doi:10.1007/s10479-009-0646-9 • 🌐 www.stefancreemers.be • ✉ info@stefancreemers.be

Queueing models for appointment-driven systems

Stefan Creemers
Marc Lambrecht

Abstract - **Many service systems are appointment-driven. In such systems, customers make an appointment and join an external queue (also referred to as the "waiting list"). At the appointed date, the customer arrives at the service facility, joins an internal queue and receives service during a service session. After service, the customer leaves the system. Important measures of interest include the size of the waiting list, the waiting time at the service facility and server overtime. These performance measures may support strategic decision making concerning server capacity (e.g. how often, when and for how long should a server be online). We develop a new model to assess these performance measures. The model is a combination of a vacation queueing system and an appointment system.**

Keywords - **appointment system, vacation model, overtime, waiting list, queueing system**

1 Introduction

In appointment-driven systems, service is administered only during predefined service sessions (e.g. during the opening hours of a doctors office). When making an appointment, a customer is assigned an appointment date (at some future service session) and joins a waiting list. At the appointment date, the customer leaves the waiting list and enters the service facility (e.g. a doctors office). At the service facility the customer once more joins a queue (e.g. the waiting room at the doctors office), receives service and leaves the system. Appointment-driven systems may be found in healthcare, legal services, administration and many other service systems.

It is clear that an appointment-driven system is in fact a combination of two distinct queueing systems. In a first queueing system, customers arrive at the queue (i.e. the waiting list) when making an appointment. At the appointment date the customer is removed from the waiting list and enters a second queueing system. In this second queueing system, the customer joins the queue at the service facility, receives the actual service and leaves the appointment-driven system. In the remainder of this article we will refer to both queueing systems as the appointment making queueing system (AMQ) and the service facility queueing system (SFQ) respectively. Both queueing systems require a rather distinct modeling approach. The AMQ can be considered as a vacation model while the SFQ is modeled as a so-called appointment system (AS). Building on the findings in both the literature on vacation models and the literature on AS, we combine the AMQ and SFQ to create a single model which allows the study of appointment-driven systems. We will refer to this combined model as the appointment-driven queueing system. Using the appointment-driven queueing system, we assess: (1) the time a customer spends in the waiting list; (2) the time a customer spends waiting at the service facility (this does not include the processing time itself); (3) The probability of a server to work overtime; (4) The amount of overtime a server performs. These performance measures can easily be implemented in an optimization procedure to support strategic decisions concerning server capacity (e.g. how often, when and for how long should a server be online).

The contribution of this article is twofold: (1) we present a new vacation model to model the AMQ; (2) we present a new model (the appointment-driven queueing system) to study an appointment-driven system and obtain several, strategically important performance measures. The remainder of this article is organized as follows. Section 2 gives a detailed problem description. Section 3 and 4 discuss the AMQ and SFQ respectively. In section 5 both models are combined to create the appointment-driven queueing system. Section 6 concludes.

2 Problem description

In this section we provide a detailed description of the dynamics at work at the appointment-driven system. First we define the problem setting. Next, we formally describe the basic concepts of the appointment-driven system.

2.1 Problem setting

We use a simple example to illustrate the problem setting. Imagine a doctor's office in which a single doctor sees patients every Thursday evening and every Friday afternoon. The doctor's office has opening hours from 6 PM until 8 PM on Thursday and from 2 PM until 6 PM on Friday. During these service sessions a maximum number of patients may be treated. Assume that on Thursday a maximum of 4 patients receives service. On Friday 8 patients may be served. Patients themselves call to make an appointment and are scheduled for service at the first service session in which the maximum number of patients has not yet been reached. For instance, suppose that on Monday 12 patients are already waiting for service. These patients will all be treated at the upcoming service sessions on Thursday and Friday. Assume that an additional patient arrives on Monday evening. The first service session in which there is still room available is on Thursday of the upcoming week. As such, we schedule this extra patient accordingly. We illustrate this procedure in Figure 1.
The making of an appointment indicates the arrival of a patient at the system. Until arrival at the doctor's office on the scheduled date, patients wait in an external queue (e.g. at home). We refer to this queue as the "waiting list". At the start of a service session, a number of patients is removed from the waiting list and is allowed to enter the doctor's office. At the doctor's office, patients are kept in the waiting room and are treated in order of arrival (FCFS). Patients leave the system after service completion. Often, the doctor has to work overtime in order to service all patients present in the waiting-room. Further assume that:

- When making an appointment, patients are assigned the first available time slot.

