used frequently in the staffing models available in the literature (see for instance Green et al., 2007):

- **Delay probability**: This is the probability that upon arrival, a customer (or patient) cannot be helped immediately and therefore has to wait until a server (or physician) is idle. This measure is often used in the literature on time-varying demand for service (e.g. Green et al., 2003; Feldman et al. (2008)). Advantages include insensitivity to model details, an interpretation independent of scale (Halfin and Whitt, 1981) and ease of computation (Green et al., 2007). Moreover, choosing delay probability targets can be convenient for some staffing methods, because the link between the desired performance and the number of servers needed to obtain these results, can be established easily if the delay probability is chosen as performance measure (Whitt, 2007). A performance target related to the delay probability could then be defined as in Feldman et al. (2008), namely: choose the staffing level in each interval as small as possible, while meeting the condition that delay probability must lie below some target in all staffing intervals.

- **Expected delay**: In this case, the aim is to obtain waiting times that – on average – lie below some target value (e.g. see Green and Kolesar (1995), Liu and Whitt (2009)). Usually virtual waiting times are used, i.e. the time that a customer would have to spend in queue if he were to arrive at a specific point in time (Gross et al. (2008)). An important advantage of this performance measure is that only little information is needed on the distribution of the waiting time. This advantage is directly linked to the main drawback: when using expected delays, the remainder of the waiting time distribution is neglected. While the length of the wait might be acceptable on average, waiting times might still be intolerably high for some customers, which is of course to be avoided.

- **Service levels**: These can be specified in terms of queue length or waiting times. The former implies controlling for the probability that the queue length is too high, whereas the latter seems more suited to an ED context, as it rather focuses on avoiding large waiting times. A service level target could be to aim at a sufficiently large probability that a patient (who has not abandoned) has a waiting time shorter than some chosen maximal waiting time (Green et al., 2007), i.e. for each patient, there should be only a small chance that the waiting time is “too long”. An example of such a service level constraint can be expressed as

\[ P(W(t) \leq \tau \text{ and served}) \geq x \quad \text{for all } t \]

where \( W(t) \) represents the virtual waiting time of a customer arriving at time \( t \) and \( \tau \) denotes the maximal allowed waiting time. In words, this means that the probability of an excessive wait should be sufficiently small for all patients, \textit{independent of the time of arrival}. The importance of meeting a performance goal performance \textit{stably} during the day is emphasized by Feldman et al. (2008) and Ingolfsson et al. (2010).

\[ \text{For a number of other performance measures based on the delay probability, we refer to Green et al. (2001).} \]

\[ \text{Expected waiting time can be conditioned on whether the customer enters service or not (see for instance the performance measures used in Whitt (2005) and Feldman et al. (2008) or on waiting times being larger than zero (e.g. in Feldman et al. (2008)).} \]
• **Abandonment probability**: If the systems is characterized by abandonments, then from a managerial perspective it might be desirable to keep the abandonment probability low, rather than to control for the often used delay probability. Intuitively, this performance measure represents customers who leave the system because the waiting time was too long and is therefore related to the customer’s perception of service.

• **Cost**: Keeping labor costs as small as possible, while providing sufficiently good service is an obvious managerial objective. In fact, a tradeoff can be made between labor costs on the one hand, and costs related to the *quality of service* (QoS) on the other hand, as pointed out by Borst et al. (2004). Quality of service is related to customer satisfaction, and a low QoS can be reflected through a penalty for long waiting times or abandonments. The emphasis to be placed on either labor costs or QoS costs depends on the specific problem setting. Based on this observation, three different operational regimes have been suggested by Garnett et al. (2002). In the *efficiency-driven regime*, labor costs are substantially higher than the cost of a low QoS and therefore emphasis is placed on minimizing the staffing levels, which generally results in higher delay probabilities and severely loaded systems. On the other end of the spectrum lies the *quality-driven regime*, where a high QoS is considered more important than keeping labor costs low, which rather leads to overstaffing. The *rationalized or quality-and-efficiency driven (QED)* regime can be situated between these two extremes; here the aim is to provide good service, whilst keeping staffing costs acceptably low. For details on these operational regimes, we refer to Garnett et al. (2002), Koole and Mandelbaum (2002), Zeltyn and Mandelbaum (2005).

In a healthcare setting, the probability of delay does not seem appropriate, since this performance measure does not account for the *length* of the wait. For instance, it seems reasonable to assume that most patients in an ED are indifferent between very small waiting times (e.g. 1 minute) and a zero waiting time. Moreover, focusing on average waiting times does not seem suitable either, because even if the waiting time is sufficiently low *on average*, there might still be some patients who have to wait unacceptably long before receiving treatment. More important in this context is the length of the waiting time and in particular the length of the *longest* waiting times. This is why waiting time percentiles (or service levels) appear to be the most interesting performance measure to focus on in an ED context. Moreover, focusing on large waiting times implicitly addresses the problem of abandonments, as abandonments only occur when waiting times are considered to be too long. Next, as service needs to be provided preferably at the lowest possible cost, it might be insightful to place an additional focus on the labor costs related to a staffing level function, instead of considering solely waiting time related performance measures (e.g. Atlason et al. (2004) minimize staffing costs while keeping service levels acceptable). However, labor costs appear not to be of utmost importance: Ingolfsson et al. (2002) refer to a survey among managers, which lead to the conclusion that several criteria such as providing better service, meeting constraints on personnel availability and simple staffing methods are often considered more important than reducing labor costs. For this reason, our main performance measure of interest is the service level.

It appears that in the literature, only little attention has been given to the use service level performance measures in staffing methods, compared to delay probability. In addition, available methods mainly target at call center applications, with often substantially different system characteristics than an ED. This emphasizes the importance of exploring methods for setting staffing levels under time-varying demand for service, with a focus on performance measures that are related to the length of patient’s waiting times.

---

4 The abandonment probability is sometimes conditioned on waiting times being larger than a target value (e.g. Mandelbaum and Zeltyn, 2005). An additional refinement that is applied occasionally, is to exclude customers with very small waits from the abandonment probability (Mandelbaum and Zeltyn, 2004).
3 Literature review

An excellent overview of past research can be found in Green et al. (2007) and Whitt (2007). Several methods that are applicable for dealing with time-varying demand when determining staffing requirements in an M(t)/GI/s(t)+GI environment, are touched upon. The emphasis mainly lies on telephone call center applications, although other examples are also provided (e.g. an application in an emergency department). Our aim is to extend this work by including some recent research that has been done in the domain of service systems with time-varying demand patterns. However, all methods are discussed with their specific applicability within the context of an emergency department in mind.

When choosing appropriate staffing levels if demand for service varies over time, two aspects are important (Ingolfsson et al., 2007). On the one hand, there is the problem of finding staffing requirements that satisfy the chosen performance target. An example in an ED context might be to strive for a sufficiently small probability that a patient's waiting time exceeds some predetermined maximum wait. On the other hand, the system performance that results from a given staffing level function, must be assessed. Therefore, one needs a method to evaluate this probability (ideally with high accuracy and limited computational effort).

For ease of reference, Table 1 gives an overview of the different performance evaluation methods that are discussed in the remainder this section, along with some key references and information on intended use, advantages and drawbacks of each method. Table 2 gives an overview of the staffing methods that will be covered.

The remainder of the section is organized as follows: in Section 3.1, we discuss the performance evaluation methods that rely on stationary results in order to approximate the nonstationary system, along with their staffing methods (for ease of reference, these are indicated in green in Tables 1 and 2). Section 3.2 focuses on the methods that explicitly take into account the nonstationarity (indicated in yellow in Tables 1 and 2).

<table>
<thead>
<tr>
<th>PERFORMANCE EVALUATION METHODS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>SSA</td>
</tr>
<tr>
<td>SPHA</td>
</tr>
<tr>
<td>PSA</td>
</tr>
<tr>
<td>SIPPP &amp; Segmented PSA</td>
</tr>
<tr>
<td>Models with time lag (Lag SIPP, Lag PSA)</td>
</tr>
<tr>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Green et al. (2001)</td>
</tr>
<tr>
<td>Green et al. (2003)</td>
</tr>
<tr>
<td>MOL &amp; IS</td>
</tr>
<tr>
<td>Massey and Whitt (1994)</td>
</tr>
<tr>
<td>Massey and Whitt (1997)</td>
</tr>
<tr>
<td>Liu and Whitt (2009)</td>
</tr>
<tr>
<td>Jennings et al. (1996)</td>
</tr>
<tr>
<td>Effective arrival rate approximation</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Simulation</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Numerical integration ODE’s</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Discrete time modeling</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Randomization</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Closure approximations</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Fluid models</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

*Table 1 Summary of performance evaluation methods*
## STAFFING ALGORITHMS

<table>
<thead>
<tr>
<th>Method</th>
<th>Relevant literature</th>
<th>Intended use &amp; advantages</th>
<th>Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose smallest staffing level satisfying constraint</td>
<td>E.g. Green et al. (2001)</td>
<td>Intuitive method</td>
<td>Inefficient</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Computationally expensive (especially in large systems)</td>
</tr>
<tr>
<td>SRS-rule</td>
<td>Whitt (1992)</td>
<td>Simple rule of thumb</td>
<td>Choice of appropriate parameter not always easy (depends on target performance measure)</td>
</tr>
<tr>
<td></td>
<td>Koole and Mandelbaum (2002)</td>
<td>Robust in many situations (empirically and theoretically validated)</td>
<td>Guideline, so better solutions might exist</td>
</tr>
<tr>
<td>ISA</td>
<td>Feldman et al. (2008)</td>
<td>Efficient staffing updates</td>
<td>Generalization to alternative performance measures unclear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fast convergence</td>
<td>Possibly not effective in small systems (effect of a single server is considered negligible)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>General applicability and automatic validation</td>
<td>Does not consider staffing intervals</td>
</tr>
<tr>
<td>Scheduling algorithms</td>
<td>Henderson and Mason (1998)</td>
<td>Staffing requirements and corresponding schedule determined simultaneously</td>
<td>Computationally expensive</td>
</tr>
<tr>
<td></td>
<td>Ingoftsson et al. (2002)</td>
<td>Both labor costs and service levels are considered</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Atlason et al. (2004, 2005)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 2 Summary of staffing methods

### 3.1 STATIONARY APPROXIMATIONS

A first way to deal with time-varying nature of the demand for service, is to try to approximate the nonstationary, time-varying system by one (or more) related stationary model(s). To obtain staffing levels for a nonstationary system using a stationary model, one needs a way to translate the nonstationary system to a similar stationary system. That is, the system characteristics of the nonstationary model have to be translated into appropriate input parameters for the stationary model(s). Often multiple stationary models are applied on smaller intervals, i.e. by dividing the time horizon into subintervals and subsequently applying a stationary model in each separate interval.

