KATHOLIEKE UNIVERSITEIT
LEUVEN

Faculty of Business and Economics

Queueing models for appointment-driven systems

Stefan Creemers and Marc Lambrecht

DEPARTMENT OF DECISION SCIENCES AND INFORMATION MANAGEMENT (KBI)

KBI 0805

Queueing models for appointment-driven systems

Stefan Creemers[1] and Marc Lambrecht[1]

[1] Faculty of Business and Economics
Department of Decision Sciences and Information Management
Catholic University Leuven, Belgium

e-mail: firstname.lastname@econ.kuleuven.be

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 (ADS), 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. ADS may be found in healthcare, legal services, administration and many other service systems.

It is clear that an ADS 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 ADS. 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 ADS. We will refer to this combined model as the appointment-driven queueing system (ADQ). Using the ADQ, we assess: (1) the time a customer spends in the waiting list; (2) the time a customer spends waiting at the treatment 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 contributions of this article are twofold: (1) we present a new vacation model to model the AMQ; (2) we present a new model (the ADQ) to study an ADS 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 ADQ. Section 6 concludes.

2 Problem description

In this section we provide a detailed description of the dynamics at work at the ADS. First we define the problem setting. Next, we formally describe the basic concepts of the ADS. Finally we provide a traditional queueing analysis (based on the availability concept) and demonstrate that such an approach is unable to accurately assess the performance measures of an ADS.

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 beginning 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.

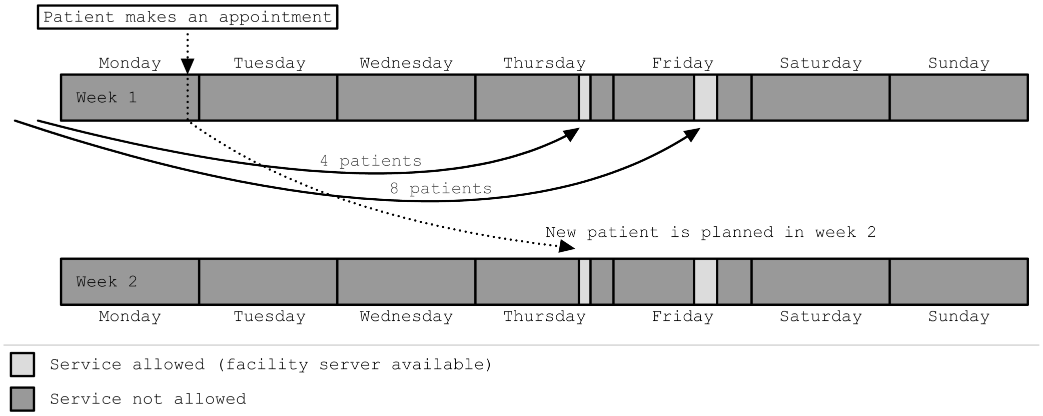

Figure 1: Scheduling of an appointment

- 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 treatment 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).

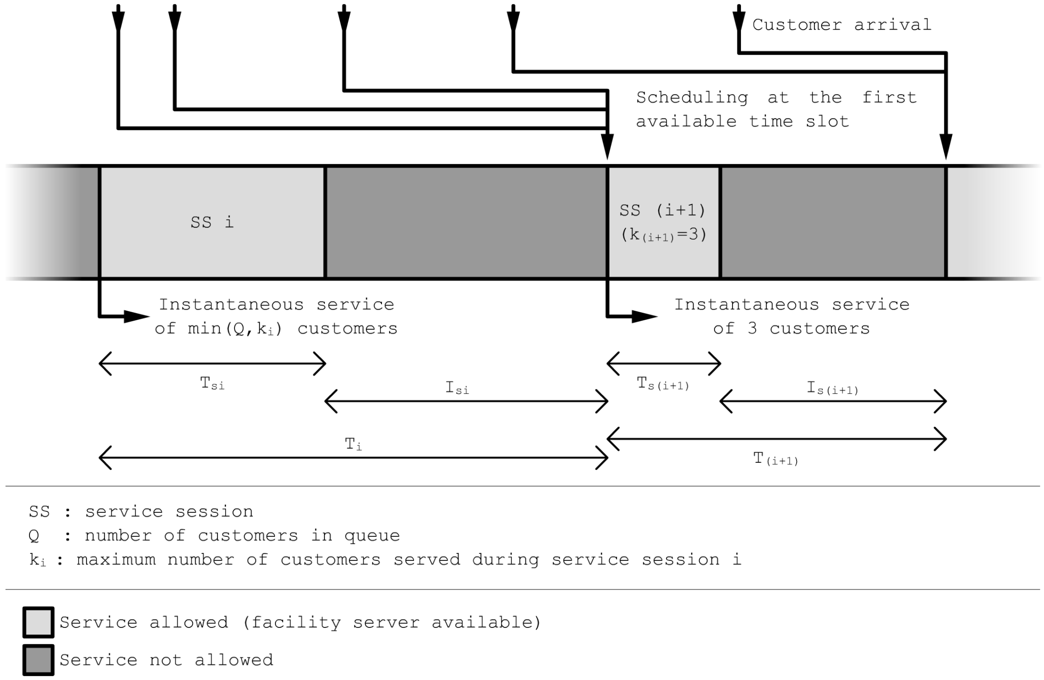
Customer arrival
Scheduling at the first available time slot
SS i
SS (i+1)
(k(i+1)=3)
Instantaneous service of min(Q,ki) customers
Instantaneous service of 3 customers
Tsi
Isi
Ts(i+1)
Is(i+1)
Ti
T(i+1)
SS : service session
Q : number of customers in queue
ki : maximum number of customers served during service session i
Service allowed (facility server available)
Service not allowed