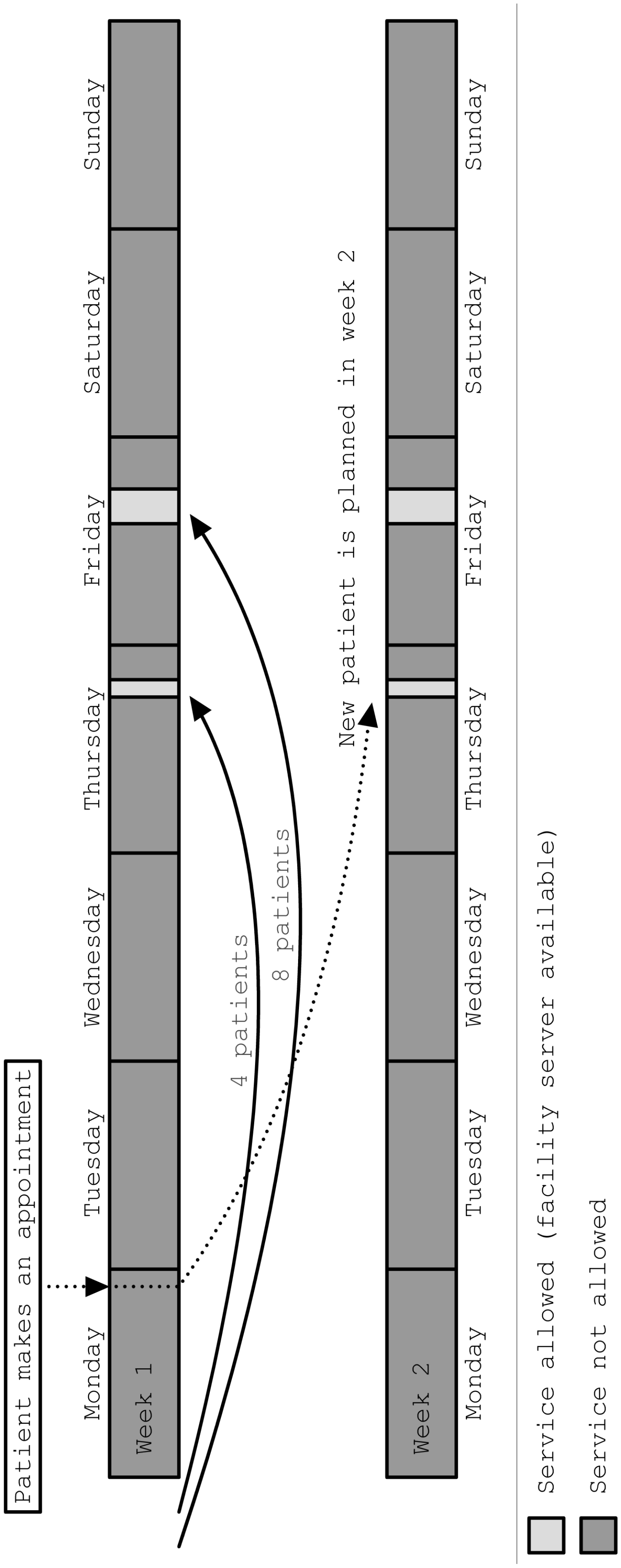

Figure 1: Scheduling of an appointment

Table 1: List of important definitions

J	The number of vacations/service sessions in a cycle
T	The length of a cycle of service sessions
T_j	The length of a vacation j
k_j	The maximum number of customers served at service session j
λ	The Poisson arrival rate of customers
υ_j	The rate of an Erlang phase used to approximate vacation j
V	The total number of phases of the Erlang distributions
μ	The service rate of customers at the service facility
C_s^2	The squared coefficient of variation of the service times
C_a^2	The squared coefficient of variation of the interarrival times

- Patients always show up on the appointed service session and they arrive on time.
- No unscheduled patients show up.
- All patients that arrive at the service session are served by the doctor (i.e. no balking occurs).
- The doctor provides service even if only a single patient has made an appointment during a given service session.

Most of these assumptions may easily be relaxed and serve only the purpose of maintaining transparency of the upcoming discourse.

In such a system, several strategically important performance measures may be assessed: (1) the time a customer spends in the waiting list; (2) the time a customer spends waiting at the service facility (this does not include the processing time itself); (3) The probability of a server to work overtime; (4) The amount of overtime a server performs. These performance measures can be used to determine the optimal frequency of service sessions (e.g. how often and when should a doctor see patients) as well as the optimal length of these service sessions (e.g. how much time should be spent servicing patients during a specific service session).

2.2 Problem definition

Prior to advancing to the formal description of the problem, we present an overview of the most important symbols in Table 1.

The service process at an appointment-driven system is a succession of service sessions during which customers are served at a single server. Each service session i (index i is defined as $i \in \{1, 2, \ldots\}$) is fully characterized by: (1) the maximum number of customers k_i allowed to receive service; (2) the length of the service session S_i; (3) the intersession time I_i (i.e. the time between the end of service session i and the start of service session $i + 1$; during which service at the service facility is unavailable). Figure 2 illustrates the service process at the appointment-driven system. We assume recurring cycles to be present in the succession of service sessions (e.g. a doctor receiving patients every Thursday evening and every Friday

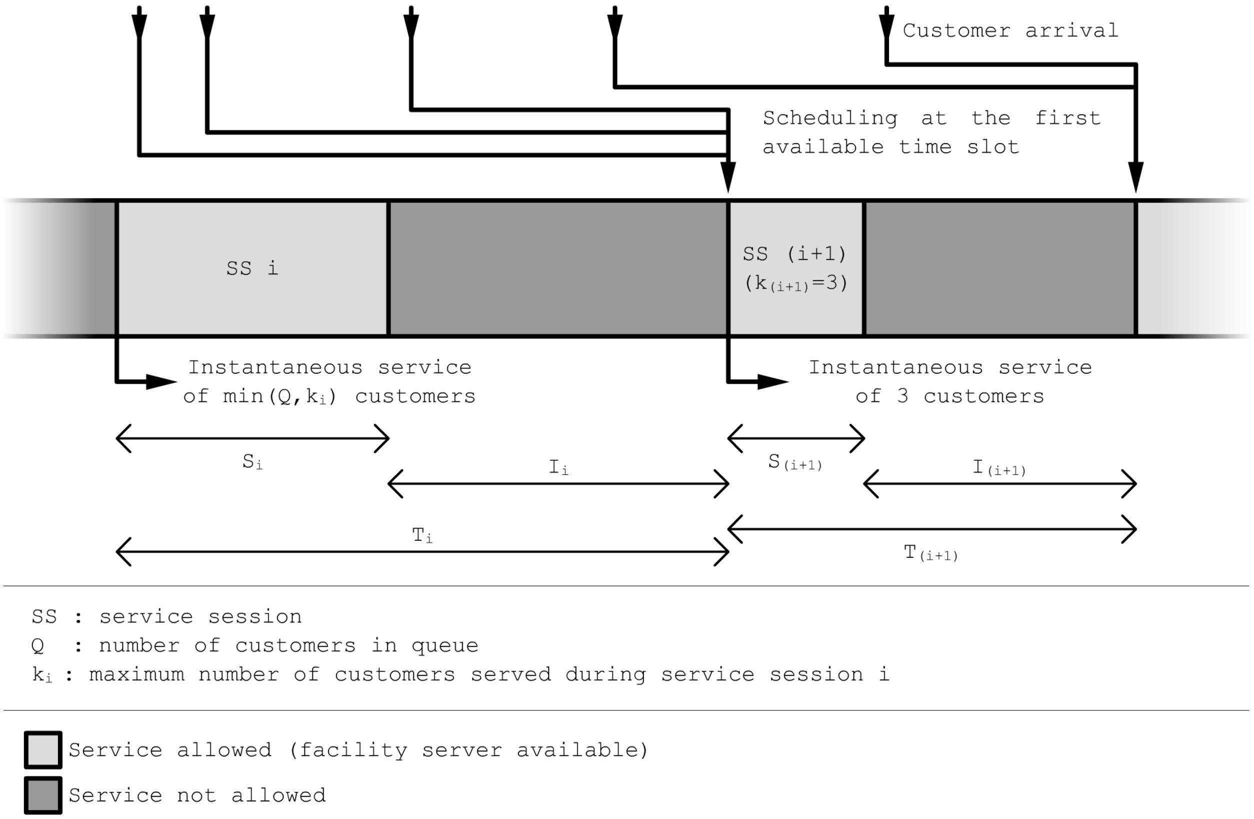
Customer arrival
Scheduling at the first available time slot
SS i
SS (i+1) ($k_{(i+1)}$=3)
Instantaneous service of min(Q,k_i) customers
Instantaneous service of 3 customers
S_i
I_i
$S_{(i+1)}$
$I_{(i+1)}$
T_i
$T_{(i+1)}$
SS : service session
Q : number of customers in queue
k_i : maximum number of customers served during service session i
Service allowed (facility server available)
Service not allowed