Using several stationary models in a nonstationary environment, implies that some assumptions are made (Green et al., 2001), among others:
- Statistical independence of delays between separate intervals.
- Steady state is reached in each interval
- Arrival rate remains constant over the staffing interval

Of course, the validity of these assumptions influences the performance of the resulting staffing levels. Whereas appropriateness of the first two assumptions tends to increase as staffing intervals become longer and service times smaller (cfr. Green et al., 2001) the last assumption is more likely to be approximately valid for shorter staffing intervals (Green et al., 2007). An important additional implication of using stationary models is that no overloading should be present, as this would cause instability in the stationary model.

In what follows, we give an overview of how stationary models can be used when dealing with a system that is in fact nonstationary. In Section 3.1.1 the problem of translating a nonstationary system to appropriate input parameters for one (or more) stationary model(s) and calculating its performance,
is addressed. Section 3.1.2 deals with determining appropriate setting staffing levels based on the obtained stationary results.

### 3.1.1 How to use stationary models to approximate nonstationary systems

#### a) Simple Stationary Approximation (SSA)

The first and most straightforward way to approximate a system with time-varying arrival rates by a stationary system, is to apply a so-called Simple Stationary Approximation (SSA). The idea is to ignore the nonstationarity in the arrival process and simply use the average arrival rate over the entire time period as an input parameter for one single stationary model (Green et al., 1991). In practice this approach is used frequently, although not surprisingly it often leads to poor results, because the time-varying nature of demand is neglected.

**SSA** : apply one M/G/s+G model with arrival rate $\lambda_{SSA} \equiv \bar{\lambda} = \frac{1}{T} \int_0^T \lambda(t) dt$

Some results on the accuracy of this method are provided in Green et al. (1991). A comparison is made between the actual performance of the nonstationary system on the one hand, which is determined by numerically solving the differential equations (ODE’s) for the M(t)/M/s model, and the performance for the stationary model with an arrival rate equal to the daily average arrival rate on the other hand. Arrival rates are assumed to vary according to a daily sinusoid pattern (Poisson distributed), with a cycle length of one day. The performance measures of interest are the expected delay, queue length and the probability of delay. To allow for a comparison between the performance measures obtained by numerically solving the system’s ODE’s (available *instantaneously*) and SSA performance (where only *one value is available for the entire time horizon*), the actual performance values will be averaged over the time horizon. Relative errors, defined as \( \frac{\text{actual value} - \text{stationary value}}{\text{actual value}} \), are calculated. Note that all performance measures concern daily (customer) averages, so no information is provided about the variability in performance throughout the cycle. Experiments were performed to check under what conditions (amplitude, frequency of events, traffic intensity, system size) simply ignoring the nonstationarity of the system leads to sufficiently accurate results. In short, Green et al. (1991) conclude that the SSA systematically underestimates performance and therefore provides a lowerbound on all considered performance measures. The SSA performs poorly even for systems with a low degree of nonstationarity and its accuracy deteriorates as the relative amplitude \( RA = \frac{A}{\lambda} \), system size (measured by the number of servers) or event frequency (i.e. the magnitude of arrival and service rates, for fixed traffic intensity) increase. From a practical point of view, ignoring nonstationarity seems only acceptable for small systems (namely 1 or 2 servers) with very low relative amplitudes (<10%) and event frequencies (low arrival and service rates). According to Jennings et al. (1996), the SSA can be used successfully when service times are long, compared to the change in arrival rate.

#### b) Simple Peak Hour Approximation

A second approach that approximates the nonstationary system with only one stationary model, relies on the idea that it is sufficient to choose staffing levels such that good performance is guaranteed during the peak period. This is referred to as Simple Peak Hour Approximation (SPHA) and is described in Green and Kolesar (1995, 1997), although a similar principle was proposed in Bear (1980), named *busy-hour engineering*. The system’s performance is approximated by a stationary model, with an arrival rate equal to the average arrival rate over the peak hour (i.e. the hour
surrounding the time instant at which the arrival rate peaks) or peak epoch\(^5\) (i.e. the time instant at which the arrival rate peaks). SPHA can lead to good results for high service rates (e.g. 100 customers per hour or more), again because steady state is then reached faster (Green and Kolesar, 1995). When staffing intervals are long (consider for instance 8-hour shifts), SPHA might be a good choice as well, provided that the targeted quality of service is high (because staffing for peak demand obviously results in overcapacity during calmer periods).

\[ \text{SPHA : apply one M/G/s+G model, with arrival rate } \lambda_{\text{SPHA}} = \frac{1}{x_z-x_1} \int_{x_1}^{x_z} \lambda(t) \, dt \]

where \([x_1, x_z]\) represents the peak period

c) Pointwise Stationary approximation (PSA)

In the Pointwise Stationary Approximation (PSA) implies that steady state performance measures are calculated for each moment in time, using the arrival rate at each moment in time instant in a stationary model (Green and Kolesar (1991), Jennings et al. (1996), Whitt (1991)). Consequently, PSA does not account for the presence staffing intervals (during which the capacity remains unchanged).

\[ \text{PSA : apply a M/G/s+G model on each time instant } t, \text{ with arrival rate } \lambda_{\text{PSA}} = \lambda(t) \]

The effectiveness of this approach is discussed in Green and Kolesar (1991). Analogous to Green et al. (1991), daily average expected queue length, daily average expected waiting time, daily average probability of delay and daily average probability of all servers busy are chosen as performance measures (again, these are all averages over the time horizon of one day). Relative errors are calculated for each measure, this time comparing the actual performance to the PSA performance. The effect of changes in system parameters (in an M(t)/M/s system) are explored for sinusoidal Poisson arrival patterns. The parameters that are varied include the event frequency, service rate and maximal traffic intensity (equal to the maximum of the time-dependent traffic intensity \( \rho(t) = \frac{\lambda(t)}{\mu}, \) which is influenced by the amplitude of the arrival rate, the average arrival rate and the number of servers).

PSA becomes more accurate as event frequency increases, i.e. as the average arrival rate \( \lambda \) and service rate \( \mu \) increase with the same amount, while keeping the offered load constant (or equivalently, as the time horizon becomes longer); a result that is consistent with Green et al. (1991), where empirical evidence implied that the expected queue length and delay probability asymptotically approach the PSA as the rates increase (provided that the PSA exists) and. A formal proof of this is provided in Whitt (1991), who shows for M(t)/M/s queues that PSA is asymptotically correct as event frequency increases. Alnowibet (2004) point out that this effect is due to the steady state being reached more quickly for higher arrival and service rates, leading to a better correspondence to steady state behavior at each moment in time (which is implicitly assumed). Increasing the arrival or service separately lead to the following conclusions: effects concerning an increase in the arrival rate are unclear, but performance can be estimated more accurately for higher service rates. Accuracy seems acceptable for sufficiently high service rates, e.g. of at least 2 customers per hour (Green and Kolesar, 1991). Results on the effect of increasing maximal traffic intensity indicate that the accuracy of PSA decreases as maximal traffic intensity approaches one. For a maximal traffic intensity that exceeds 1, PSA does not produce finite values anymore, because of instability in the stationary models. For all components influencing the maximal arrival rate (amplitude, average arrival rate and number of servers), the effect is as could be expected: larger amplitudes and average arrival rates result in larger errors, whereas an increase in the number of servers leads to lower errors. Also, experiments with

\(^5\) Using a peak period (e.g. an hour) or a peak epoch only lead to minor changes in results.
varying amplitudes indicate that PSA is most appropriate when arrival rate changes relatively slowly compared to the service rate. This is intuitive: over an interval in time, there is a higher resemblance between a slowly fluctuating arrival rate function and a constant arrival rate (i.e. the starting point for a stationary model), compared to an arrival rate function that changes more rapidly. Because PSA implies that a stationary model is applied on each moment in time, this approach will evidently perform better as the resemblance to a stationary system increases. In Jennings et al. (1996), this intuition is confirmed for infinite server models with a sufficiently smooth arrival rate, by means of the quadratic approximation for the offered load that was provided in Eick et al. (1993a) (which will be discussed more in detail in a next subsection). There, it was shown that PSA is approximately correct (and thus more effective) if the arrival rate changes slowly and if service times are small. It has been proven that the PSA approximation provides an upper bound on the expected queue size in an M(t)/M/s system under the condition that stationary models are applicable, i.e. traffic intensity never exceeds one (cfr. Grassmann (1983), who first used this approach, and Green et al. (1991)).

The PSA approach does not impose capacity to remain constant throughout a staffing interval. To account for the fact that capacity can usually only be altered at specific points in time, two refinements to the PSA approach can be used: SIPP and Segmented PSA.

d) PSA-based approaches accounting for staffing intervals

Intuitively, SSA and PSA can be considered as two opposites. In SSA, arrival rate is averaged over the entire time horizon and used in one single stationary model, whereas the PSA approach uses no averaging at all and applies a stationary model on each moment in time. According to Whitt (1991), an approach that lies between these two extremes (for instance by averaging arrival rates over an interval) could be useful, although it is not clear how the length of this interval should be determined. A common choice is 30 minutes (cfr. Green et al., 2001) but an interval length proportional to the expected service rate (Whitt, 1991) can be used as well. As stated in Massey and Whitt (1996), this interval should be small enough to obtain a sufficiently precise performance measure, yet large enough to capture the “relevant past” within the interval (particularly if independence between intervals is assumed).