Figure 2: The service process at an ADS

2.2 Problem definition

The service process at a ADS is a succession of service sessions during which customers are served at a single server. Each service session i (for the remainder of this text, 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 T_{s_i}; (3) the intersession time I_{s_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 ADS. 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 afternoon). A cycle of service sessions has length T and contains J service sessions j (for the remainder of this text, index j is defined as $j \in \{1, \ldots, J\}$). Remark 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

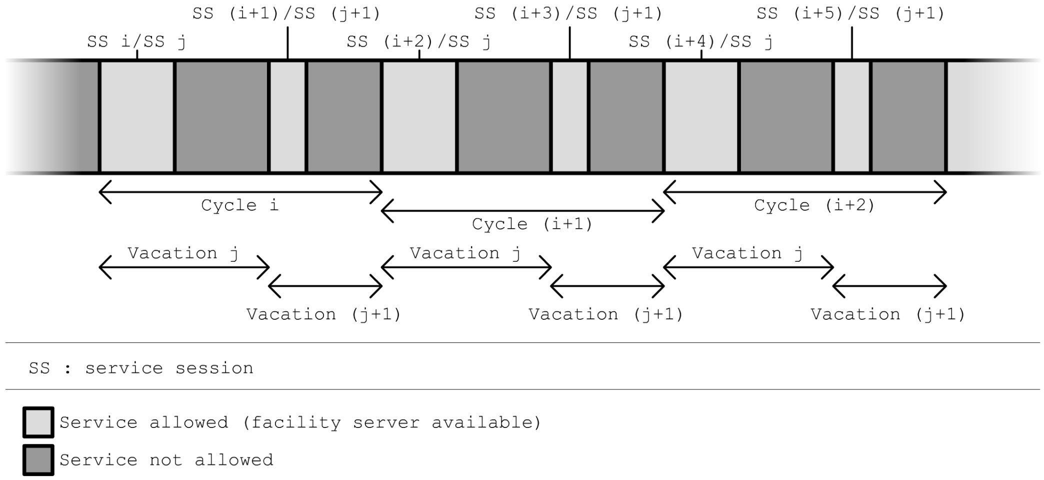

Figure 3: Succession of service cycles

with a vacation i of deterministic length $T_i = T_{s_i} + I_{s_i}$. We illustrate these dynamics in Figure 3. 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 minimize the variance of the resulting Erlang distribution of parameters V and υ_i.

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. Remark 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 beginning of a service session i, a maximum number of customers

k_i is removed from the waiting list. 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). Remark however that other AS can easily be implemented in the ADS. 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 mean and variance of the gamma distribution are given by

$$\frac{1}{\mu} = \alpha\theta, \tag{2}$$

$$\sigma_s^2 = \alpha\theta^2. \tag{3}$$

Remark that other distributions may also be implemented in the ADS. 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).

2.3 Traditional queueing approximation

In order to demonstrate that traditional queueing models do not suffice to accurately model an ADS, we provide the following example. The example builds on the setting discussed in section 2.1. 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

$$\rho = \lambda \sum_{j=1}^{J} \frac{T_j}{k_j}. \tag{4}$$

Remark 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 $C_s^2 = {}^{2}/_{3}$.

