doi:10.1016/j.ejor.2018.10.036 • www.stefancreemers.be • info@stefancreemers.be

Evaluation of appointment scheduling rules: A multi-performance measurement approach

Stefan Creemers
Marc Lambrecht
Jeroen Beliën
Maud Van den Broeke

Abstract - **Appointment scheduling rules are used to determine when a customer is to receive service during a service session. In general, appointment scheduling rules do not consider the sequencing of individual customers, but provide simple guidelines on how to assign appointment times to a set of (arriving) customers. Many appointment scheduling rules exist and are being used in practice (e.g., in healthcare and legal services). Which appointment scheduling rule is best, however, is still an open question. In order to answer this question, we develop an analytical model that allows to assess the performance (in terms of customer waiting time, server idle time, and server overtime) of appointment scheduling rules in a wide variety of settings. More specifically, the model takes into account: (1) customer unpunctuality, (2) no-shows, (3) service interruptions, and (4) delay in session start time. In addition, we allow the use of general distributions to capture system processes. We adopt an efficient algorithm (with respect to computational and memory requirements) to assess the performance of 314 scheduling rules and use data envelopment analysis to identify the rules that have good, robust performance in a wide variety of settings.**

Keywords - **OR in health services, appointment scheduling rules, Markov chain, data envelopment analysis**

1 Introduction

Professionals in health care and other services face the problem of allocating time windows to customers. This allocation can be done by means of Appointment Scheduling Rules (ASRs). ASRs determine when a customer is to receive service during a service session. Although the literature on ASRs is mainly focused on health care (e.g., in an emergency department, a doctor's office, or an operating theater), the research topic is generic and applicable in many industries: attorneys, faculty receiving students, tax accountants, consultants, barbers, service centers, trailers at receiving bays, and many others.

With increasing customer expectations on being served quickly, both in health care and in other service industries, timeliness of appointments is crucial (Grote, Newman, & Sutaria, 2007; Hall, 2012).Moreover, timely delivery of care has been shown to reduce mortality and morbidity associated with a variety of medical conditions (Smart & Titus, 2011). The customer waiting time consequently is a relevant performance measure. A second important

objective of appointment scheduling has to do with the efficiency of the service. For private companies, the impetus to efficiency comes naturally. However, health care systems are under pressure to use their capacity effectively and efficiently (Hall & Partyka, 2016). Doctors' (or more general servers') idle time and overtime are hereby important performance measures. In addition, a distinction needs to be made between private healthcare institutions (who may focus more on patient waiting time) and public care systems (where taxpayer money needs to be spent as efficiently as possible).

The literature on appointment system design can broadly be divided into two classes. In the first class the objective is to give individual patients a fixed appointment time, that means, we have to decide on the order the appointments will be scheduled. It comes down to a sequencing decision. Patient class information will be used for sequencing purposes. In the second class of studies, authors try to find the best appointment rule. Here we have to decide on the length of the appointment intervals (fixed or variable) and the block sizes (number of patients scheduled to each appointment slot, individual or multiple). In this case, patient information such as no-shows, patient unpunctuality and other disrupting factors are taken care of by adjusting the appointment intervals and/or the block sizes. In this article, we focus on the second class, namely on ASRs but in a complex environment characterized by unpunctuality, no-shows, and server lateness. In literature there is a debate on the question which approach is best, sequencing or appointment rules. We opt for appointment rules because we believe that ASRs will perform more robustly in complex environments. Cayirli, Veral, & Rosen (2008), who combine both approaches, clearly mention that sequencing-based appointment systems are less flexible than those that assign patients on a first-call, first-appointment basis. They continue the argument by stating that the biggest challenge for future research will be to find new appointment systems that will perform more robustly across different clinical environments and patient panels. We position our paper in the ASR literature, and focus on identifying good ASRs that have robust performance in a variety of settings. Of course, we fully appreciate the work done by authors focusing on sequencing. We refer to the excellent paper of Deceuninck, Fiems, & De Vuyst (2018). Their approach allows to take prior individual knowledge about the patients into account. If such information is available and correctly exploited, the sequencing approach may lead to substantial cost reductions.