Stationary independent period-by-period method (SIPP)

One approach to account for the length of staffing intervals is to use a stationary model in each interval, with the arrival rate averaged over that interval as input parameter. This is commonly referred to as the Stationary Independent Period-by-Period (SIPP) approach (Green et al., 2001), although in Whitt (1991) a similar approach is proposed, named the Average Stationary Approximation. From a practical point of view, note that SIPP is advantageous in the sense that it tends to avoid arrival rate estimation and smoothing issues, because call centres often only provide piecewise constant arrival rate functions. An application of SIPP to set staffing levels in an ED, is provided in Green (2006).

SIPP : apply one M/G/s+G model per interval $i$ with arrival rate $\lambda_i \equiv \frac{1}{x_2-x_1} \int_{x_1}^{x_2} \lambda(t)dt$

where $[x_1, x_2]$ represents the $i^{th}$ staffing interval

Some results on the accuracy of the SIPP approach in an M(t)/M/s(t) model for varying model parameters, are given in Green et al. (2001). The reliability of the SIPP method as a function of arrival rate, service rate, relative amplitude, target delay probability and the length of the staffing interval is examined. All considered performance measures are related to the delay probability. In the SIPP method, arrival rates are averaged over the staffing interval, therefore it can be expected that its accuracy depends upon the length of this interval. Indeed, the main result is that SIPP should be avoided when the arrival rate changes substantially over the staffing interval. This implies that SIPP
performance can be expected to be poor for long staffing intervals or arrival rates varying with a large (relative) amplitude. Moreover, SIPP is most appropriate for short service times (as is also confirmed in Green et al. (2007), who provide an example with an average service time of 300 minutes where the SIPP method fails to achieve reasonable performance). This effect is probably related to the assumption of independence between consecutive staffing intervals: within each staffing interval, steady state is reached faster for larger service rates. Results concerning the effect of changes in the average arrival rate are not consistent, although SIPP performance tends to deteriorate slightly as the average arrival rate increases. Furthermore, SIPP seems most appropriate for small systems, and only works well if no overloading is present, because this would cause instability in the stationary models (Green et al., 2007).

However, the SIPP approach has some downsides, for example the underestimation of staffing requirements in staffing intervals where the arrival rate decreases (this is due to the use of the averaged arrival rate over an interval). Therefore, to further improve the method, three refinements to the SIPP approach are provided in Green et al. (2001).

In the **SIPP Max** approach, the maximum arrival rate over staffing interval is used, instead of the average arrival rate. As can be expected, this approach is always more reliable, but unfortunately at the expense of higher staffing costs, as the staffing function will always be higher than SIPP staffing. In an attempt to combine the reliability of SIPP Max while keeping staffing costs as low as possible, the **SIPP Mix** approach was developed. Here, SIPP is applied if the arrival rate is strictly increasing over interval and SIPP Max is used otherwise. Performance turned out to be better than SIPP in some settings, although in general, it does not outperform the original SIPP approach. A third extension of SIPP, **Lagged SIPP**, was presented as well but this will be discussed in a next subsection.

**Segmented PSA**

Another way to keep staffing levels constant during a staffing interval is to set the staffing level equal to the maximum of the PSA staffing requirements over that interval, which is referred to as the **Segmented PSA** (Green et al., 2007). This way, staffing levels may turn out to be too high, but these can be improved afterwards, for instance by means of simulation. Segmented PSA is equivalent to the SIPP Max approach discussed in the previous subsection.

<table>
<thead>
<tr>
<th>Segmented PSA :</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. PSA: apply a M/G/s+G model on each time instant $t$, with arrival rate $\lambda_{PSA} \equiv \lambda(t)$</td>
</tr>
<tr>
<td>2. Staffing level $s_{SegmentedPSA}(t)$ in interval $i$ = max$<em>{t \in interval}(S</em>{PSA}(t))$</td>
</tr>
</tbody>
</table>

e) **Models with a time lag**

When comparing the arrival rate to the actual offered load, it becomes clear that the peaks in actual offered load lag the arrival rate peaks (Green and Kolesar, 1995). This is illustrated in Figure 3, where the actual offered load, the PSA offered load, and the arrival rate are plotted for an example setting. Apparently, the size of the time lag is proportional to the expected service time, as can be deduced by comparing Figure 3 with Figure 4, where arrival rate and offered loads are plotted for small and large service rates respectively. Green et al. (2001) describe the concept of such a **time lag** intuitively as “the amount of time during which the arrival rate at a given epoch will continue to impact system congestion” (Green et al., 2001). As each customer stays in the system for $E[S] = \frac{1}{\mu}$ time units on average, the presence of a link between this time lag and the service time seems reasonable. As the PSA offered load is in phase with the arrival rate, this implies that the actual offered load lags the PSA

---

6 Notice that the actual offered load curve is intersected by the PSA offered load at its peak value. Empirical findings hereon were reported by Green and Kolesar (1997) and proved to be valid for the infinite server $M/G/\infty$ queue in Eick et al. (1993a).
too, resulting in a decreased effectiveness as service times increase. This implies that taking this time lag into account is particularly useful when service times are long.

Evidence supporting this intuition was given in Eick et al. (1993a), who derive a quadratic approximation of the infinite server offered load\(^7\). Besides a service time random variable \(S\), this

\(^7\) The infinite server offered load can be described as the time-varying expected number of customers in system if infinitely many servers would be available. In an infinite server system, this equals the expected number of busy servers.
approximation utilizes the concept of a service time stationary-excess random variable \( S_e(t) \), with cumulative distribution function
\[
P(S_e \leq x) = \frac{1}{E[S]} \int_0^x P(S \leq u) \, du
\] (1)

The quadratic approximation is then represented as
\[
m_{\infty}^{\text{QUAD}}(t) = \lambda(t - E[S_e]) \, E[S] + \frac{\lambda''(t)}{2} \text{Var}[S_e] \, E[S]
\] (2)

Which approximately equals the PSA offered load \( \lambda(t) \, E[S] \), although a time lag in the arrival rate as well as a space shift are present, defined as
\[
\text{Time lag: } -E[S_e]
\] (3)

\[
\text{Space shift: } \frac{\lambda''(t)}{2} \text{Var}[S_e] \, E[S]
\] (4)

Thus, the quadratic approximation indicates that the PSA is approximately correct if this time lag and space shift are negligible. The presence of a time lag again can be visualized: the actual offered load lags both the arrival rate and the PSA offered load (the latter two are in phase). This observation supports the use of a lagged variant of the PSA (denoted \( \text{LagPSA} \)), which would be represented graphically by shifting the PSA curve in Figure 3 and Figure 4 to the right. So, if one would be able to determine an appropriate value for this lag (denote it as \( L \)), then a better approximation entails the application of the PSA logic with arrival rate \( \lambda(t - L) \) for all \( t \), instead of \( \lambda(t) \). The lag is commonly approximated by the expected service time \( E[S] \). Theoretical background for this choice can be found in Eick et al. (1993a), where it is shown that this provides good results if the arrival rate is approximately quadratic before time \( t \) and the squared coefficient of variation lies around 1.

\[
\begin{align*}
\text{Lag PSA:} & \text{ apply a M/G/s+G model on each time instant } t, \text{ with arrival rate } \\
& \lambda_{\text{LagPSA}} \equiv \lambda(t - E[S])
\end{align*}
\]

\[
\begin{align*}
\text{Lag SIPP:} & \text{ apply one M/G/s+G model per interval } i \text{ with arrival rate } \\
& \lambda_{\text{LagSIPP}} \equiv \frac{1}{x_2-x_1} \int_{x_1}^{x_2} \lambda(t - E[S]) \, dt \text{ where interval } i \text{ starts at } x_1 \text{ and ends at } x_2
\end{align*}
\]

Similarly, a time lag can be added when applying the SIPP or SIPP Max method (Green et al. (2001), Green et al. (2003)). In the \( \text{Lag SIPP} \) approach, performance in interval \([x_1, x_2]\\n\) is approximated by a stationary model with arrival rate \( \frac{1}{x_2-x_1} \int_{x_1}^{x_2} \lambda(t - E[S]) \, dt \), whereas \( \text{Lag SIPP Max} \) approach, the arrival rate in the interval would be the maximum of the arrival rate function over interval \([x_1 - E[S], x_2 - E[S]]\\n\). In Green et al. (2003), an evaluation of staffing levels generation by these methods indicates that adding a lag always leads to better results. Roughly speaking, they suggest to use Lag SIPP when staffing intervals are short and the amplitude of the arrival rate is rather small. In all other situations, Lag SIPP Max is more appropriate.

\( f) \) Infinite server (IS) and modified offered load (MOL) approximations

If service times are long and the time lag and space shifts can no longer be neglected, the need arises for a better method to translate nonstationarity into input parameters for the stationary model. This is where the Modified Offered Load (MOL) approach comes in: based on convenient closed-form results for infinite server queues, a modified arrival rate function is derived which proves to be a better
approximation of the offered load in the nonstationary system. Similar to the previous methods discussed in this section, this modified arrival rate is then again used in a series of stationary models.

Instead of using MOL, one could use an Infinite Server (IS) approximation too. The difference between IS approximations and MOL approximations is not always stated clearly in the related literature, yet it is important to differentiate between both methods. In the IS approximation on the one hand, the nonstationary M(t)/G/s(t)+G model is approximated by a nonstationary M(t)/G/∞ model, which can be solved by numerical integration of ODE’s or by means of convenient analytical expressions for the infinite server offered load. The number in system is then approximated by the infinite server number in system, or equivalently, the number of busy servers that would prevail if infinitely many servers would be available. In the MOL approach on the other hand, the analytically tractable results on infinite server queues are used to obtain an adapted arrival rate that leads to better results when plugged into a stationary model. In a strict sense, IS approximations do not fall in the category of stationary approximations, but because IS and MOL approximations are frequently discussed simultaneously in the literature, some results concerning the IS approach will be given here as well.