These are the input parameters required to assess a traditional queueing model. Such queueing models however, assume service to take place in a time continuum (i.e. 24 hours per day, 7 days per week). In our problem setting service is not a seamless concatenation of events, but is divided into separate, predefined service sessions. Therefore we need to rescale (i.e. inflate) the service process in order to fit a time continuum. We use the availability concept in order to rescale all service times (for a detailed account on the concept of availability see Lambrecht et al. (1998), Hopp and Spearman (2000) and Creemers and Lambrecht (2007)). The availability A of a workstation serves as a rescaling factor. The availability of a workstation is computed as the fraction of time that is available for service:

$$A = \sum_{j=1}^{J} \frac{T_{s_j}}{T_j}. \tag{5}$$

In our example the availability of the doctor's office is given by $A = {}^{1}/_{28}$. Using the availability concept we obtain the average rescaled, effective service time

$$\frac{1}{\mu_e} = \frac{1}{A\mu}. \tag{6}$$

With respect to the variance of the effective service times, one has

$$\sigma_{s_e}^2 = \frac{\sigma_s^2}{A^2}. \tag{7}$$

Reverting to our example, ${}^{1}/_{\mu_e} = 840$ minutes and $\sigma_{s_e^2} = 470,400$. The squared coefficient of variation C_s^2 is unaffected by the rescaling process.

Table 1: Simulation results of performance measures at the ADS

$E\left[W_{AMQ}\right]$	$E\left[W_{SFQ}\right]$	π_o	${}^{1}/_{\mu_o}$
4,281.3099	106.8222	0.1852	9.4721

We have defined all parameters required to approximate the total expected waiting time of a patient (i.e. the time from the making of an appointment until the start of service). We assess the total expected waiting time using the well-known Kingman equation (Kingman 1962)

$$E\left[W\right] = \left(\frac{C_a^2 + C_s^2}{2}\right)\left(\frac{\rho}{1-\rho}\right)\frac{1}{\mu_e}. \tag{8}$$

Using the Kingman equation, the total expected waiting time in the system amounts to 1,400 minutes (0.9722 days).

These results were validated using a simulation study. We constructed a simulation model that simulates the queueing behavior of patients as is observed in reality. The service and interarrival time distributions mentioned earlier were used to draw i.i.d. service and interarrival times of patients. Service takes place each Thursday from 6 PM until 8 PM and on Friday from 2 PM until 6 PM. Patients are allowed to make appointments at any time. If they make an appointment, they enter a queue outside the doctor's office (i.e. the waiting list). At the beginning of each service session, a number of patients is removed from the waiting list and is allowed to enter the doctor's office (on Thursday a maximum of 4 patients are selected while on Friday 8 patients are allowed). Before returning home, the doctor always treats all patients present in the doctor's office, even if this means that he has to work overtime. After treatment, patients leave the system.

Using the simulation, we keep track of: (1) $E\left[W_{AMQ}\right]$, the waiting time of a patient at the waiting list; (2) $E\left[W_{SFQ}\right]$, the waiting time of a patient at the service facility itself; (3) $\pi_o\left(j\right)$ the probability of the doctor to work overtime during a service session j as well as π_o, the overall probability of the doctor to work overtime; (4) ${}^{1}/_{\mu_o(j)}$, the average amount of overtime performed during a service session j as well as ${}^{1}/_{\mu_o}$, the overall expected overtime performed. We summarize the results of the simulation in Table 1. One may observe that the total expected waiting time of a patient (including the time spent in the waiting list as well as the time spent at the doctor's office) amounts to 4,388 minutes (3.0473 days). This result differs significantly from the result obtained using traditional queueing models. In addition, traditional queueing models are unable to assess performance measures associated with server overtime. It is clear that a new methodology is required to accurately assess the dynamics at work in an ADS.

3 Appointment making queueing system

In this section we develop the AMQ. First we provide a brief overview on literature of vacation models. Next we give a problem definition and finally we present the model itself.

3.1 Vacation model literature review

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) and Takagi (1988). A more recent yet less general survey can be found in Fiems (2004) and Vishnevskii and Semenova (2006).

The policy deciding on when the server leaves on a vacation is called a service discipline. Vacation models may operate under different service disciplines or combinations thereof. The most important disciplines reported in literature are listed below (refer to Fuhrman and Cooper (1985), Fiems (2004) and Vishnevskii and Semenova (2006)):

- Exhaustive service. The server leaves on a vacation if and only if all customers in the queue have been served.

- Gated service. The server serves only those customers that were present in the queue at the beginning of the current service session.

- k-limited service. The server serves customers in a queue until either k customers have been served or the queue becomes empty. For large k, the vacation model behaves as if operating under exhaustive service. Often k is assumed to equal unity (Takagi 1991).