Figure 2: The service process at an appointment-driven system

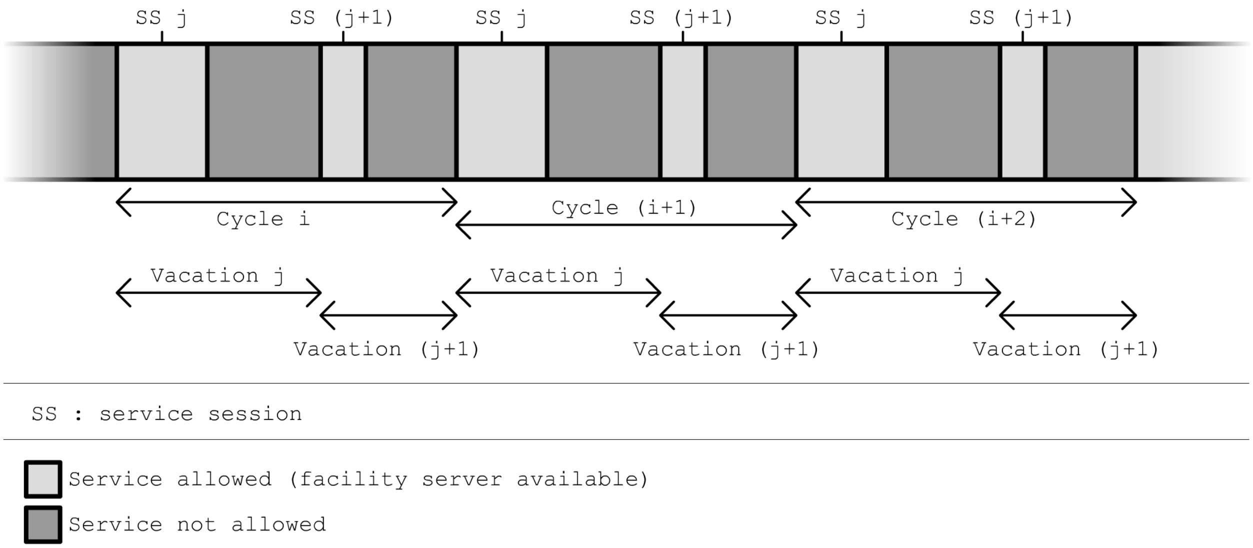

Figure 3: Succession of service cycles

afternoon). A cycle of service sessions has length T and contains J service sessions (indexed by $j \in \{1, \dots, J\}$). Note that, due to the cyclic nature of the service process, a service session of type $j + (iJ)$ is also a service session of type j. In addition, each service session i may be associated with a vacation i of deterministic length $T_i = S_i + I_i$. We illustrate these dynamics in Figure 3. In the example, each cycle of service sessions contains two service sessions (service sessions j and $j + 1$). Their corresponding vacations are of deterministic length T_j and T_{j+1}.

In this article, we model the deterministic vacation length using an Erlang distribution of sufficient phases. Each phase of the Erlang distribution is exponentially distributed with rate υ_i and

$$\frac{1}{\upsilon_i} = \frac{T_i}{V}, \tag{1}$$

where V is some number sufficiently large as to safeguard the approximation of a deterministic vacation length T_i by means of an Erlang distribution of parameters V and υ_i (note that as V approaches infinity, the variance of the resulting Erlang distribution approaches zero).

Whenever a customer makes an appointment, an arrival at the system takes place. The time between two successive appointments is assumed to be exponentially distributed with mean $^1/_\lambda$ and squared coefficient of variation $C_a^2 = 1$. The interarrival times of individual customers are assumed to be i.i.d. Note that the assumption of exponentially distributed interarrival times has only a limited impact on the precision of the model while it has been shown by Palm (1943) and Khinchin (1960) that the sum of a large numbers of independent renewal processes (i.e. the arrival processes of the different customers) will tend to a Poisson process. In addition, Lariviere and Van Mieghem (2004) show that the assumption of exponential interarrival times is reasonable in many service systems.

At the start of a service session i, $min(Q, k_i)$ customers are removed from the waiting list (where Q is the number of customers in queue at the start of the service session). These customers are served during service session i. The arrival of these customers at the service facility itself is managed by the AS. In our model we adopt a simple AS in which all customers are assumed to be present at the service facility at the start of the service session (this AS is also referred to as the block appointment rule). Note however that other AS can be implemented in the appointment-driven system. Once at the service facility, customers receive the actual service. Let ${}^{1}/_{\mu}$ and σ_s^2 denote the mean and the variance of the service time respectively. The squared coefficient of variation of the service times is given by $C_s^2 = \sigma_s^2 \mu^2$. In addition, the service times of individual customers are assumed to be i.i.d.

In this article, we use the gamma distribution to model the service times of the customers at the service facility. The gamma distribution is characterized by a shape parameter α and a scale parameter θ. The probability density function of the gamma distribution is:

$$f(x, \alpha, \theta) = x^{\alpha-1} \frac{e^{\frac{-x}{\theta}}}{\Gamma(\alpha)\theta^{\alpha}}. \tag{2}$$

The mean and variance of the gamma distribution are given by:

$$\frac{1}{\mu} = \alpha\theta, \tag{3}$$

$$\sigma_s^2 = \alpha\theta^2. \tag{4}$$

Note that other distributions may also be implemented in the appointment-driven system. For our purposes however, we use the gamma distribution while it provides a simple and transparent framework to model a general class of practical settings. The following set of features further motivates the use of the gamma distribution:

- The convolution of i i.i.d. gamma distributions of parameters α and θ results in a gamma distribution of parameters $i\alpha$ and θ.
- The gamma distribution may be used to match the first two moments of any continuous distribution in the $[0, \infty)$ interval.
- The truncated mean of the gamma distribution may easily be obtained (this feature is particularly useful to compute overtime performance measures).

Further note that we assume the probability of the server working overtime longer than the interval between subsequent service sessions to be negligible (i.e. we assume that there is no overlap in service between subsequent service sessions).

3 Appointment making queueing system

In this section we develop the AMQ. We first provide a problem definition and next present the model itself.

3.1 Problem definition

Over the past decades, queueing systems with server vacations have received a lot of attention in queueing literature. Vacation models observe the queueing behavior of systems in which the server takes a vacation (i.e. becomes unavailable) when certain conditions are met. Whenever a server leaves on a vacation, arriving customers are stored in the queue. Once the server returns, service begins once more. A wide variety of vacation models exists. For a general overview we refer to Doshi (1986), Takagi (1988) and Tian and Zhang (2006).

The AMQ consists of a single queue and a single virtual server. The virtual server acts as a device to allocate customers to service sessions (consequently, no processing time is required). At the start of a service session i, $min(Q, k_i)$ customers are served at the virtual server of the AMQ (i.e. the AMQ has a k-limited service discipline; where k depends on the service session that is about to start). After service, a vacation is initiated. This vacation has a deterministic length equal to the difference between the start of the current service session and the start of the next (i.e. a vacation i has length $T_i = S_i + I_i$). Note that, while service is instantaneous, the end of a vacation and the start of a new vacation occur at the same moment in time (i.e. the server is virtually always on vacation). During the vacation, arrivals are allowed to occur with rate λ. At the start of the next service session, the virtual server returns from vacation, instantaneously serves another batch of customers and once more leaves on a vacation of deterministic length.