The MOL concept was first introduced in Jagerman (1975), who applied MOL for time-dependent multiserver Markovian Erlang loss models and further explored in Massey and Whitt (1994), where MOL was analysed from a more mathematical perspective and error bounds were provided. In Massey and Whitt (1997), the accuracy of MOL for predicting peak congestion in M(t)/M/s models is examined. In the literature, the main focus lies on Markovian systems, aiming at the realization of a certain delay probability target. Nonetheless, the MOL approach should be extendable to both more general M(t)/G/s(t)+G models (Whitt, 2007) and other performance measures (results can be found Liu and Whitt (2009) and Feldman et al. (2008), which make this a promising approach for application in the context of an emergency department.

The idea behind the MOL approach is fairly simple: use a stationary model on each moment in time (similar to the PSA), but replace the nonstationary arrival rate function by the product of the service rate and the infinite server offered load function prevailing at that time (denoted by \( m_{\infty}(t) \)). The latter represents the number of servers that would be used on each moment in time if infinitely many servers would be available. The convenience of this approach lies in the fact that the IS-model is analytically tractable: closed form expressions for the infinite-server offered load are available. Eick et al. (1993a) prove that number of customers in an M(t)/G/∞ queue follows a Poisson distribution, with mean

\[
m_{\infty}(t) = \int_{-\infty}^{t} \left( 1 - G_u(t - u) \right) \lambda(u) \, du
\]

With \( G_u(t) = P(S(u) \leq t) \) the cumulative distribution function related to the service process service, and \( S(u) \) the service time of an arrival at time \( u \). In the case of a Markovian service process, simply solving the ordinary differential equations to obtain \( m_{\infty}(t) \) is an option, whereas under phase-type service time distributions, a set of differential equations can be solved as is done in Davis et al. (1995) and Massey and Whitt (1994). However, closed-form expressions for \( m_{\infty}(t) \) exist (see Eick et al. (1993b) for the case of sinusoidal arrival rates) and several efficient algorithms to determine \( m_{\infty}(t) \) are provided in Jennings et al. (1996), who also evaluate the accuracy of the IS approximation for M(t)/M/s(t) models.

---

8 Loss models can be described as systems where no waiting room is available and customers leave (instead of entering a queue) if no idle servers are available upon arrival (Gross 2008).
Jennings et al. (1996) state that the MOL approximation results in more accurate performance than the IS approximation, although in their opinion, the improvements have only little influence on the resulting staffing levels. Green et al. (2007) experienced that MOL becomes less effective if the system is overloaded, because of instability in the corresponding stationary models. Moreover, the MOL approximation performs well if the number of servers is sufficiently high (Ingolfsson et al., 2007) because then, the infinite server approximation proves to be accurate. The error bounds on the MOL approximation provided in Massey and Whitt (1994) lead to the conclusion that MOL is asymptotically correct as the arrival rate changes more slowly and thus its derivative approaches zero. In contrast, examples given in Jennings et al. (1996), show that the IS approximation should work well for both slowly and rapidly changing arrival rates (contrary to the PSA approach, which is in general more suitable for slowly varying arrival rates). As the number of servers declines, the MOL approximation becomes less accurate because of a lower resemblance to the infinite server system (Massey and Whitt, 1997). However, MOL still performs surprisingly well in small systems because the IS approximation remains fairly accurate even when system size is small (Jennings et al., 1996). Also, IS seems more advisable for small target delay probability, although large delay probabilities do not substantially worsen performance (Jennings et al., 1996). In comparison with other methods, Ingolfsson et al. (2007) report that both perform better than Lag SIPP (at the expense of somewhat higher computation times), although IS is less accurate than MOL.

Recently, Liu and Whitt (2009) further explored the possibilities of this MOL approximation, resulting in two new approximation methods for M(t)/G/s(t)+G queues. Here, instead of targeting a particular delay probability, other performance measures are considered, namely the abandonment probability and expected waiting time.

When an infinite server approximation is used, the effect of abandonments is neglected. To account for this, the delayed infinite-server offered load (DIS-OL) model is proposed. Two infinite server models are applied, the first of which represents a virtual waiting room where all patients are delayed for a fixed amount of time, depending on the desired expected waiting time or abandonment probability. After departing from this virtual space, all non-abandoned customers enter a second infinite server system, that will provide the infinite server offered load to be used for making staffing choices, similarly to the regular MOL approach.

Comparing the DIS-OL and MOL, it appears that DIS-OL succeeds in keeping abandonment probability stable over time, independent of the load, whereas in MOL the delay probability was only stabilised under lighter loads (or for small delay probabilities). Moreover, results indicate that DIS-MOL works well even if no Markovian assumptions are made, although results depend on the system size (as an indication, arrival rates of approximately 100 customers are mentioned). In addition, the authors address issues such as discretization of the staffing level function (because the model returns continuous values), the exhaustiveness of the service process and accounting for staffing intervals.

g) Effective arrival rate approximation

In the MOL approach, arrival rates are altered such that the time-varying system characteristics are approximated more accurately. In Thompson (1993), a similar approach is used, but now a different calculation method is used to specify the adapted arrival rate that is to be used in a stationary model. Waiting times are assumed to be deterministic and are approximated by the expected waiting time (denoted by $W$); service times are assumed to be equal to $\frac{1}{\mu}$. The arrival rate $\lambda(t)$ is then determined as a moving average over the interval $[t-W-\frac{1}{\mu}, t-W]$, as this interval will determine the number of customers in service at time $t$. Results on M(t)/M/s(t) models by Ingolfsson et al. (2007) indicated that this approach is similar to MOL in terms of accuracy, but systematically faster (although differences are rather small).
Effective arrival rate approximation: apply a M/G/s+G model on each time instant $t$, with arrival rate $\lambda_{effective\ ArrRate}(t) \equiv \mu \int_{s-W}^{s} \lambda(t)dt$

### 3.1.2 Determining staffing levels using a stationary approximation

Once the nonstationary model is translated into a series of related stationary models and the outcome of those models is obtained, staffing levels need to be chosen such that the performance target is satisfied (using as few servers as possible).

A straightforward yet computationally expensive way to address this problem is to calculate stationary outcomes for several staffing levels in each stationary model, after which the smallest staffing level that meets the constraint is chosen. This logic is applied in Green et al. (2001), where SIPP staffing levels are determined by repeatedly assessing the Erlang delay equation (Gross et al., 2008) for each interval and then choosing the smallest staffing level that meets the delay probability target. One important drawback lies in the calculation of steady state performance measures: depending on the system characteristics, closed form expressions for the stationary performance are not always available and therefore “applying a stationary model” is not always as straightforward as one might expect. This particularly holds for the difficult M/G/s+G model, which we focus on. Consequently, approximations are often necessary for obtaining the steady state performance measures that are needed to determine appropriate staffing requirements. Research has been done on approximating performance measures of stationary models. In Table 3, references to the relevant literature are given, along with a short description. Reported performance measures include the delay probability (given that the customer is served), abandonment probability (given that the wait exceeds a target), probability that the waiting time exceeds a target, expected waiting time (given that the customer is served or given that the wait exceeds a target).

<table>
<thead>
<tr>
<th>$M/M/s+M$</th>
<th>Garnett et al. (2002)</th>
<th>Intended for large systems</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mandelbaum and Zeltyn (2005)</td>
<td>Performance measures for QD, QED, and ED regimes. QED approximations are suitable for small system sizes than ED approximations.</td>
</tr>
<tr>
<td>$M/M/s+G$</td>
<td>Zeltyn and Mandelbaum (2005)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Whitt (2005)</td>
<td></td>
</tr>
<tr>
<td>$M/GI/s+GI$</td>
<td>Iravani and Balcioglu (2008)</td>
<td>In general outperforms Whitt (2005), especially in heavily loaded systems</td>
</tr>
<tr>
<td>$G/GI/s + GI$</td>
<td>Whitt (2006)</td>
<td>2 fluid models, intended for heavily loaded or overloaded systems</td>
</tr>
<tr>
<td>$G/GI/n/m$</td>
<td>Whitt (2004)</td>
<td>Diffusion approximations, intended for large number of servers</td>
</tr>
</tbody>
</table>

Table 3 Overview of methods for the approximation of steady state performance measures

Sometimes staffing levels can be determined without explicitly calculating steady state performance measures. One rule of thumb that has received a fair amount of attention because of its simplicity and robustness is the square-root staffing rule (SRS), often also referred to as square root safety staffing (general background and applicability of SRS is provided in Whitt (1992) and Koole and Mandelbaum (2002)). Here, staffing levels $s(t)$ are chosen to be equal to the offered load $\mu$ (i.e. the expected...
number of customers in the system), augmented with an amount of safety capacity, proportional to the square root of the offered load (cfr. expression (6)). The SRS rule has proven to be remarkably accurate in different settings (e.g. in the PSA approach or IS approximation, cfr. Jennings et al. (1996)).

\[
\text{SRS-rule: } s = \left[ m + \beta \sqrt{m} \right] 
\] (6)

Again, infinite server approximations prove to be helpful to improve general understanding of this rule (for extensive theoretical background on the SRS rule we refer to Whitt (1992)). Here, we briefly discuss how the SRS-rule can be applied in an M(t)/G/s(t)+G system. First, the M(t)/G/s(t)+G system is approximated by an infinite server M(t)/G/∞ model. In this latter model, the number of busy servers at time \( t \) follows a Poisson distribution with mean equal to the offered load \( m_{\infty}(t) \), if the number in system at time 0 is Poisson as well (Eick et al. (1993a), Eick et al. (1993b)). Although this distribution is not valid exactly in the M(t)/G/s(t)+G system, it tends to be more suitable as the number of servers increases (Whitt, 1992). Or, the number in system in the M(t)/G/s(t)+G (denoted by \( N(t) \)) is approximated by the infinite server number in system (represented as \( N_{\infty}(t) \)):

\[
N(t) \approx N_{\infty}(t) 
\] (7)