The objective of this article is to identify ASRs that simultaneously minimize customer waiting time, server idle time, and overtime. This has to be done in an environment where both demand and supply characteristics are highly uncertain, and subject to many sources of variability. For this purpose, we develop an analytical model to determine the best appointment policy under a wide range of assumptions. In contrast to most of the literature (e.g., Jerbi & Kamoun, 2011; Lee, Min, Ryu, & Yih, 2013), we do not rely on simulation but use a Discrete-Time Markov Chain (DTMC) to model the Appointment System (AS).

ASRs determine the planned (scheduled) arrival rate of customers during a service session. The actual arrival time may differ from the planned arrival time. Therefore, we assign each customer a probability of being too late or too early. In addition, we assign each customer a probability of not showing up. Because of the no-show problem (i.e., customers not showing up for their appointment), the actual number of customer arrivals is unknown, even if the number of customers per session is fixed and predetermined. The performance of ASRs is not only influenced by the arrival rate and service rate characteristics. Other types of

outages during the service session are also important. We therefore allow for delays at the start of a service session due to late arrival of a server, or due to setup activities at the start of a session. We also allow preemptive and non-preemptive interruptions during the service session (e.g., it is well known that scheduled appointments can be disrupted by emergencies). All these extensions allow us to model real-life appointment systems, and to identify ASRs that have a robust performance across different settings.

We develop an analytical model that uses an efficient (in terms of computational and memory requirements) algorithm to assess the performance of ASRs. The validity and accuracy of the model are supported by a simulation study. We use the model to assess the performance of a set of 314 ASRs in an elaborate computational experiment. To compare the performance of these ASRs (in terms of waiting time, idle time, and overtime), we apply Data Envelopment Analysis (DEA).

The contribution of this article is threefold:

1. We develop a new analytical model to assess the performance of an ASR in a general setting.

2. We perform an elaborate computational experiment to analyze the performance of a large number of ASRs. As such, we provide insight in what ASR is best in a given environment. This analysis is particularly useful for smaller general practices and hospitals that don't have the time/resources to optimize their appointment system. We also confirm, and unify, the findings of several other studies in the field of appointment scheduling.

3. We use DEA to identify the best ASR based on multiple performance measures. The use of DEA in appointment scheduling is novel and allows to overcome limitations of traditional approaches that require to fix subjective weights for different objectives (i.e., waiting time, idle time, and overtime). As a non-parametric method, DEA does not require to specify weights (i.e., it allows for a fair comparison of ASRs). DEA also offers tools to measure the robustness of the decision rules, we refer to the Maverick score that is used in this paper.

We also provide a number of important managerial insights. The first insight is that simple individual ASRs, like the Bailey-Welch rule, perform very well, certainly in the case where only a small number of customers needs to be scheduled. Secondly, we find that Variable Interval (VI) (or dome-shaped) ASRs are among the best performing ASRs and that their performance is robust over complex and dynamic environments. This means these VI ASRs are recommended in AS where the environmental variables are prone to change. The third important insight is that the relative performance of ASRs drastically changes when waiting time becomes more important. Thus, if customer waiting time is a crucial factor for success (and more relevant than overtime or idle time), managers may want to consider an ASR that results in smaller waiting times, even if this results in more idle time and/or overtime. Finally, our results suggest that it is not necessary to update good performing ASRs after a change in customer punctuality as we found that the performance of ASRs does not depend heavily on customer punctuality.

This article is organized as follows. Section 2 provides a description of the problem setting. The literature on appointment systems is discussed in Section 3. Section 4 defines the basic

processes that govern the appointment system, and Section 5 presents the basic model. The design and the results of the computational experiment are discussed in Section 6. Section 7 concludes.

2 Problem Description

ASRs are used to schedule the servicing of a given number of customers during a service session. Complexity is introduced in the form of so-called "environmental factors". An extensive overview of such environmental factors is provided in Cayirli & Veral (2003), Gupta & Denton (2008), and Ahmadi-Javid, Jalali, & Klassen (2017). In this article, we take the following factors into account:

- Customers are allowed to arrive early or late ("customer unpunctuality"), or may even fail to show up.
- Each customer has a unique arrival process characterized by (1) a probability to show up, (2) probabilities to arrive early or late, and (3) distributions to model the amount of time a customer arrives early or late.
- The start of a service session may be delayed due to the absence or lateness of staff, the setup of equipment, etc.
- The service process of a customer may be interrupted (e.g., a doctor who is called away for an emergency). We allow for both preemptive interrupts and non-preemptive interrupts.