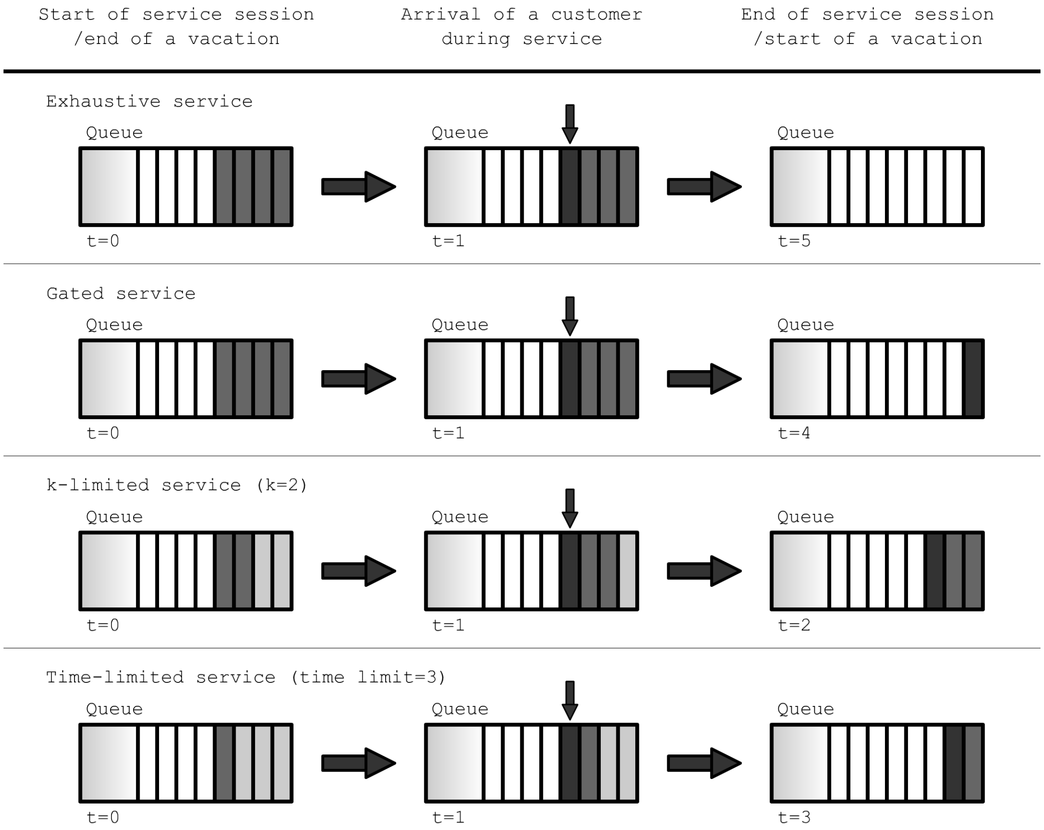

Figure 4: Illustration of the queuing process in a single service session under different service disciplines

- Time-limited service. Time-limited systems initialize a timer at the beginning of a service service session. Whenever the timer expires or when no more customers are present in the queue, a vacation is initiated.

Figure 4 provides an illustration of these service disciplines.
With respect to the vacation itself, numerous possibilities arise. Vacation models may have single vacations or multiple vacations. Multiple vacation models assume that when the server returns from a vacation to find the queue not empty, the vacation period ends; otherwise a new vacation is initiated (this process repeats itself until the server finds at least one customer in the

queue upon returning from vacation). Single vacation models on the other hand presume that when the server returns from a vacation, the vacation period ends, irrespective of the status of the queue (i.e. it is possible for a server to become "idle"). The vacation length depends on the vacation policy applied (Kleinrock (1975) and Tijms (2003)). We limit ourselves to the T-policy in which the server is activated T time units after the server left on vacation. Under a T-policy, vacations are often assumed to have an exponential duration (see for instance Gray et al. (2000)).

A wide variety of possible combinations of service and arrival distributions have been presented in literature. Most vacation models assume a Poisson arrival process and general service times (Stidham 2002). However, some research has also been performed on Markovian arrivals (MAP) and batch arrivals (BMAP). Models featuring such arrival processes have been proposed by Niu et al. (2003) among others. Bulk service has been considered by Katayama and Kobayashi (2003).

3.2 Problem definition

The AMQ consists of a single queue and a single virtual server which instantaneously can serve any number of customers (i.e. bulk service). 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, a maximum of k_i customers is served at the virtual server of the AMQ. 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 $T_i = T_{s_i} + I_{s_i}$). 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 may be modeled as a bulk service vacation model featuring: (1) a gated, k-limited service discipline (also referred to as a G-limited service discipline (de Souzae Silva et al. 1995)); (2) vacations of deterministic length; (3) multiple vacations; (4) state-dependent values of k as well as state-dependent vacation lengths. To the best of our knowledge, no such model exists in published literature.

Due to their analytical intractability, models featuring limited service disciplines (k-limited and time-limited service disciplines) are only scarcely dealt with (de Souzae Silva et al. (1995) and Borst et al. (1995)). Most of the relevant studies present approximative results or impose restrictive assumptions on either the maximum value of k or the distribution of arrivals, services and/or vacation lengths (refer to Leung and Eisenberg (1990), de Souzae Silva et al. (1995), Rubin and Wu (1995) and Katayama (2001)). There exists no research on vacation models in which k depends on the state of the system.