This Poisson distribution can in turn be approximated by a normal distribution with both mean and variance equal to the offered load, under the condition that the offered load is not too small and targeted quality of service is high (Green et al., 2007).

\[
N_{\infty}(t) \xrightarrow{\text{approx.}} \text{NORMAL} \left( m_{\infty}(t), m_{\infty}(t) \right) 
\] (8)

Where \( \text{NORMAL} \left( \mu, \sigma^2 \right) \) represents a normal distribution with mean \( \mu \) and variance \( \sigma^2 \). Now the targeted delay probability \( \alpha \) can be written as

\[
\alpha \equiv P(N_{\infty}(t) \geq s(t)) \approx 1 - P \left( \frac{N_{\infty}(t) - m_{\infty}(t)}{\sqrt{m_{\infty}(t)}} < \frac{s(t) - m_{\infty}(t)}{\sqrt{m_{\infty}(t)}} \right) 
\] (9)

\[
\approx 1 - \Phi \left( \frac{s(t) - m_{\infty}(t)}{\sqrt{m_{\infty}(t)}} \right) 
\] (10)

Where \( \Phi \) denotes the standard normal cdf. Which results in an expression that links the target delay probability to a quality of service parameter \( \beta \):

\[
\beta \equiv \frac{s(t) - m_{\infty}(t)}{\sqrt{m_{\infty}(t)}} 
\] (11)

This leads to the square root staffing formula for the M(t)/G/s(t)+G model, a rule of thumb that can be applied in order to obtain staffing levels that meet the targeted delay probability.

\[
s(t) = \left[ m_{\infty}(t) + \beta \sqrt{m_{\infty}(t)} \right] 
\] (12)

The factor \( \beta \) is chosen depending on the performance target and evidently, the extent to which SRS will lead to overstaffing or understaffing is directly related to the specification of this parameter. However, the normal approximation might not always be suitable and thus other ways were explored to link the desired performance to a parameter \( \beta \) that can be plugged into the SRS rule. If performance is

\footnote{In one particular case, the distribution of number in system in M(t)/G/s(t)+G is identical to that of the infinite server model, namely if the specific condition that the abandonment rate is equal to the service rate (Whitt, 2007).}
defined in terms of the delay probability, two well-known expressions can be used, although additional 
Markovian assumptions might be necessary. Inverting the Halffin-Whitt delay function can be used to 
obtain the appropriate $\beta$ in a stationary M/M/s model (Halfin and Whitt, 1981); the Garnett delay 
function is meant for Markovian M/M/s+M models with abandonments (Garnett et al., 2002) and in 
Zeltyn and Mandelbaum (2005) the M/M/s+G model is addressed. These functions are derived for 
many-server heavy-traffic regimes, but should work for small system sizes as well (although accuracy 
increases as the number of servers and the arrival rates approach infinity); examples supporting this are 
provided in Jennings et al. (1996). If other performance targets are used, these delay functions can be 
used after reformulating the performance measure of interest in terms of the delay probability. For 
instance in M/M/s models, waiting time percentiles can be expressed in terms of the delay probability 
in the following way, because the conditional waiting time given that all servers are busy is exponential 
(Whitt, 1992):

$$\tilde{a} \equiv P(W \leq x)$$

$$= P(W > 0) \frac{P(W \leq x | W > 0)}{P(W > 0)} = P(W > 0) \frac{e^{-\frac{x}{1+\beta}}}{e^{-\frac{1}{1+\beta}}}$$

The symbols $W$ and $x$ represent the waiting time and maximal allowed waiting time respectively and $\tilde{a}$
stands for the targeted probability of a waiting time smaller than the maximal allowed wait (the tilde is 
used to emphasize the difference with previous targets which were related to the delay probability 
instead of waiting time service levels). This enables the use of the Halfin-Whitt delay function, as the 
delay probability $P(W > 0)$ is linked to $\beta$ through this expression. This approach can be extended to 
other performance measures, e.g. Borst et al. (2004) suggest approximate $\beta$-values that are related to 
labor and waiting costs. While approximate values of $\beta$ are often available, determining exact 
expressions remains an important challenge. If available delay functions or approximations fall short, 
an alternative option is to determine the appropriate beta through simulation (Liu and Whitt, 2009).

### 3.2 NONSTATIONARY MODELS

In the previous section, the use of stationary models over smaller intervals was proposed as a fairly 
simple and computationally inexpensive approach to approximate nonstationary models. When using 
stationary approximations, some implicit assumptions are made which might not always be valid 
(Green et al., 2001). First of all, a stationary model assumes that arrival rate is constant over the time 
period. In a nonstationary model this obviously is not the case, but as described in the previous 
sections, some stationary approximations succeed quite well in translating nonstationary characteristics 
in constant arrival rates that are then used in stationary models. Also, successive staffing intervals are 
considered to be independent and it is assumed that steady state is reached within each staffing interval, 
so transient effects are neglected. These conditions are not always met and therefore might cause 
stationary approximations to be ineffective.

This section will describe some models that attempt to model the nonstationarity explicitly. These 
methods are often less straightforward and require more computational effort Green et al. (1991) but 
may be more appropriate in situations where stationary approximations fall short.

#### 3.2.1 Performance evaluation in nonstationary models

To find out whether a particular staffing function is appropriate, its performance needs to be assessed. 
In a nonstationary system, performance evaluation tends to be more complicated than in its stationary 
counterpart. Moreover, some performance measures, such as delay probability, can be computed quite 
easily, whereas others are more difficult to compute or approximate. Although we are mainly interested 
in service levels (w.r.t. waiting times) that correspond to a specific staffing function, performance 
evaluation methods for several well-known performance measures are discussed in this section.
**a. Simulation**

Simulation is a technique in which the evolution of a system’s performance over time is imitated. Often, simulation software packages are available and can be used to this purpose. In discrete event simulation, the system is modeled in a noncontinuous way, only considering the time epochs at which a change in the system’s state occurs. Discrete-event simulation is used frequently in healthcare (see Jacobson et al. (2006) and Jun et al. (1999) for an extensive overview of discrete-event simulation in healthcare settings).

The main advantage of simulation is its applicability to general problem settings (Feldman et al., 2008): in contrast to many analytic models, making Markovian assumptions is not mandatory. Moreover, simulation enables analysing complex systems, for which no analytic results are available. Fletcher et al. (2007a) point out that simulation facilitates model validation through the possibility to visualise system dynamics. This seems particularly useful in a healthcare setting, as visualization of the simulation model allows those who are unfamiliar with simulation tools (e.g. healthcare personnel) to check whether the simulation model is implemented correctly.

Drawbacks include the need for multiple replications to obtain accurate results, which may result in considerable computation times. Consequently, simulation seems less suitable to find optimal staffing levels, as only a limited number of alternatives can be evaluated, due to this high computational effort (Green et al., 2007). On the other hand, the simulation-based ISA approach proposed by Feldman et al. (2008) proves that simulation can be used to determine staffing levels, with an acceptable computational effort (ISA will be discussed in detail in Section 3.2.2). Moreover, simulation models are often context-specific (see Evans et al. (1996), Garcia et al. (1995), Hung et al. (2007), McGuire (1994), Takakuwa and Shiozaki (2004) for applications of simulation in ED’s) although efforts are made to develop generic simulation models in hospital environments (Fletcher et al. (2007a), Fletcher et al. (2007b), Pitt (1997), Sinrech and Marmor (2004)).

**h) Numerical integration of ODE’s**

For the M(t)/M/s(t) system, performance can be evaluated by numerically integrating the following ordinary differential equations (Gross et al., 2008):

\[
p'_{n,t} = -\lambda(t)p_{n,t} + \mu_{n+1,t} \quad 1 \leq n < s(t) \quad (15)
\]

\[
p'_{n,t} = \lambda(t)p_{n-1,t} - (\lambda(t) + n\mu)p_{n,t} + (n + 1)\mu_{n+1,t} \quad 1 \leq n < s(t) \quad (16)
\]

Where \( p_n(t) \) represents the probability of \( n \) customers in system at time \( t \). Let \( p(t) \) represent the vector containing all \( p_n(t) \) values, then this can be expressed equivalently as

\[
p'(t) = p(t)Q(t)
\]  

(17)

Where \( Q(t) \) represents the **generator matrix** of a continuous-time Markov chain (CTMC) with entries

\[
q_{ii}(t) = \begin{cases} 
-\lambda(t) + n\mu & 1 \leq n < s(t) \\
-\lambda(t) + s(t)\mu & n \geq s(t)
\end{cases}
\]  

(18)

\[
q_{i,i+1}(t) = \lambda(t) 
\]  

(19)

\[
q_{i-1,i}(t) = \begin{cases} n\mu & 1 \leq n < s(t) \\
s(t)\mu & n \geq s(t)
\end{cases}
\]  

(20)
In an actual implementation, one can obviously not solve for an infinite number of ODE’s, so only the first $K$ differential equations are considered (with $K$ large enough to keep approximation error small). This can be done by an ODE-solver such as the Runge-Kutta ODE solver from the Matlab ODE Suite (Shampine and Reichelt, 1997). Ingolfsson et al. (2007) use this tool for each separate staffing interval. As an important addition, the exhaustiveness of the service process is accounted for by applying a matrix transformation on the resulting probability vector. The computational effect of this addition appears to be modest, compared to the effort needed to solve the ODE’s numerically (Ingolfsson, 2005). Intuitively this transformation can be explained as follows: at the end of each staffing interval, the number of servers might change and therefore it is possible that some servers are scheduled to leave. However, in an exhaustive policy it is plausible that a server works past its scheduled time because customer service is never interrupted. When a server is scheduled to leave, the customer it processes no longer has an influence on the waiting time, as no new customers are accepted by this server. The matrix transformation then reflects the effect that some customers are no longer relevant in calculations and will be neglected further on. The resulting probability vector (after the matrix transformation has been applied) is then used as a starting point for solving the ODE’s in the next staffing interval. This way, the system is modeled as a mixed discrete-continuous time Markov chain: a continuous time Markov chain is used, except when the number of scheduled servers decreases (then an instantaneous transition occurs, as in a discrete-time Markov chain).