The AMQ is a rather complex vacation model that has various unique features, rendering the modeling exercise rather complex (e.g. the length of a vacation as well as the value of k depends on the state of the system; on the service session that is about to start). To the best of our knowledge, no model exists in published literature that is able to cope with the prerequisites imposed by the AMQ.

3.2 The AMQ model

We model the AMQ using a continuous-time Markov chain (CTMC) $X = \{X(t) : t \geq 0\}$. The CTMC X is a threedimensional stochastic process whose statespace can be represented by triplets (Q, j, v), where:

- $Q : Q \in \{0, 1, 2, \ldots\}$ represents the number of customers in queue,
- $j : j \in \{1, 2, \ldots, J\}$ represents the vacation type,
- $v : v \in \{1, 2, \ldots, (V + 1)\}$ represents the phase of the vacation process.

For each queue size Q and each vacation type j we have V states in which either an arrival takes place (thereby incrementing the queue size Q) or a vacation phase is finished (indicating that the end of the vacation approaches). After finishing the final vacation phase (i.e. vacation phase V) of a vacation of type j, one ends up in a state in which the vacation process is at phase $(V + 1)$. At that point, the vacation of type j is finished. As such, the server returns from vacation instantaneously serves up to k_φ (where $\varphi = j + 1$ if $j < J$ and $\varphi = 1$ if $j = J$) customers and leaves on a vacation once more. No arrivals are allowed to occur during the infinitesimal amount of time during which the system remains in this state. Instead, a transition takes place towards a state in which: (1) the queue size Q is reduced by $min(Q, k_\varphi)$ customers; (2) the vacation phase v is reset at 1; (3) the vacation type j is set equal to φ. We can define the set of feasible state transitions as follows:

- Upon arrival of a customer (with rate λ), one moves from state (Q, j, v) to state $(Q + 1, j, v)$ if $v \leq V$.
- Upon finishing a vacation phase v at a vacation j (with rate υ_j), one moves from state (Q, j, v) to state $(Q, j, v + 1)$ if $v \leq V$.
- Upon finishing a vacation of type j, one moves from state $(Q, j, V + 1)$ to state $(max(0, Q - k_\varphi), \varphi, 1)$ (with infinitesimal rate ω).

Using these state transitions, we can construct the infinitesimal generator $\mathbf{Q}$ that is associated with the CTMC X. The infinitesimal generator $\mathbf{Q}$ is given by:

$$\mathbf{Q} = \begin{bmatrix} \hat{\mathbf{L}} & \mathbf{F} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{B} & \mathbf{L} & \mathbf{F} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \ddots \end{bmatrix},$$

where $\mathbf{0}$ is a matrix of appropriate size containing only zeros and where $\hat{\mathbf{L}}$, $\mathbf{L}$, $\mathbf{F}$ and $\mathbf{B}$ are the respective “local”, “forward” and “backward” transition rate matrices. An outline of these matrices is provided below (s and t represent the queue size at the departure and arrival state respectively):

$$\hat{\mathbf{L}} = \begin{array}{c|ccccc|} {}^{s}/_{t} & 0 & 1 & \cdots & Q_c-2 & Q_c-1 \\ 0 & \hat{\mathbf{L}}^{*} & \mathbf{F}^{*} & \cdots & \mathbf{0} & \mathbf{0} \\ 1 & \mathbf{B}^{*}_{\mathbf{s,t}} & \mathbf{L}^{*} & \cdots & \mathbf{0} & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ Q_c-2 & \mathbf{B}^{*}_{\mathbf{s,t}} & \mathbf{B}^{*}_{\mathbf{s,t}} & \cdots & \mathbf{L}^{*} & \mathbf{F}^{*} \\ Q_c-1 & \mathbf{B}^{*}_{\mathbf{s,t}} & \mathbf{B}^{*}_{\mathbf{s,t}} & \cdots & \mathbf{B}^{*}_{\mathbf{s,t}} & \mathbf{L}^{*} \end{array},$$

$$\mathbf{L} = \begin{array}{c|ccccc|} {}^{s}/_{t} & iQ_c & iQ_c+1 & \cdots & 2iQ_c-2 & 2iQ_c-1 \\ iQ_c & \mathbf{L}^{*} & \mathbf{F}^{*} & \cdots & \mathbf{0} & \mathbf{0} \\ iQ_c+1 & \mathbf{B}^{*}_{\mathbf{s,t}} & \mathbf{L}^{*} & \cdots & \mathbf{0} & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 2iQ_c-2 & \mathbf{B}^{*}_{\mathbf{s,t}} & \mathbf{B}^{*}_{\mathbf{s,t}} & \cdots & \mathbf{L}^{*} & \mathbf{F}^{*} \\ 2iQ_c-1 & \mathbf{B}^{*}_{\mathbf{s,t}} & \mathbf{B}^{*}_{\mathbf{s,t}} & \cdots & \mathbf{B}^{*}_{\mathbf{s,t}} & \mathbf{L}^{*} \end{array},$$

$$\mathbf{F} = \begin{array}{c|ccccc|} {}^{s}/_{t} & iQ_c & iQ_c+1 & \cdots & 2iQ_c-2 & 2iQ_c-1 \\ (i-1)Q_c & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ (i-1)Q_c+1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ (i-1)Q_c+Q_c-2 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ (i-1)Q_c+Q_c-1 & \mathbf{F}^{*} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{array},$$

$$\mathbf{B} =$$

$$\begin{array}{c|ccccc|}
{}^{s}\!/_{t} & (i-1)Q_c & (i-1)Q_c+1 & \cdots & (i-1)Q_c+Q_c-2 & (i-1)Q_c+Q_c-1 \\
\hline
iQ_c & \mathbf{B^*_{s,t}} & \mathbf{B^*_{s,t}} & \cdots & \mathbf{B^*_{s,t}} & \mathbf{B^*_{s,t}} \\
iQ_c+1 & \mathbf{0} & \mathbf{B^*_{s,t}} & \cdots & \mathbf{B^*_{s,t}} & \mathbf{B^*_{s,t}} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
2iQ_c-2 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{B^*_{s,t}} & \mathbf{B^*_{s,t}} \\
2iQ_c-1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B^*_{s,t}}
\end{array},$$

where $Q_c = max(k_j);\ \forall j \in \{1, 2, \ldots, J\}$. Q_c is also referred to as the critical queue size and indicates the maximum decrease of queue size when performing a backward transition (i.e. no more than Q_c customers may be removed from the queue at the end of any vacation $j;\ j \in \{1, 2, \ldots, J\}$). The matrices $\hat{\mathbf{L}}^*$, $\mathbf{L}^*$, $\mathbf{F}^*$ and $\mathbf{B^*_{s,t}}$ are given by (u and w represents the vacation type of the departure and arrival state respectively)