With respect to vacation length, general and phase-type distributions have been considered in literature (refer to Takagi (1994)). For those relevant models reported (Yijun and Quanlin (1996) and Shin and Pearce (1998)), the length of state-dependent vacations depends on the number of customers in the system at the beginning of a vacation. The AMQ however, requires the vacation length to depend on the current time in the system (i.e. a vacation is initiated at the start of a service session). No research on vacation models featuring time-dependent vacations is currently available.

3.3 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 a maximum of k_φ 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)

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

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

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

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

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)

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

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

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

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

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

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

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

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

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

$$\delta_{s,t,w} = \begin{cases} 1 & \text{if } (s-t) = k_w;\ \forall t > 0, \\ 1 & \text{if } s \leq k_w;\ \forall t = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

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 or matrix analytical methodology (MAM). MAM has been studied for several decades and has attracted the attention of many researchers in the queueing field. For an overview of literature and an introduction to MAM, refer to Latouche and Ramaswami (1999), Riska (2002), Osogami (2005) and Bini et al. (2006) among others. In short, MAM allows 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 auxilliary matrix $\mathbf{R}$. $\mathbf{R}$ may be obtained as the solution of the quadratic equation (Riska and Smirni 2002)

$$\mathbf{F} + \mathbf{R} \cdot \mathbf{L} + \mathbf{R}^2 \cdot \mathbf{B} = \mathbf{0}. \tag{10}$$

The stationary distribution π may be obtained by solving the following system of linear equations (Riska and Smirni 2002)

$$\left[\pi^{(0)}, \pi^{(1)}\right] \cdot \left[\begin{array}{ccc} \mathbf{e} & \mathbf{L}^{\diamond} & \mathbf{F} \\ (\mathbf{I} - \mathbf{R})^{-1} \cdot \mathbf{e} & \mathbf{B}^{\diamond} & \mathbf{L} + \mathbf{R} \cdot \mathbf{B} \end{array} \right] = [1, \mathbf{0}], \tag{11}$$

and through the recursive relationship

$$\pi^{(i)} = \pi^{(1)} \cdot \mathbf{R}^{i-1};\ \forall i \geq 1. \tag{12}$$

where:

- $\pi^{(Q)}$ is the vector of stationary probabilities associated with a queue size Q (i.e. given a queue size Q, $\pi^{(Q)}$ holds the stationary distribution of states (Q, j, v); $j \in \{1, 2, \ldots, J\}$; $v \in \{1, 2, \ldots, V + 1\}$).

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

- $\mathbf{e}$ is a vector of ones of appropriate size.

The symbol $\diamond$ indicates that an arbitrary column of the corresponding matrix may be discarded (while a column was added to represent the normalization condition).

Let $\pi\left(Q, j, v\right)$ denote 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 in queue at the start of 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 in queue at the start of a service session φ is associated with the stationary probability of states $(Q, j, V + 1)$. After rescaling these stationary probabilities, we obtain

$$\pi_{SFQ,\varphi}\left(Q\right) = \frac{\pi\left(Q, j, V+1\right)}{\sum\limits_{Q=0}^{\infty} \pi\left(Q, j, V+1\right)}. \tag{13}$$

In fact, $\pi_{SFQ,\varphi}\left(Q\right)$ may be used to determine the probability of having $min(Q, k_j)$ customers at the SFQ during a service session of type φ.

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

$$\pi_{AMQ,j}\left(Q\right) = \sum_{v=1}^{V} \frac{\pi\left(Q, j, v\right)}{\sum\limits_{Q=0}^{\infty} \sum\limits_{v^*=1}^{V} \pi\left(Q, j, v^*\right)}. \tag{14}$$

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

$$\overline{Q}_{AMQ,j} = \sum_{Q=0}^{\infty} Q\pi_{AMQ,j}\left(Q\right). \tag{15}$$

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

$$p_j = \frac{T_j}{\sum\limits_{j=1}^{J} T_j}. \tag{16}$$

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

$$\overline{Q}_{AMQ} = \sum_{j=1}^{J} p_j \overline{Q}_{AMQ,j}. \tag{17}$$

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

$$E\left[W_{AMQ}\right] = \frac{\overline{Q}_{AMQ}}{\lambda}. \tag{18}$$

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 and Veral (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 and Veral (2003). Examples of environmental variables include customer unpunctuality, the number of customer classes, the number of servers, ...