Performance measures such as the delay probability and the average number in queue can be derived easily from the state probabilities $p(t)$. For service levels (in our case the probability that the wait exceeds a maximum), this is less straightforward. Ingolfsson et al. (2007) provide insights on how the resulting probability vector can be translated to service level performance measures (a similar approach was suggested independently in Green and Soares (2007) although there a pre-emptive service process is assumed). If a customer arrives when all servers are busy, then he will have to wait for a number of service completions before entering service (this number is equal to the queue length, which is denoted as $i$, augmented with 1). Consequently, the probability that the virtual wait $W(t)$ at time $t$ exceeds some maximal value $\tau$, is equal to the probability that there are at most $i$ service completions over the interval $(t, t + \tau]$, i.e. not enough for the customer to start service during interval $(t, t + \tau]$. If the service process is assumed to be Poisson distributed, this probability equals the sum of a number of terms, which are Poisson distributed with mean $\mu = \mu s(t)$, where $\mu$ represents the service time and $s(t)$ the available number of servers at time $t$ (under the assumption that $s(t)$ does not change during interval $(t, t + \tau]$). In this case, the service level $SL(t)$ at time $t$ can be calculated as follows:

$$SL(t) = P(W(t) \leq \tau)$$

$$= 1 - P(W(t) > \tau)$$

$$= 1 - \sum_{i=0}^{\infty} p_{s(t)+i}(t) \frac{\mu e^{-\mu}}{\mu}$$

Where $p_{s(t)+i}(t)$ represents the probability of $s(t) + i$ customers in system upon arrival at time $t$. In a similar fashion, the service level can be calculated if the number of servers is scheduled to decrease or increase over interval $(t, t + \tau]$, or even when the number of servers changes more than once during the interval (details on this are provided in Ingolfsson (2005)).

An extension of this approach including Markovian abandonments, is described in Ingolfsson (2005). The only change occurs in the continuous time Markov chain, where the departure rate from a specific state is augmented with the product of the abandonment rate and the number in queue. However, this implies that the number of departures over an interval is no longer Poisson distributed and as a result the service level can no longer be calculated as described before. In this case, the authors suggest an alternative approach to determine the service level, focusing on the queue length process. Let $\phi_n(t)$
denote the probability of \( n \) customers in queue at time \( t \), then the link between the number in queue and number in system can be described as follows:

\[
\phi_0(t) = \sum_{m=0}^{(t)} p_m(t) \tag{24}
\]

\[
\phi_n(t) = p_{(t)+n}(t) \quad \text{for } n = 1, 2, \ldots \tag{25}
\]

The service level for a customer arriving at time \( t \) (i.e. the probability of a waiting time larger than \( \tau \)) then equals the probability that the queue is not empty at time \( t + \tau \) (i.e. the moment at which the customer has waited for the maximal allowed time \( \tau \)). Consequently, the service level at time \( t \) can be expressed as

\[
SL(t) = 1 - \phi_0(t + \tau) \tag{26}
\]

The state probabilities concerning the queue length can be computed by numerically solving ODE’s, but the uniformization approach (to be described in the next subsection) can be used as well.

From an ED viewpoint it is appealing that exhaustive service policies, abandonments and service level performance targets can be accounted for. But this method relies heavily on the assumption of Markovian service and abandonment processes (which are unlikely to be valid in an ED context) and departing from these assumptions severely complicates the method. Moreover, although exact results can be obtained, the computational effort of numerically solving ODE’s might not be acceptable. Computation times as well as accuracy depend heavily on how frequently performance is evaluated; a common choice is to opt for an interval length of 10% of the average service time (cfr. Feldman et al. (2008) and Jennings et al. (1996)), although 1 or 5 minute intervals are common as well (cfr. Green et al. (2003) and Green and Kolesar (1995) respectively). As a result, more efficient ways to approximate system performance are often preferred. Nevertheless, solving ODE’s numerically is often used for comparison purposes, to check whether a staffing method that relies on performance approximations indeed generates the desired performance (see for instance Green and Kolesar (1995), Green and Kolesar (1991), Green et al. (2003) and Green and Kolesar (1997)).

An efficient way to solve ODE’s is presented in Davis et al. (1995). There, the ODE’s in expression (17) are solved by approximating the continuous-time Markov chain (CTMC) by a discrete-time Markov chain (DTMC) in several intervals. Therefore, the CTMC generator matrix \( \mathbf{Q}(t) \) must be translated to a discrete-time transition matrix \( \mathbf{P}(k) \), where \( k \) is an index that indicates the interval. This transition matrix has the following entries:

\[
\mathbf{P}_{ij}(k) = \begin{cases} 
    hQ_{ij}(kh) & i \neq j \\
    1 - \sum_{m \neq i} \mathbf{P}_{ij}(k) & i = j 
\end{cases} \tag{27}
\]

Where \( Q_{ij}(t) \) denotes the element \((i,j)\) in \( \mathbf{Q}(t) \), \( h \) denotes the interval size (so \( kh \) represents the time instant corresponding to the \( k \)th interval). Then, the probability vector \( \mathbf{p}(kh) \) can be approximated by its discrete time equivalent (represented as \( \mathbf{\bar{p}}(kh) \)), which in turn can be obtained by recursively solving the following equation (where \( \mathbf{\bar{p}}(0) = \mathbf{p}(0) \)):

\[
\mathbf{\bar{p}}(jh) = \mathbf{\bar{p}}((j-1)h)\mathbf{p}(j-1) \quad j \geq 1 \tag{28}
\]

The efficiency of this approach is mainly due to the structure of the \( \mathbf{P}(k) \): expression (28) can be evaluated by only considering non-zero entries (instead of working with the matrix as a whole), which leads to a highly efficient method because \( \mathbf{P}(k) \) contains many zero-entries. This approach is used frequently in practice (e.g. in Feldman et al. (2008), Massey and Whitt (1996) and Massey and Whitt (1997)).
i) Discrete time modeling

A discrete-time modeling approach can be used to obtain the state probability vector \( p(t) \) in an M(t)/G/s model, based on a discretization of the service process (see Chassioti and Worthington (2004), Brahimi (1990), Brahimi and Worthington (1991) and Wall and Worthington (1994)).

The duration of the general service process is approximated by a discrete process (in Brahimi and Worthington (1991) and Brahimi (1990) this is done by two-moment matching) which can be interpreted as follows: suppose that \( k \) denotes the length a unit interval defining the discretization, then a service time will be expressed in terms of the number of intervals with length \( k \) that are needed to complete service (instead of the exact time duration). This discretization allows keeping track of the remaining service time of each customer, at a specific point in time. Moreover, the probability of any number of arrivals occurring during interval \( k \) can be calculated as well. As a result the system can be described completely: given the state probabilities at time \( p(t) \) and using the expected number of arrivals and service completions, the state probability at time \( p(t+1) \) can be determined. One advantage of this approach is its applicability to overloaded systems (Chassioti and Worthington, 2004). To our knowledge, this approach has not yet been implemented to M(t)/G/s(t) models, but there is no evidence indicating that the method cannot be generalized to this model (Wall and Worthington, 1994).

j) Randomization/uniformization method

The randomization or uniformization approach is an efficient approximation technique that is often used to solve the set of differential equations \( p'(t) = p(t)Q \). This method was introduced by Jensen (1953), and further explored in Grassmann (1977) and Gross and Miller (1984). This method requires arrival rates to be piecewise constant over small intervals (so-called calculation intervals), so constant arrival rates will have to be approximated by averaging over this interval (Ingolfsson et al., 2007). Moreover, the assumption of a finite state space is made, i.e. the number in system at time \( t \) is assumed to be at most \( K \). This should not pose problems provided that \( K \) is chosen large enough such that the probability of the number in system exceeding \( K \) is negligible (Gross and Miller, 1984).

The method relies on linking the continuous time Markov chain to a discrete time equivalent. We refer to Grassmann (1977) and Gross and Miller (1984) for an in-depth description of this approach but without going too much in detail, the main idea behind the randomization approach is as follows: suppose \( \{X(t); t \geq 0\} \) is a continuous Markov process with generator \( Q \) (for instance representing the number of customers in system. Let \( A \) denote the largest absolute value among all diagonal elements of \( Q \) and let \( \{N(t); t \geq 0\} \) represent a counting process (characterized by rate \( A \)), indicating the number of occurrences in time interval \([0,t]\). Note that for this counting process, events are generated according to a Poisson distribution and therefore \( P(N_t = n) = e^{-At} \frac{(At)^n}{n!} \). Then there exists a discrete time Markov process \( \{Y_n; n = 0,1,2,\ldots\} \) with transition matrix \( P = \frac{Q}{A} + I \) such that the probabilistic distributions of \( \{X(t); t \geq 0\} \) and \( \{Y_n; t \geq 0\} \) are identical. Using this feature, the probability of \( s \) customers in system at time \( t \), represented by \( p_s(t) \), can be expressed as

\[
p_s(t) = P(X(t) = s) = P(Y_{N(t)} = s) = \sum_{n=0}^{\infty} P(Y_n = s \mid N(t) = n)P(N(t) = n) = \sum_{n=0}^{\infty} P(Y_n = s) e^{-At} \frac{(At)^n}{n!}
\]

Which results in the following expression for \( p(t) = (p_0(t), p_1(t), p_2(t), \ldots) \):
This expression can be used in each calculation interval $c$ to determine $p(c)$ based on the probability vector that resulted from the previous interval (corresponding to the $p(0)$ used in the expression). Moreover, this approach is readily extendable to include for an exhaustive service policy, in a similar way as was suggested in Ingolfsson (2005) for solving ODE's.