$$\hat{\mathbf{L}}^* = \begin{array}{c|ccccc|}
{}^{u}\!/_{w} & 1 & 2 & \cdots & J-1 & J \\
\hline
1 & \Upsilon_u & \Omega_{s,t,w} & \cdots & \mathbf{0} & \mathbf{0} \\
2 & \mathbf{0} & \Upsilon_u & \cdots & \mathbf{0} & \mathbf{0} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
J-1 & \mathbf{0} & \mathbf{0} & \cdots & \Upsilon_u & \Omega_{s,t,w} \\
J & \Omega_{s,t,w} & \mathbf{0} & \cdots & \mathbf{0} & \Upsilon_u
\end{array},$$

$$\mathbf{L}^* = \begin{array}{c|ccccc|}
{}^{u}\!/_{w} & 1 & 2 & \cdots & J-1 & J \\
\hline
1 & \Upsilon_u & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\
2 & \mathbf{0} & \Upsilon_u & \cdots & \mathbf{0} & \mathbf{0} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
J-1 & \mathbf{0} & \mathbf{0} & \cdots & \Upsilon_u & \mathbf{0} \\
J & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \Upsilon_u
\end{array},$$

$$\mathbf{F}^* = \begin{array}{c|ccccc|}
{}^{u}\!/_{w} & 1 & 2 & \cdots & J-1 & J \\
\hline
1 & \mathbf{\Lambda} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\
2 & \mathbf{0} & \mathbf{\Lambda} & \cdots & \mathbf{0} & \mathbf{0} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
J-1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{\Lambda} & \mathbf{0} \\
J & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{\Lambda}
\end{array},$$

$$\mathbf{B^*_{s,t}} = \begin{array}{c|ccccc|}
{}^{u}\!/_{w} & 1 & 2 & \cdots & J-1 & J \\
\hline
1 & \mathbf{0} & \Omega_{s,t,w} & \cdots & \mathbf{0} & \mathbf{0} \\
2 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
J-1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \Omega_{s,t,w} \\
J & \Omega_{s,t,w} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0}
\end{array}.$$

The matrices Υ_u, Λ and $\Omega_{s,t,w}$ are the characterizing matrices of the infinitesimal generator $\mathbf{Q}$. They are presented below

$$\Upsilon_u = \begin{array}{c|ccccc|} v & 1 & 2 & \cdots & V & V+1 \\ 1 & -\lambda-\upsilon_u & \upsilon_u & \cdots & 0 & 0 \\ 2 & 0 & -\lambda-\upsilon_u & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ V & 0 & 0 & \cdots & -\lambda-\upsilon_u & \upsilon_u \\ V+1 & 0 & 0 & \cdots & 0 & -\omega \end{array},$$

$$\Lambda = \begin{array}{c|ccccc|} v & 1 & 2 & \cdots & V & V+1 \\ 1 & \lambda & 0 & \cdots & 0 & 0 \\ 2 & 0 & \lambda & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ V & 0 & 0 & \cdots & \lambda & 0 \\ V+1 & 0 & 0 & \cdots & 0 & 0 \end{array},$$

$$\Omega_{s,t,w} = \begin{array}{c|ccccc|} v & 1 & 2 & \cdots & V & V+1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 2 & 0 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ V & 0 & 0 & \cdots & 0 & 0 \\ V+1 & \omega\delta_{s,t,w} & 0 & \cdots & 0 & 0 \end{array},$$

where $\delta_{s,t,w}$ may be defined as:

$$\delta_{s,t,w} = \begin{cases} 1 & \text{if } (s-t) = k_w;\ \forall t > 0, \\ 1 & \text{if } s \leq k_w;\ \forall t = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

One can observe that the infinitesimal generator $\mathbf{Q}$ is endowed with a special repetitive structure. This repetitive structure may be exploited when deriving the stationary distribution π of the corresponding CTMC X. To obtain π we adopt matrix analytical techniques. Pioneered by Neuts (1981) several decades ago, matrix analytical techniques have attracted the attention of many researchers in the queueing field. For an overview of literature and an introduction to matrix analytical techniques, refer to Latouche and Ramaswami (1999), Osogami (2005) and Bini, Meini, Steffe and Van Houdt (2006) among others. In short, matrix analytical techniques allow the (numerically) exact analysis of a wide variety of queueing systems featuring some repetitive structure (more specifically, $M/G/1$, $GI/M/1$ and quasi-birth-death (QBD) processes). The AMQ may be considered a QBD process and may be solved using the techniques that apply for $M/G/1$ as well as $GI/M/1$ processes. Obtaining the stationary distribution of a QBD process involves the computation of an auxiliary matrix $\mathbf{R}$. $\mathbf{R}$ may be obtained as the solution of the quadratic equation (Latouche and Ramaswami 1999):

$$\mathbf{F} + \mathbf{R} \cdot \mathbf{L} + \mathbf{R}^2 \cdot \mathbf{B} = \mathbf{0}. \tag{6}$$

The stationary probability vector $\pi^{(0)}$ may be obtained by solving the following system:

$$\pi^{(0)}\left(\hat{\mathbf{L}}+\mathbf{R}\mathbf{B}\right) = \mathbf{0}, \tag{7}$$

$$\pi^{(0)}\left(\mathbf{I}-\mathbf{R}\right)^{-1}\mathbf{e} = 1, \tag{8}$$

whereas the stationary distribution π is obtained through the recursive relationship:

$$\pi^{(i)} = \pi^{(0)}\cdot\mathbf{R}^{i};\ \forall i \geq 1. \tag{9}$$

Where:

- $\pi^{(i)}$ is the vector of stationary probabilities associated with a queue size $s : s \in \{(i-1)Q_c, \ldots, iQ_c\}$. More specifically, $\pi^{(i)}$ holds the stationary probabilities of states $(s, j, v) : \forall s \in \{(i-1)Q_c, \ldots, iQ_c\} \wedge \forall j \in \{1, 2, \ldots, J\} \wedge \forall v \in \{1, 2, \ldots, V+1\}$.
- $\mathbf{I}$ is an identity matrix of appropriate dimension.
- $\mathbf{e}$ is a vector of ones of appropriate size.

From $\pi^{(i)}$ we obtain $\pi\left(Q, j, v\right)$, the probability of having Q customers in queue at a vacation of type j at vacation phase v. We use $\pi\left(Q, j, v\right)$ to determine: (1) the stationary distribution $\pi_{SFQ,j}\left(Q\right)$ of the number of customers to be served at the SFQ during a service session of type j; (2) the stationary distribution $\pi_{AMQ,j}\left(Q\right)$ of the number of customers in queue during a vacation of type j.