In AS literature, customers are either scheduled using some appointment scheduling rule (ASR) or a procedure is developed to determine the (optimal) arrival times of customers at the service facility (examples of the latter category may be found with Weiss (1990), Liao et al. (1993), Wang (1997),

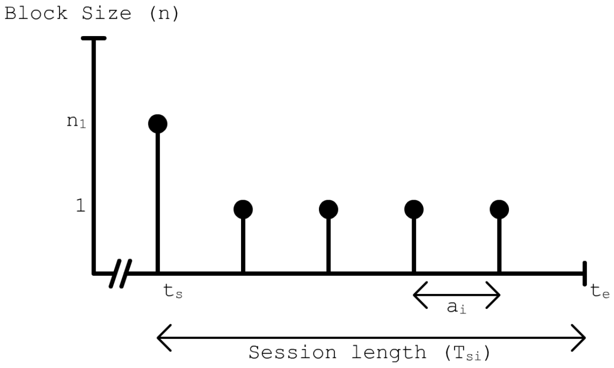

Figure 5: Appointment scheduling rules

Vanden Bosch and Dietz (2000) and Vanden Bosch and Dietz (2001) among others). With respect to ASR, comprehensive comparisons of various ASR are available with Ho and Lau (1992), Ho and Lau (1999) and Mondschein and Weintraub (2003). In the remainder of this work, we will focus only on ASR.
ASR 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,
- appointment interval (a_{i_l}); indicating the interval between two successive appointments during service session i.

Remark 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 T_{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; $b \in \{1, 2, \dots, B\}$, a number of customers n_b is scheduled to arrive. Figure 5 provides further insight. Many ASR 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 ASR is referred to as the Bailey-Welch rule. Another popular ASR is the block appointment rule in which all customers are assigned to arrive in the initial block (thereby minimizing server idle time but effectively maximizing customer waiting time). 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

$$O_j = \frac{k_j}{\mu}. \tag{19}$$

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) are one of the few researchers to 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 M 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 M) is given by

$$E\left[W_{SFQ,j,M}\right] = \frac{M-1}{2\mu}. \tag{20}$$

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

$$f\left(x,\alpha,\theta\right) = x^{\alpha-1}\frac{e^{\frac{-x}{\theta}}}{\Gamma\left(\alpha\right)\theta^{\alpha}}. \tag{21}$$

The Laplace transform of $f\left(x,\alpha,\theta\right)$ is easily obtained (where z is the image of x)

$$\mathcal{L}\left\{f\left(x,\alpha,\theta\right)\right\} = \theta^{-\alpha}\left(z+\frac{1}{\theta}\right)^{-\alpha}. \tag{22}$$

The convolution $\left(f*f\right)\left(x,\alpha,\theta\right)$ is given by

$$\mathcal{L}\left\{\left(f*f\right)\left(x,\alpha,\theta\right)\right\} = \left[\theta^{-\alpha}\left(z+\frac{1}{\theta}\right)^{-\alpha}\right]^{2}. \tag{23}$$

The inverse Laplace transform yields

$$\left(f*f\right)\left(x,\alpha,\theta\right) = x^{2\alpha-1}\frac{e^{\frac{-x}{\theta}}}{\Gamma\left(2\alpha\right)\theta^{2\alpha}}. \tag{24}$$

Analogously, one can show that the convolution of M gamma distributions of parameter α and θ may be defined as

$$f\left(x,M\alpha,\theta\right) = x^{M\alpha-1}\frac{e^{\frac{-x}{\theta}}}{\Gamma\left(M\alpha\right)\theta^{M\alpha}}. \tag{25}$$

Which corresponds to the pdf of a gamma distribution of parameters $M\alpha$ and θ (i.e. the distribution of the total service time of M customers). The cumulative distribution function (cdf) is given by

$$F\left(x, M\alpha, \theta\right) = \frac{\gamma\left(M\alpha, {}^{x}\!/\!{}_{\theta}\right)}{\Gamma\left(M\alpha\right)}. \tag{26}$$

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 M customers require service

$$\pi_o\left(j, M\right) = 1 - F\left(O_j, M\alpha, \theta\right). \tag{27}$$

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

$$\frac{1}{\mu_o\left(j, M\right)} = \int\limits_{O_j}^{\infty} \left(x - O_j\right) f\left(x, M\alpha, \theta\right) dx. \tag{28}$$

Which can be simplified to the following closed form formula

$$\frac{1}{\mu_o\left(j, M\right)} = \frac{\left[-O_j\gamma\left(M\alpha, {}^{O_j}\!/\!{}_{\theta}\right)\right] + \left[O_j^{M\alpha}\left(\frac{O_j}{\theta}\right)^{-M\alpha}\theta^{1-M\alpha}\gamma\left(1+M\alpha, {}^{O_j}\!/\!{}_{\theta}\right)\right]}{\Gamma\left(M\alpha\right)}. \tag{29}$$