In Ingolfsson et al. (2007), a comparison of the randomization method with exact results (obtained by numerically solving ODE’s using a Runge-Kutta ODE solver) indicate that randomization is usually very accurate and requires substantially less computational effort than the exact method (Reibman and Trivedi, 1988). However, as in the exact method, computation times still tend to be dependent on the model structure, e.g. a larger state space might be necessary as the system size increases. This is why randomization is often used in practice to obtain an highly accurate performance approximation, like for instance in Ingolfsson et al. (2010), where the randomization approach is applied as a subroutine to evaluate performance in a staffing algorithm. On the downside, Reibman and Trivedi (1988) mention that small calculation intervals are usually necessary in order to obtain acceptable accuracy. On the other hand, the randomization approach requires longer computation times than MOL, IS, Lag SIPP, and therefore should only be used when accuracy is important and the computation times pose little constraints.

In Ingolfsson et al. (2007), the performance in an $M(t)/M/s(t)$ system is determined by repeatedly applying the randomization approach for stationary systems to consecutive small intervals. Massey and Whitt (1998) provide a direct generalization of the randomization-principle to nonstationary systems, by explicitly accounting for the fact that matrix the CTMC generator matrix $Q(t)$ is not constant over time. The variable $\lambda$ is now determined as the maximum of diagonal elements over all $Q(t)$ matrices and the transition matrices $P(t) = \frac{Q(t)}{\lambda} + I$ account the time-variation in $Q(t)$. Instead of using $P^n$ in expression (35), a more complex function of the all $P(t)$ matrices is used. For more details, we refer to Massey and Whitt (1998).

**k) Other performance evaluation methods**

**Closure approximations**

A second way to calculate system performance more efficiently consists in applying a closure approximation. The general principle was suggested in Rothkopf and Oren (1979) for $M(t)/M/s$ queues, and entails the use of merely two differential equations: one for the expected number in system and one for the corresponding variance. Because these cannot be solved without knowledge of the transient state probability vector $p(t)$, the authors use the assumption that $p(t)$ follows a negative binomial distribution. Unfortunately, this assumption lead to inferior results for multiserver queueing systems, compared to single server systems. In this respect, the approach suggested by Clark (1981) proved to be more accurate. There, a set of five differential equations is used, representing the expected value and variance of the number in system, the number of busy servers and the probability of an empty queue. Taaffe and Ong (1987) present a similar approach with some further generalizations (by allowing for a phase type arrival process, time-varying service and limited system capacity) and this proved to be more accurate and more efficient than both Rothkopf and Oren (1979) and Clark (1981). In this respect, based on the observation that accuracy stays acceptable even for the worst settings tested for, the authors state that the proposed approach should be sufficiently accurate in many applications. Although the idea of using only a small set of differential equations seems appealing, Ingolfsson et al. (2007) state that the closure approximation approach can be quite challenging from the point of view of implementation. The reason is that some additional parameters have to be determined before the
differential equations can be solved (which can be done by means of a regular ODE solver) and this can be quite burdensome. Moreover, experiments performed in Ingolfsson et al. (2007) indicate that besides longer computational times, closure approximations provide less accurate results than the MOL, IS, Lag SIPP, effective arrival rate approximation and randomization methods.

**Fluid approximations**

Fluid approximations can be used to evaluate performance in an M(t)/G/s(t)+G system. Whitt (2006) point out that fluid approximations are particularly useful to assess performance in overloaded systems, where many traditional methods fail (e.g. stationary approximations). Under those circumstances, even a small amount of abandonment can keep the system stable. For underloaded systems, fluid approximations often fail to capture system dynamics accurately, although in Liu and Whitt (2010b), a fluid-based approach is suggested that works for overloaded as well as underloaded systems (separate models are applied in both situations). However, a key assumption in fluid approximations is that capacity changes in a continuous way (the staffing function is not bound to integer values); an assumption that becomes more acceptable as the number of servers increases. Although Liu and Whitt (2010b) report reasonably accurate results for a system with 20 servers, this assumption of fluid scaling seems incompatible with the characteristics of an emergency department, where extremely small system sizes are prevalent and the discreteness of capacity is an essential characteristic of the system. For these reasons, fluid approximations do not fall within the scope of this paper and are not discussed in detail. For additional literature on the use of fluid approximations to assess performance in systems with nonstationary arrival patterns, we refer to Mandelbaum et al. (1998) for M(t)/M(t)/s(t) models, Mandelbaum et al. (1999) for M(t)/M(t)/s(t)+M(t) models, Whitt (2006) for G(t)/GI/s+GI models, Liu and Whitt (2010b) for the G(t)/GI/s(t)+GI model, and references therein.

### 3.2.2 Determining staffing levels in nonstationary models

#### a. Iterative staffing algorithm (ISA)

In Feldman et al. (2008), a promising simulation-based technique for determining staffing requirements in time-varying queues is proposed. As the name suggests, the **Iterative Staffing Algorithm (ISA)** will alter the staffing function repeatedly, until the desired performance is attained. For each considered staffing function, system performance is evaluated by means of simulation and subsequent staffing levels are updated based on the observed performance. This sequence of evaluating performance and updating staffing levels is called an iteration. The ISA is time-continuous and thus it does not aim at explicitly accounting for the length of the staffing intervals. Similarly to the PSA approach or numerically solving ODE’s however, system performance is evaluated almost continuously (i.e. at discrete points in time, over small intervals), so the effect of applying ISA to (longer) staffing intervals is unclear.

The planning horizon $T$ is divided into small intervals and that staffing changes are only allowed at the start of each interval $t$. Let $N_i^t$ denote the number of customers in the system at time $t$, in iteration $i$ of the algorithm. Suppose that $s_i^t$ represents the staffing level at time instant $t$, at the start of iteration $i$ and that $\alpha$ stands for the target delay probability. The ISA then proceeds as follows. Initially, all staffing levels are set equal to some large number. Subsequently, the system performance is simulated by performing a fixed number independent replications, which results in a distribution of the number of customers in system at each moment in time (no confidence intervals are considered). Then, staffing levels are improved such that at each time $t$, the staffing level corresponds to the smallest value for which the performance constraint (which focuses on the delay probability) is met. That is, after estimating the distribution of $N_i^t$ by performing a simulation based on $s_i^t$, the new staffing levels $s_i^{t+1}$ are chosen such that
This sequence of simulating performance and updating the staffing function stops when the staffing changes in subsequent iterations become sufficiently small for all time instants (that is, staffing levels differ by at most 1, for all t).

In Feldman et al. (2008), it is shown that for the M(t)/M/s(t)+M queue, the ISA converges. However, the speed of convergence (i.e. the number of iterations needed) depends on the ratio of the abandonment rate to the service rate ($\theta/\mu$). When service rate equals abandonment rate, the ISA needs only 2 iterations to converge. On the contrary, if the abandonment rate is small compared to the service rate and the system is characterized by a high traffic intensity, a large number of iterations might be required before convergence is obtained. Since this last situation is most likely to be valid in an ED context, the ISA might not be the most efficient way to determine staffing requirements.

In Feldman et al. (2008) the performance of ISA is illustrated by means of two examples: on the one hand a sinusoidal arrival pattern is examined, on the other hand a more realistic example is provided, using the arrival rate of a medium-sized call center. For both examples, the performance corresponding to the final ISA staffing levels was simulated, using 5000 independent replications. Some other performance measures, such as abandonment probabilities, average queue lengths and average waiting times, were evaluated as well. Performance measures (in each interval) were measured and averaged over all replications, so no confidence intervals w.r.t. these variables were considered. For the sinusoidal arrival rate example, ISA performs well, as all delay probabilities are close to the targeted value and more or less stable over time. Surprisingly, for higher targeted delay probabilities, several other performance measures were time-stable as well, even though these were not part of the objective. For lower targeted delay probabilities, only the delay probability displayed time-stable performance and other performance measures displayed a more outspoken time-dependency (again, this is not surprising, since these are not incorporated in the algorithm’s objective function). The same conclusions hold for the more realistic call center example, although larger fluctuations around the targeted delay probability were present at the beginning and end of the day, i.e. in periods with extremely low arrival rates. This decreased performance could be explained by the inherent discreteness of the system because when arrival rates are low, changing capacity by one unit has a larger impact on performance.

As both examples are characterized by relatively high average arrival rates (e.g. 100 customers per hour), the question rises whether the performance of ISA remains valid for smaller system sizes. Moreover, the assumption is made that service rates and abandonment rates are equal, and consequently the distribution of the number in system in the M(t)/M/s(t)+M is identical to the infinite server equivalent (Whitt, 2007). Nonetheless, ISA results should not be dependent on this assumption (but the results on the MOL-approximation reported in Feldman et al. (2008) are, as this approximation makes use of an IS-approximation).

The major advantage of the ISA lies in the use of simulation to evaluate system performance. As a result, the appropriateness of the staffing function generated by ISA is validated automatically (that is, under the assumption that the simulation model is adequate). Moreover, this method has potential to be applied to much more general settings, for which analytical results are no longer available.

The ISA as it is described by Feldman et al. (2008), aims at a stable delay probability throughout the day. However, as pointed out before, the delay probability appears not to be the most appropriate performance measure in an ED environment. However, using the delay probability is advantageous because of the clear link between what is measured in the simulation and the staffing levels needed to achieve the desired performance: once the distribution of the number in system is known, it is
straightforward to determine the smallest staffing level that meets the delay probability constraint, because this latter can be expressed as:

\[ \alpha \geq P(W(t) > 0) \quad t = 0, \ldots, T \]

\[ = P(N(t) > s(t)) \quad t = 0, \ldots, T \]

(37)

(38)

Where \( \alpha \) denotes the target delay probability, \( W(t) \) the waiting time of a customer arriving at time \( t \), \( N(t) \) the number of customers in system and \( s(t) \) the available capacity. Because of expression (38), the available capacity can be chosen such that probability that the number in system exceeds the available capacity (and thus the delay probability) is sufficiently small. This is the staffing level that would have resulted in a good solution, if the distribution on the number in system obtained from the simulation would have been exact (of course, when simulating a different staffing level function, these distributions will change).