The number of customers to be served at a service session φ depends on the stationary probability of states $(Q, j, V+1)$. The stationary distribution $\pi_{SFQ,\varphi}\left(Q\right)$ may be obtained as follows:

$$\pi_{SFQ,\varphi}\left(Q\right) = \frac{\pi\left(Q, j, V+1\right)}{\sum\limits_{Q=0}^{\infty}\pi\left(Q, j, V+1\right)};\ \forall Q < k_{\varphi}, \tag{10}$$

$$\pi_{SFQ,\varphi}\left(k_{\varphi}\right) = \frac{\sum\limits_{Q=k_{\varphi}}^{\infty}\pi\left(Q, j, V+1\right)}{\sum\limits_{Q=0}^{\infty}\pi\left(Q, j, V+1\right)}. \tag{11}$$

The number of customers in queue during a vacation of type j is associated with the stationary distribution of states $(Q, j, v) : v \in \{1, 2, \ldots, V\}$. After rescaling we obtain:

$$\pi_{AMQ,j}\left(Q\right) = \frac{\sum\limits_{v=1}^{V}\pi\left(Q, j, v\right)}{\sum\limits_{Q=0}^{\infty}\sum\limits_{v=1}^{V}\pi\left(Q, j, v\right)}. \tag{12}$$

Using $\pi_{AMQ,j}\left(Q\right)$ we can compute the average number of customers in queue at the AMQ during a service session of type j as follows:

$$\overline{Q}_{AMQ,j} = \sum_{Q=0}^{\infty} Q\pi_{AMQ,j}\left(Q\right). \tag{13}$$

The probability of finding oneself at a vacation of type j equals:

$$p_j = \frac{T_j}{\sum\limits_{j=1}^{J} T_j}. \tag{14}$$

As such, the average number of customers in queue at the AMQ equals:

$$\overline{Q}_{AMQ} = \sum_{j=1}^{J} p_j \overline{Q}_{AMQ,j}. \tag{15}$$

Using Little's law, we can compute the expected waiting time of a customer at the AMQ:

$$E\left[W_{AMQ}\right] = \frac{\overline{Q}_{AMQ}}{\lambda}. \tag{16}$$

4 Service facility queueing system

In this section we develop the SFQ. We provide a short overview of literature on AS. Next we define the problem. A final subsection presents the model itself.

4.1 Appointment system literature review

AS have been studied extensively during the past 50 years. Excellent overviews of literature may be found with Mondschein and Weintraub (2003) and Cayirli (2003). In short, AS deal with the operational issue of scheduling a number of customers as to optimize some measure of performance (e.g. customer waiting time, staff overtime, . . .). In the most simple case, all customers arrive punctually at their appointment dates and receive service at a single server workstation. Complexity is introduced in the form of so-called environmental variables. An extensive overview of such environmental variables is provided in Cayirli (2003). Examples of environmental variables include customer unpunctuality, the number of customer classes and the number of servers.

In AS literature, customers are either scheduled using some appointment scheduling rule or a procedure is developed to determine the (optimal) arrival times of customers at the service facility in order to optimize some set of performance measures (examples of the latter category may be found with Weiss (1990), Liao, Pegden and Roshenshine (1993), Wang (1997), Vanden Bosch and Dietz (2000; 2001) among others). With respect to appointment scheduling rules, comprehensive comparisons of various appointment scheduling rules are available with Ho and Lau (1992; 1999) and Mondschein et al. (2003). In the remainder of this work, we will focus only on appointment scheduling rules.

Appointment scheduling rules can be described in terms of:

- block size (n_{i_l}); indicating the number of customers scheduled in block l during service session i,
- initial block size (n_{i_1}); indicating the number of customers given an appointment date at the start of service session i,

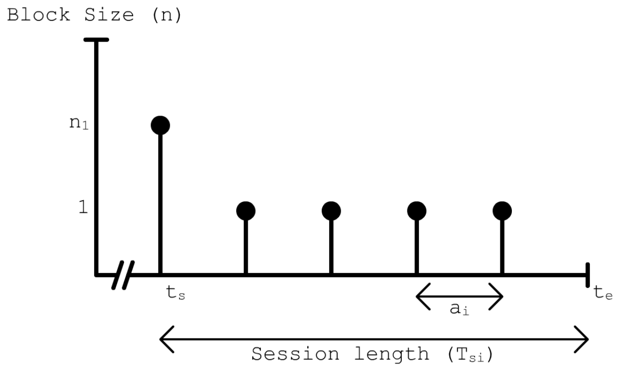

Figure 4: Appointment scheduling rules

- appointment interval (a_{i_l}); indicating the interval between two successive appointments during service session i.

Note that all but a few AS reported in literature study a single service session. Vanden Bosch & Dietz (2000 and 2001) are one of the few exceptions to study an AS spanning over multiple service sessions. Each service session i of length S_i is divided in a number of blocks B; t_s and t_e indicating the start of the first and the end of the last block respectively. At the beginning of each block b; $b \in \{1, 2, \ldots, B\}$, a number of customers n_b is scheduled to arrive. Figure 4 provides further insight. Many appointment scheduling rules start a service session with an initial block of a few customers (who serve as a buffer to minimize server idle time in the occasion of customers arriving late or failing to show up) and constant appointment intervals. When $n_{i_1} = 2, n_{i_l} = 1, a_{i_l} = {}^1\!/\!_\mu$, the appointment scheduling rule is referred to as the Bailey-Welch rule. Another popular appointment scheduling rule is the block appointment rule in which all customers are assigned to arrive in the initial block. Notwithstanding their simplicity, the Bailey-Welch and block appointment rule are well-known and widely implemented in practice.

4.2 Problem definition

In this article, we model the SFQ as an AS using the block appointment rule. We assume no environmental variables to be in effect. As such, all customers are present at the start of their assigned service session. Service starts at the beginning of a service session and continues uninterruptedly until all customers have been served. Under such a policy, customer waiting time is maximized while server idle time is minimized.

The SFQ models the service process of customers at a single service session. While the service process is stochastic, there exists a probability that overtime has to be performed. Overtime is the time a server has to work in excess of a certain time capacity O_j in order to serve all customers at a service session of type j. We define O_j as follows:

$$O_j = \frac{k_j}{\mu}. \tag{17}$$

In the literature on AS, the concept of overtime is regularly encountered. However, AS are generally limited to the study of a single service session. Research relating to overtime in a more general setting (i.e. a queueing system) is rather rare. Bitran and Tirupati (1991) study the subject in the context of a traditional queueing system. Their results however, remain limited to approximations and are focussed on systems that are not appointment-driven.

4.3 The SFQ model

The SFQ models the service process of Q customers at a service session j. The measures of interest are: (1) the expected waiting time of an individual customer at the SFQ (this does not include processing itself); (2) the probability of the server to perform overtime; (3) the expected amount of overtime performed.