5 Appointment driven queueing system

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

5.1 The ADQ model

From the AMQ we have obtained $\pi_{SFQ,j}\left(M\right)$, the distribution of the number of customers in queue at the start of a service session of type j. We can use this distribution to determine the probability of having to serve M patients at a service session j. In turn, this information can be used at the SFQ to obtain the measures of interest (average customer waiting time at the service facility, probability of server overtime and the expected amount of overtime performed). Using the stationary distribution $\pi_{SFQ,j}\left(M\right)$ as a weighing factor for the results obtained at the SFQ corresponding to M customers served at a service session j, we obtain general results at the ADQ.

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

$$\begin{aligned} E\left[W_{SFQ,j}\right] &= \left(\sum_{M=0}^{k_j-1}\pi_{SFQ,j}\left(M\right)E\left[W_{SFQ,j,M}\right]\right)+ \\ &\quad+\left(E\left[W_{SFQ,j,k_j}\right]\sum_{M=k_j}^{\infty}\pi_{SFQ,j}\left(M\right)\right). \end{aligned} \tag{30}$$

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

$$\overline{Q}_{SFQ,j}=\left(\sum_{M=0}^{k_j-1}\pi_{SFQ,j}\left(M\right)M\right)+\left(k_j\sum_{M=k_j}^{\infty}\pi_{SFQ,j}\left(M\right)\right). \tag{31}$$

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

$$E\left[W_{SFQ}\right]=\sum_{j=1}^{J}\frac{E\left[W_{SFQ,j}\right]\overline{Q}_{SFQ,j}}{\sum\limits_{j^*=1}^{J}\overline{Q}_{SFQ,j^*}}. \tag{32}$$

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

$$\pi_o\left(j\right)=\left(\sum_{M=0}^{k_j-1}\pi_{SFQ,j}\left(M\right)\pi_o\left(j,M\right)\right)+\left(\pi_o\left(j,k_j\right)\sum_{M=k_j}^{\infty}\pi_{SFQ,j}\left(M\right)\right). \tag{33}$$

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

$$q_j=\frac{1}{J}. \tag{34}$$

Therefore the overall probability of the server to work overtime is given by

$$\pi_o = \sum_{j=1}^{J} q_j \pi_o\left(j\right). \tag{35}$$

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

$$\frac{1}{\mu_o\left(j\right)} = \left(\sum_{M=0}^{k_j-1} \pi_{SFQ,j}\left(M\right) \frac{1}{\mu_o\left(j, M\right)}\right) + \left(\frac{1}{\mu_o\left(j, k_j\right)} \sum_{M=k_j}^{\infty} \pi_{SFQ,j}\left(M\right)\right). \tag{36}$$

The overall expected amount of overtime performed at the server equals

$$\frac{1}{\mu_o} = \sum_{j=1}^{J} q_j \frac{1}{\mu_o\left(j\right)}. \tag{37}$$

The overall expected waiting time at the AMQ is given in equation 18. Together with equation 32, 35 and 37, all performance measures of interest at the ADS are defined.

5.2 Numerical example

In this section we revisit the example discussed in section 2.1 and 2.3. However, at this point, we will use the ADQ 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 Tables 2 and 3. One can observe that the ADQ is able to provide very accurate results when assessing strategic performance measures at the ADS. 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 five service sessions overtime is performed. The overall 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

Table 2: Model results with varying number of vacation phases

V	$E\left[W_{AMQ}\right]$	$E\left[W_{SFQ}\right]$	π_o	$1/\mu_o$
10	5,126.9400	105.9466	0.1979	10.0794
50	4,440.3660	106.5436	0.1882	9.5996
100	4,360.2300	106.6709	0.1868	9.5329
200	4,320.7920	106.7410	0.1861	9.4991

Table 3: Simultation results with varying number of vacation phases

V	$E\left[W_{AMQ}\right]$	$E\left[W_{SFQ}\right]$	π_o	$1/\mu_o$
10	5,125.0903	105.9409	0.1975	10.0804
50	4,440.3269	106.5627	0.1879	9.6094
100	4,359.7361	106.6793	0.1864	9.5327
200	4,320.3242	106.7557	0.1859	9.5117
∞	4,281.3099	106.8222	0.1852	9.4721

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 ADQ developed in this article, provide the tools to construct such an optimization procedure.

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 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 up to the task 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 remarked that other appointment systems can easily 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.

References

Bini, D., B. Meini, S. S. and Van Houdt, B.: 2006, Structured markov chains solver: algorithms, *Proc. of SMCtools*, ACM Press, Pisa (Italy).