Furthermore, an ED is commonly characterized by a rather small system size and hence the conventional stopping rule of the ISA might pose problems. The algorithm stops when the change in staffing requirements is at most 1 unit in all intervals and thus staffing changes of \( \pm 1 \) server are disregarded. However, in a small system such as an ED, the addition or removal of one server can lead to noticeable changes in system performance. These effects can be explained by the inherent discreteness of staffing levels, a feature that can no longer be disregarded when system size is small. Similarly, Feldman et al. (2008) report an example of a call center, where ISA does not succeed in achieving the desired performance in periods with relatively low arrival rates.

Moreover, the ISA does not explicitly deal with the length of the staffing intervals, i.e. the time period over which capacity remains constant. All examples used in Feldman et al. (2008) assume staffing intervals with a length of \( 0.1/\mu \) and although this decision might have a large impact on the algorithms performance, no explanation for this choice is provided. Intuitively, we would expect ISA to perform worse as the length of these intervals increases, in particular because in the simulation model, the number of customers in system is measured only once every interval, so increasing interval length obviously leads to a reduction in accuracy. An alternative is then to average the number in system over the interval, instead of assessing this quantity merely once every interval.

**b. Scheduling algorithms**

All methods discussed so far, result in minimal staffing requirements, i.e. the smallest number of servers needed in each staffing interval, to meet a chosen performance measure constraint. In reality however, additional constraints have to be taken into account, e.g. shift length constraints and scheduled breaks. Often this is dealt with by using a two phase approach (Henderson and Mason, 1998): in a first step, minimal staffing requirements are determined and in a second step, a roster or schedule (i.e. an assignment of personnel to shifts) is sought such that eventual staffing levels at each moment in time are at least the minimal staffing requirements\(^{10}\). However, a separate treatment of both phases is not always ideal. There can be several alternative minimal staffing requirement functions resulting in the same service level (e.g. adding 1 server in period \( t \) and removing one in the next period might not impact performance), but not all are equally suitable for obtaining a good roster. To avoid this, Henderson and Mason (1998) outline an approach which simultaneously addresses the determination of staffing requirements and rostering.

Henderson and Mason (1998) suggest iteratively using an IP to determine a roster, followed by simulation to assess performance. After each simulation, the IP is updated by adding additional

\(^{10}\) An example of such an approach is described in Centeno et al. (2003), where simulation is used to determine staffing requirements and subsequently a schedule is sought by means of an ILP.
constraints ("cuts") on the staffing requirement function, leading to a reduction of the solution space. This is called a cutting plane method (we refer to Kelley (1960) for general background on this method). In contrast to the two-phase approach, both the staffing requirement function and the roster are changed in an iteration. When solving the IP, a roster that minimizes cost is sought (this can be related to labor costs or employee satisfaction). Two sets of constraints are added: on the one hand, the scheduled number of servers (or employees) corresponding to the roster must exceed the minimal staffing requirement function at each point in time. On the other hand, constraints concerning the staffing requirement function (cuts added during previous iterations) must be satisfied. The performance of the resulting roster is then evaluated by means of simulation. Based on the outcome, additional constraints concerning the minimal staffing requirements are added to the IP and a new roster is determined. This process is iterated until convergence is achieved. The authors only outline the general idea behind this approach; no results are provided (although they predict a high computational effort because many IP’s need to be solved and many simulation need to be performed). In Atlason et al. (2004), this method is elaborated further and results for a small M(t)/M/s(t) example setting are analyzed, where the service level is used as a performance measure (i.e. the probability that the wait is lower than the maximal acceptable waiting time). After each simulation, constraints are added for those periods where the service level was unacceptable. The algorithm terminates when the service level constraint is satisfied in every period. The cutting plane method that is used in this algorithm relies on the assumption of concavity of service levels: the service level is assumed to increase if the number of servers increases and the marginal increase in service level becomes smaller as the number of added servers augments. Atlason et al. (2005) extend this approach to the less strict assumption on pseudo-concavity (a definition of the concept can be found in Atlason et al. (2005) by using different cutting plane method (namely the ACCPM-method; we refer to Gondzio et al. (1996) for details). More elaborate experiments are conducted, comparing the resulting roster with staffing requirements generated using several stationary approximation methods (SIPP, SIPP Max, SIPP Mix, Lag SIPP, Lag SIPP Max and Lag SIPP Mix, as discussed in Section 3.1.1). The method proposed in Atlason et al. (2005) outperformed the stationary approximations for almost all tested settings, because fewer servers were needed to meet the service level constraints. In contrast to the stationary approximations, the general applicability of this method is an advantage (no assumptions concerning the arrival or service rate are made), although considerable computation times form a major drawback. Ingolfsson et al. (2002) suggest a similar approach which does not rely on assumptions of concavity or pseudo-concavity. Instead of using simulation, the randomization method (Section 3.2.1) is used to evaluate the service level for a given schedule. In staffing intervals where the service level is unacceptable, the number of servers needed to obtain an acceptable service level is estimated. Based on this estimation, a constraint on the staffing requirement function is added to the IP and a new schedule is calculated. This approach is compared to a method where instead of the randomization approach, SIPP (cfr. Section 3.1.1) is used to evaluate performance. In general, the SIPP-based method resulted in higher labor costs, compared to the approach based on combining randomization and an IP.

Ingolfsson et al. (2002) present a genetic algorithm to obtain employee schedules, which satisfy a service level constraint (general background on genetic algorithms can be found in Goldberg (1989)). New schedules are generated by combining elements of two “parent” schedules drawn randomly from a population (i.e. a set of schedules generated during a previous iteration). Schedules performing well in terms of cost and service level have a larger probability of being chosen as a parent schedule. New schedules are then evaluated by numerically integration of ODE’s (cfr. Section 3.2.1) and these form the population in a next iteration. This process terminates when 100 new populations have been generated, or when the changes performance between subsequent populations become sufficiently small. This approach was tested (for relatively small system sizes) and compared to a method where staffing requirements were determined using SIPP and an optimal schedule was derived afterwards (aiming at minimal labor costs while meeting service level constraints). The latter will be referred to as the SIPP-IP method. In many settings, the genetic algorithm lead to better schedules than SIPP-IP (i.e. lower costs for the same service levels, or better service for equal costs), albeit at the expense of large
computation times (one run could require several hours). Moreover, the SIPP approach often leads to inaccurate performance measures, resulting in poor schedules.

In conclusion, the staffing methods described in this section succeed in determining schedules that satisfy service level constraints at a low labor cost. This is often done by means of an iterative procedure, in which the performance of a schedule is evaluated using the methods presented in Section 3.2.1. One the one hand, these staffing methods are appealing because no explicit assumptions concerning the distribution of waiting times are needed (resulting in general applicability). On the other hand, computation times are usually high, either because of the choice of the performance evaluation method (e.g. simulation), or because a large number of IP’s need to be solved before a good solution is found. As such, faster and more simple methods (such as the SRS-rule discussed in Section 3.1.2) might be preferred, especially in situations where these provide sufficiently good results.

4 Conclusion and future research

This paper gives an overview of methods that can be applied to set staffing levels in an emergency department, where demand for service varies throughout the day. The performance measures of interest are service levels, i.e. we want to guarantee that the probability of long waiting times is sufficiently small.

Two subproblems can be distinguished: on the one hand, staffing levels need to be determined for each moment of the day and on the other hand, the system’s performance resulting from a staffing level function needs to be assessed, to check whether the imposed performance constraints are met. Methods to address both problems were described, making a distinction between stationary approximations (which rely on a translation of the nonstationary model to a stationary equivalent) and nonstationary models (which explicitly account for the nonstationarity in the demand process). In terms of computational effort, the preference goes to stationary approximations, as these are usually fast and fairly easy to implement. However, under certain circumstances stationary approximations fail; in this case, one might resort to the often more complex and computationally expensive nonstationary methods.

An ED context is characterized by a small system size (few servers), low abandonment probabilities (most patients are willing or obliged to wait) and periods of overloading. Typically, the aim is to strike a balance between quality of service (in terms of waiting times) and efficiency (in terms of labor costs). Promising methods include

- Lag SIPP or MOL, combined with the square root staffing rule, because of the simplicity and limited computational effort. However, this requires an appropriate specification for the quality of service parameter $\beta$ in the square root staffing rule, to allow for waiting time related performance measures (and preferably general service times and abandonments as well).
- ISA, because of its applicability to general problem settings. The algorithm also needs adjustments to account for service level related performance measures. Moreover, an efficient implementation of the simulation model is crucial to keep computation times low.
- A combination of randomization or discrete time modeling techniques and the discussed scheduling algorithms. Randomization or discrete time modeling are appealing because they allow for an accurate evaluation of performance, in a relatively efficient way. Scheduling algorithms are useful because of the simultaneous treatment of staffing requirements and shift constraints, and because of the explicit consideration of labor costs (as opposed to focusing solely on meeting performance measure constraints).

For completeness, we would like to draw attention to some limitations of the methods discussed in this paper. In reality, an emergency department consists of a network of queues, whereas all described
models deal with a single-stage multiserver queue. For additional literature on time-dependent queueing networks, we would like to refer the interested reader to Liu and Whitt (2010a), Massey and Whitt (1993), Mandelbaum et al. (1998), Talreja and Whitt (2008), and references therein. Moreover, additional reading on modeling challenges that arise due to the presence of multiple server-types, multiple customer classes and priorities in the service process, can be found in Green et al. (2007), Harrison and Zeevi (2005), Chen and Henderson (2001), Bassamboo and Zeevi (2009), Bassamboo et al. (2006), Talreja and Whitt (2008), Mandelbaum et al. (1998), Atar 2005, Gurumurthi and Benjaafar (2004) and references therein.

Acknowledgements

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

References


Liu, Y., W. Whitt. 2009. Stabilizing customer abandonment in many-server queues with time-


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