The expected waiting time of an individual customer at the SFQ (given a service session of type j and a number of customers to be served Q) is given by (Lambrecht et al. 1998):

$$E\left[W_{SFQ,j,Q}\right] = \frac{Q-1}{2\mu}. \tag{18}$$

In order to compute $\pi_o\left(j,Q\right)$ (i.e. the probability that the server performs overtime at a vacation of type j when Q customers require service) we require the distribution of the total service time at service session j. The service processes of Q individual customers are assumed to follow i.i.d. gamma distributions of parameters α and θ. While the service process of the Q customers occurs uninterruptedly, the total service time distribution is the convolution of Q i.i.d. gamma distributions of parameters α and θ, yielding a gamma distribution of parameters $Q\alpha$ and θ (Dudewicz and Mishra 1988). The cumulative distribution function (cdf) of the total service time is given by:

$$F\left(x, Q\alpha, \theta\right) = \frac{\gamma\left(Q\alpha, {}^{x}\!/\!{}_{\theta}\right)}{\Gamma\left(Q\alpha\right)}. \tag{19}$$

Where γ represents the incomplete gamma function. Using the cdf of the total service time, we obtain the probability of the server to perform overtime at a service session of type j when Q customers require service:

$$\pi_o\left(j,Q\right) = 1 - F\left(O_j, Q\alpha, \theta\right). \tag{20}$$

The expected amount of overtime performed at a service session of type j with Q customers requiring service, is determined using the truncated distribution of $f\left(x, Q\alpha, \theta\right)$. More specifically, the expected amount of overtime equals:

$$\frac{1}{\mu_o\left(j,Q\right)} = \int\limits_{O_j}^{\infty} \left(x - O_j\right) f\left(x, Q\alpha, \theta\right) dx. \tag{21}$$

Which can be simplified to the following closed form formula:

$$\frac{1}{\mu_o\left(j,Q\right)} = \frac{\left[-O_j \gamma\left(Q\alpha, {}^{O_j}\!/\!{}_{\theta}\right)\right] + \left[O_j^{Q\alpha}\left(\frac{O_j}{\theta}\right)^{-Q\alpha} \theta^{1-Q\alpha} \gamma\left(1+Q\alpha, {}^{O_j}\!/\!{}_{\theta}\right)\right]}{\Gamma\left(Q\alpha\right)}. \tag{22}$$

5 Appointment driven queueing system

In this section we combine both the AMQ and the SFQ to create a single model, the appointment-driven queueing system, that is able to study an appointment-driven system. A first section presents the appointment-driven queueing system. In a second section we return to the numerical example first presented in section 2.1 and solve it using the appointment-driven queueing system.

5.1 The appointment-driven queueing system

From the AMQ we have obtained $\pi_{SFQ,j}(Q)$, the stationary distribution of the number of customers to be served at the SFQ during a service session of type j. We will use the stationary distribution $\pi_{SFQ,j}(Q)$ as a weighing factor for the results obtained at the SFQ corresponding to Q customers served at a service session j. As such we obtain general results at the appointment-driven queueing system (i.e. average customer waiting time at the service facility, probability of server overtime and the expected amount of overtime performed).

Define $E[W_{SFQ,j}]$, the average waiting time of a customer at the service facility during a service session of type j. $E[W_{SFQ,j}]$ may be obtained as follows:

$$E[W_{SFQ,j}] = \sum_{Q=0}^{k_j} \pi_{SFQ,j}(Q)\, E[W_{SFQ,j,Q}] \tag{23}$$

In addition, the average number of customers present at the start of a service session of type j may be defined as:

$$\overline{Q}_{SFQ,j} = \sum_{Q=0}^{k_j} \pi_{SFQ,j}(Q)\, Q. \tag{24}$$

For a given service session j, the average number of customers served will serve as the weighing factor of the average waiting time (i.e. the results of a service session in which a lot of customers receive service has a larger impact on the average waiting time of a customer in overall). We obtain the average waiting time of a customer at the service facility as follows:

$$E[W_{SFQ}] = \frac{\sum\limits_{j=1}^{J} E[W_{SFQ,j}]\, \overline{Q}_{SFQ,j}}{\sum\limits_{j=1}^{J} \overline{Q}_{SFQ,j}}. \tag{25}$$

With respect to the probability of the server working overtime at a service session of type j, we have:

$$\pi_o(j) = \sum_{Q=0}^{k_j} \pi_{SFQ,j}(Q)\, \pi_o(j,Q). \tag{26}$$

While there are J service sessions in a service cycle, the probability of randomly picking a service session j from the set of service sessions equals:

$$q_j = \frac{1}{J}. \tag{27}$$

Therefore the total probability of the server to work overtime is given by:

$$\pi_o = \sum_{j=1}^{J} q_j \pi_o\left(j\right). \tag{28}$$

Analogously we have that the expected amount of overtime at a service session of type j may be expressed as:

$$\frac{1}{\mu_o\left(j\right)} = \sum_{Q=0}^{k_j} \pi_{SFQ,j}\left(Q\right) \frac{1}{\mu_o\left(j,Q\right)}. \tag{29}$$

The total expected amount of overtime performed at the server equals:

$$\frac{1}{\mu_o} = \sum_{j=1}^{J} q_j \frac{1}{\mu_o\left(j\right)}. \tag{30}$$

The total expected waiting time at the AMQ is given in equation 16. Together with equation 25, 28 and 30, all performance measures of interest at the appointment-driven system are defined.

5.2 Numerical example

In this section we revisit the setting of the example discussed in section 2.1. In addition, assume that on average 8 patients make an appointment at the doctor's office every week (i.e. patients arrive at a rate of $\lambda = {}^{1}\!/_{1,260}$ per minute during a service cycle of length $T = 10,080$ minutes). While 12 patients are allowed to receive service in a single service cycle, the utilization rate of the doctor's office may be expressed as follows:

$$\rho = \lambda \frac{\sum\limits_{j=1}^{J} T_j}{\sum\limits_{j=1}^{J} k_j}. \tag{31}$$

Note that all parameters are expressed in minutes unless mentioned otherwise. In our example $\rho = {}^{2}\!/_{3}$. Further assume the service times to follow a gamma distribution of parameters $\alpha = 1.5$ and $\theta = 20$. The mean and variance of the service times amount to ${}^{1}\!/_{\mu} = 30$ minutes and $\sigma_s^2 = 600$ minutes respectively. The squared coefficient of variation is given by $C_s^2 = {}^{2}\!/_{3}$.

At this point, we will use the appointment-driven queueing system as developed in the previous sections to obtain the performance measures of interest. In order to assess the impact (on the accuracy of the results) of approximating the deterministic vacation times using an Erlang distribution of V phases, we perform a number of experiments featuring different values of V. The results of the analysis are presented in Table 2. The simulation results corresponding to values $V : V \in \{10, 50, 100, 200\}$ clearly demonstrate the validity of the analytical model. With respect to the accuracy of the model, we compare analytical results with simulation results when $V = \infty$ (i.e. the simulation was performed using deterministic vacation lengths). One can observe that the appointment-driven queueing system

Table 2: Results with varying number of vacation phases

V	$E\left[W_{AMQ}\right]$		$E\left[W_{SFQ}\right]$		π_o		$\frac{1}{\mu_o}$	
	Model	Sim	Model	Sim	Model	Sim	Model	Sim
10	5,127	5,125	105.95	105.94	0.198	0.198	10.08	10.08
50	4,440	4,440	106.54	106.56	0.188	0.188	9.60	9.61
100	4,360	4,360	106.67	106.68	0.187	0.186	9.53	9.53
200	4,321	4,320	106.74	106.76	0.186	0.186	9.50	9.51
∞		4,281		106.82		0.185		9.47

is able to provide very accurate results when assessing strategic performance measures at the appointment-driven system. When the vacation process is approximated by an Erlang distribution of 200 phases the results nearly match those obtained in the simulation when deterministic vacation lengths were used. Even an Erlang approximation of 50 phases performs well.