In what follows, we assume that:

- Only one customer can be served at the same time (i.e., customers are served by a single server).
- Customers have i.i.d. service time distributions.
- All customers that arrive during the service session are served.
- Customers that arrive early (i.e., prior to their scheduled arrival time) receive service if the server is idle, note that this implies the possibility of overtaking other customers who arrive late.

Note that, in this study, we do not classify customers into distinct groups (see for instance Bhattacharjee & Ray, 2016; Cayirli & Yang, 2014; Sickinger & Kolisch, 2009).

We use an example to illustrate the dynamics of an ASR. Figure 1 provides visual support. Suppose we have a service session that starts at 12 AM. Assume we schedule customers using an ASR with an initial block of one customer. More specifically, we schedule the first customer to arrive at the start of the service session. The other customers are scheduled to arrive at 2 PM and at 4 PM, respectively. Their expected service time requirement is 2 hours. In the example, the first customer arrives on time, but the server is 15 minutes

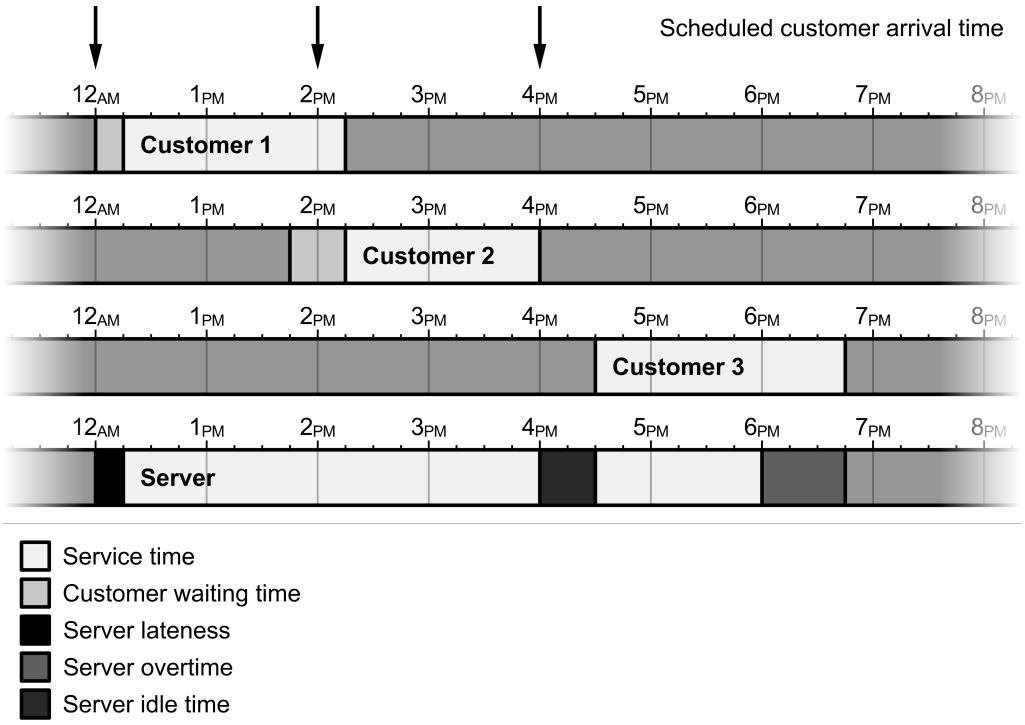

Figure 1: Dynamics of the appointment system.

late, resulting in 15 minutes waiting time for customer 1. Customer 2 arrives 15 minutes early. As the server is still serving customer 1, however, the waiting time for customer 2 is 30 minutes. The service time of customer 2 takes 15 minutes less than expected (i.e., it takes 1 hour and 45 minutes), which means serving customer 2 is finished at 4 PM. The third customer, expected at 4 PM, is 30 minutes late, leading to 30 minutes of server idle time (regular operating costs such as staff wages and equipment costs are still incurred). Because service starts immediately upon entry of the third customer, customer 3 has no waiting time. As such, the average waiting time of a customer amounts to 15 minutes. The service session of customer 3 takes 15 minutes longer than expected (i.e., 2 hours 15 minutes instead of 2 hours), resulting in an overall server overtime of 45 minutes (at which point additional costs such as penalties or staff compensation might be incurred). The duration of the service session then equals 6 hours and 45 minutes.