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

Borst, S., Boxma, O. and Levy, H.: 1995, The use of service limits for efficient operation of multistation single-medium communication systems, *IEEE/ACM Transactions on Networking* **3**, 602–612.

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

Creemers, S. and Lambrecht, M.: 2007, Modeling a healthcare system as a queueing network: the case of a Belgian hospital, *Technical Report 0710*, Department of Decision Sciences & Information Management, Research Center for Operations Management, Katholieke Universiteit Leuven.

de Souzae Silva, E., Gail, H. and Muntz, R.: 1995, Polling systems with server timeouts and their application to token passing networks, *IEEE/ACM Transactions on Networking* **3**, 560–575.

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

Fiems, D.: 2004, *Analysis of discrete-time queueing systems with vacations*, PhD thesis, Ghent University.

Fuhrman, S. and Cooper, R.: 1985, Stochastic decompositions in the M/G/1 queue with generalized vacations, *Operations Research* **33**, 1117–1129.

Gray, W., Wang, P. and Scott, M.: 2000, A vacation queueing model with service breakdowns, *Applied Mathematical Modelling* **24**, 391–400.

Ho, C. and Lau, H.: 1992, Minimizing total cost in scheduling outpatient appointments, *Management science* **38**, 1750–1764.

Ho, C. and 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.

Hopp, W. and Spearman, L.: 2000, *Factory Physics*, 2 edn, McGraw-Hill Higher Education, New York.

Katayama, T.: 2001, Waiting time analysis for a queueing system with time-limited service and exponential timer, *Naval Research Logistics* **48**, 638–651.

Katayama, T. and Kobayashi, K.: 2003, Sojourn time analysis of a two-phase queueing system with exhaustive batch-service and its vacation model, *Mathematical and Computer Modelling* **38**, 1283–1291.

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

Kingman, J.: 1962, On queues in heavy traffic, *Journal of the Royal Statistical Society. Series B (Methodological)* **24**, 383–392.

Kleinrock, L.: 1975, *Queueing systems - Volume I: theory*, John Wiley & Sons, New York.

Lambrecht, M., Ivens, P. and Vandaele, N.: 1998, Aclips: a capacity and lead time integrated procedure for scheduling, *Management Science* **44**, 1548–1561.

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

Latouche, G. and Ramaswami, V.: 1999, *Introduction to Matrix Analytic Methods in Stochastic Modeling*, 1 edn, ASA-SIAM Series on Statistics and Applied Probability, Philadelphia PA.

Leung, K. and Eisenberg, M.: 1990, A single-server queue with vacations and gated time-limited service, *IEEE Transactions on Communcications* **38**, 1454–1462.

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

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

Niu, Z., Shu, T. and Takahashi, Y.: 2003, A vacation queue with setup and close-down times and batch Markovian arrival processes, *Performance Evaluation* **54**, 225–248.

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.

Riska, A.: 2002, *Aggregate matrix analytic techniques and their applications*, PhD thesis, College of William and Mary, Williamsburg, VA.

Riska, A. and Smirni, E.: 2002, *M/G/1-type processes: A tutorial*, Springer-Verlag, New York, pp. 36–63. In M. Calzarossa and S. Tucci: Performance Evaluation of Complex Computer Systems; Techniques and Tools.

Rubin, I. and Wu, J.: 1995, Analysis of an M/G/1/N queue with vacations and its iterative application to FDDI timed-token rings, *IEEE/ACM Transactions on Networking* **3**, 842–856.

Shin, Y. and Pearce, C.: 1998, The BMAP/G/1 vacation queue with queue-length dependent vacation schedule, *Journal of the Australian Mathematical Society - Series B* **40**, 207–221.

Stidham, S.: 2002, Analysis, design and control of queueing systems, *Operations Research* **50**, 197–216.

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

Takagi, H.: 1991, Analysis of finite-capacity polling systems, *Advances in Applied Probability* **23**, 373–387.

Takagi, H.: 1994, M/G/1//N queues with server vacations and exhaustive service, *Operations Research* **42**, 926–939.

Tijms, H.: 2003, *A first course in stochastic models*, John Wiley & Sons, New York.

Vanden Bosch, P. and Dietz, D.: 2000, Minimizing expected waiting in a medical appointment system, *IIE Transactions* **32**, 841–848.

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

Vishnevskii, V. and Semenova, O.: 2006, Mathematical methods to study the polling systems, *Automation and Remote Control* **2**, 3–56.

Wang, P.: 1997, Optimally scheduling N customer arrival times for a single-server system, *Computers and operations research* **24**, 703–716.

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

Yijun, Z. and Quanlin, L.: 1996, Analysis of a two-stage cyclic queue with state-dependent vacation policy, *Optimization* **36**, 75–91.