With respect to the server itself, one may observe that in nearly one out of five service sessions overtime is performed. The total expected amount of overtime encountered amounts to 9.5 minutes at a service session. These figures are relatively surprising considering the fact that: (1) the utilization rate of the server only amounts to $^2/_3$; (2) the service process of customers features low variability ($C_s^2 = {}^2/_3$); (3) the AS used minimizes server overtime (all customers are present at the start of a session, customers are not allowed to arrive late, unscheduled customers are not allowed to show up, ...). These observations illustrate the importance of assessing overtime in queueing models. Indeed, there is a pressing need for tools that are able to detect, not only the impact, but also the levers required to minimize the harmful effects of overtime. An optimization procedure indicating how often a server should be online, for how long and when, is of great strategic value to any administrator of an appointment-driven system. The performance measures obtained using the appointment-driven queueing system developed in this article, provide the tools to construct such an optimization procedure.

In the optimization procedure, a yet to be defined solution space is to be searched. As such, it is important to have an indication of the computational effort involved in obtaining performance measures at the appointment-driven system. With respect to the numerical example presented above, the computation times corresponding to the different instances are reported in Table 3. The computations are performed on an AMD Athlon with 2.0 GHz CPU-speed and 768 MB of RAM. It is clear that more complex real-life problems (featuring larger values of Q_c and J) require a trade-off between precision and model accuracy. Therefore, improving computational performance is key to the successful implementation of the model in an optimization procedure fit to assess more complex problems.

6 Conclusion

Appointment-driven systems are widespread in services. Important strategic performance measures in such systems include the time spent at the waiting list, the waiting time at the

Table 3: Computation times (in seconds) when varying the number of vacation phases

V	CPU time
10	2
50	103
100	726
200	6,907

service facility itself and the overtime performed by the server. These measures of interest may support strategic decision making concerning server capacity.

In this article we show that traditional queueing models are unable to accurately assess the performance of appointment-driven systems. The model we develop is fit for purpose and offers a large amount of modeling freedom. The model is a combination of a vacation queueing system and an appointment system. The vacation queueing system is a complex bulk service model with a G-limited service discipline, vacations of deterministic length and various state dependencies. With respect to the appointment system, the block appointment rule was selected to manage the arrival of customers at the service facility (it should be noted that other appointment systems can be modeled as well, however at the price of increased model complexity). Both systems are combined to create a queueing system that assesses performance measures of the appointment-driven system. A numerical example (and corresponding simulation validation study) shows that the model is able to provide very accurate results.

It is clear that both a vacation model and an appointment system are required to assess the performance of an appointment-driven system. The study of the vacation model or the appointment system separately, would only offer a myopic view of the problem setting. On the one hand, the vacation model is limited to the dynamics of the waiting list and remains blind to what happens at the service facility itself. Appointment systems on the other hand, have no input on the number of customers requiring service during a service session. As such, appointment systems are able to optimize performance at a single service session (i.e. local) but fail to optimize the service process as a whole (i.e. global, over all service sessions). The model developed in this article, provides the strategic performance measures required to perform such a global optimization. More specifically, the model allows the development of an optimization procedure that may be used (among others) to determine the optimal frequency of service sessions (e.g. how often and when should a server be online) as well as the optimal length of these service sessions (e.g. how much time should be spent servicing customers during a specific service session).

While the presented model provides a new approach to analyze appointment-driven systems, a considerable amount of work is left to be done. Future extensions of the model may include: (1) the adoption of multiple servers at the service facility; (2) a general, time-dependent arrival process using phase type distributions; (3) the use of different appointment systems that relax the assumptions imposed in this work. In addition, future research should focus on increasing the computational performance of the analytical model.

References

Bini, D., Meini, B., Steffe, S., & Van Houdt, B. (2006). Structured Markov chains solver: algorithms. In ACM international conference proceeding series, *proceedings of SMCtools*. Pisa, Italy.

Bitran, G., & Tirupati, D. (1991). Approximations for networks of queues with overtime. *Management Science*, *37*, 282–300.

Cayirli, T., & Veral, E. (2003). Outpatient scheduling in health care: a review of literature. *Production and Operation Management*, *12*, 519–549.

Doshi, B. T. (1986). Queueing systems with vacations - a survey. *Queueing Systems*, *1*, 29–66.

Dudewicz, E. J., & Mishra, S. N. (1988). *Modern mathematical statistics*. New York: John Wiley Sons.

Ho, C., & Lau, H. (1992). Minimizing total cost in scheduling outpatient appointments. *Management Science*, *38*, 1750–1764.

Ho, C., & Lau, H. (1999). Evaluating the impact of operating conditions on the performance of appointment scheduling rules in service systems. *European Journal of Operational Research*, *112*, 542–553.

Khinchin, A. J. (1960). *Mathematical Methods in the Theory of Queueing*. New York: Hafner.

Lambrecht, M. R., Ivens, P. L., & Vandaele N. J. (1998). ACLIPS: a capacity and lead time integrated procedure for scheduling. *Management Science*, *44*, 1548–1561.

Lariviere, M. A., & Van Mieghem, J. A. (2004). Strategically seeking service: how competititon can generate Poisson arrivals. *Manufacturing & Service Operations Management*, *6*, 23–40.

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

Liao, C., Pegden, C. D., & Roshenshine, M. (1993). Planning timely arrivals to a stochastic production or service system. *IIE Transactions*, *25*, 63–73.

Mondschein, S. V., & Weintraub, G. Y. (2003). Appointment policies in service operations: a critical analysis of the economic framework. *Production and Operations Management*, *12*, 266–286.

Neuts, M. F. (1981). *Matrix-geometric solutions in stochastic models*. Baltimore: Johns Hopkins University Press.

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

Palm, C. (1943). Intensittsschwankungen im Fernsprechverkehr. *Ericsson Technics*, *44*, 1–89.

Takagi, H. (1988). Queueing analysis of polling models. *ACM Computing Surveys*, *20*, 5–28.

Tian, N., & Zhang, Z. (2006). *Vacation queueing models*. New York: Springer Science.

Vanden Bosch, P. M., & Dietz, D. C. (2000). Minimizing expected waiting in a medical appointment system. *IIE Transactions*, *32*, 841–848.

Vanden Bosch, P. M., & Dietz, D. C. (2001). Scheduling and sequencing arrivals to an appointment system. *Journal of Service Research*, *4*, 15–25.

Wang, P. P. (1997). Optimally scheduling N customer arrival times for a single-server system. *Computers and Operations Research*, *24*, 703–716.

Weiss, E. N. (1990). Models for determining estimated start times and case orderings in hospital operating rooms. *IIE Transactions*, *22*, 143–150.