The performance measures of interest are: (1) the expected waiting time of a customer, (2) the expected amount of time that the server resides in an idle state, and (3) the expected amount of server overtime. Our model can provide these performance measures for any given schedule of customers (i.e., the outcome of any given ASR or scheduling procedure). We use DEA to identify the optimal set of weights attached to the aforementioned measures for each ASR individually. Moreover, we allow prioritization of certain performance measures (e.g., customer waiting time) by incorporating additional constraints in the DEA. This is also referred to as subjective valuation of performance measures.

3 Literature Review

Appointment systems have been studied extensively over the past 50 years. They arise in many contexts. In transportation, AS have been used to schedule the arrival of cargo ships and trucks at ports (Giuliano & O'Brien, 2007; Namboothiri & Erera, 2008; Sabria & Daganzo, 1989), to schedule railway operations (Lawley, Parmeshwaran, Richard, Turkcan, Dalal, & Ramcharan, 2008; Wendler, 2007), and to allocate airport slots (Madas & Zografos, 2006, 2008). AS have also been adopted in telecommunication networks to schedule data transmissions (Rose & Yates, 1995; van Leeuwaarden, Denteneer, & Resing, 2006). In manufacturing settings, AS have been used to schedule deliveries in just-in-time inventory systems (Liao, Pegden, & Rosenshine, 1993; Wang, 1993), to support lot-sizing decisions (Dellaert & Melo, 1998), and to schedule job release times (Biskup, Herrmann, & Gupta, 2008; Yan & Lai, 2007). The bulk ofthe AS literature, however, deals with the scheduling of patients in a healthcare context. Excellent overviews of relevant literature may be found with Mondschein & Weintraub (2003), Cayirli & Veral (2003), Gupta & Denton (2008), and Ahmadi-Javid et al. (2017).

Nearly all of the literature on AS deals with the scheduling of customers during a single service session. Studies observing AS ranging over multiple service sessions are rather scarce. In Vanden Bosch & Dietz (2000), customers are scheduled over several days using a heuristic approach. The computational complexity involved limited applicability of their model to settings in which only a small number of customers can be scheduled. Creemers & Lambrecht (2009b, 2010) analyze appointment-driven systems, and observe the queueing behavior of a customer from the making of an appointment until the start of the service session in which the customer will receive service.

It is known that no-shows have a dire impact on the performance of an AS (Alaeddini, Yang, Reeves, & Chandan, 2015; Cayirli, Veral, & Rosen, 2006; Cayirli, Yang, & Quek, 2012; Green, 2014; Gupta & Denton, 2008; Ho & Lau, 1992; Zacharias & Pinedo, 2012). As such, all but a few studies incorporate the possibility of customer no-shows.

The modeling of customer unpunctuality is less prevalent. Relevant literature includes Mercer (1960), Blanco White & Pike (1964), Fetter & Thompson (1966), Mercer (1973), Vissers (1979), Sabria & Daganzo (1989), and Wang (1993). Most of these models only allow for the late arrival of customers. Recent models that take into account early as well as late customer arrivals include Schuetz & Kolisch (2012), Tai & Williams (2012), Klassen & Yoogalingam (2014), and Samorani & Ganguly (2016). While our model allows for individual unpunctuality probabilities, all earlier studies assume customer unpunctuality to be homogeneous (i.e., independent from the scheduled arrival times and patient characteristics).

Staff lateness (such that service cannot commence at the start of a service session) is considered in Blanco White & Pike (1964), Fetter & Thompson (1966), Vissers (1979), Babes & Sarma (1991), and Liu & Liu (1998a,b). More recent contributions that allow for physician lateness are found in Klassen & Yoogalingam (2013, 2014).

Server interruptions are modeled in Rising, Baron, & Averill (1973), Klassen & Yoogalingam (2013), and Luo, Kulkarni, & Serhan (2012).

Most AS literature assumes that customers are scheduled for arrival at discrete moments in time only. Individual ASRs assume a single customer to be scheduled at each of the discrete appointment times. Often, the time intervals between two such discrete appointment times

are assumed to be fixed. Such studies may be found with Bailey (1952), Welch (1964), Fetter & Thompson (1966), Klassen & Rohleder (1996), and Rohleder & Klassen (2000). When allowing for multiple initial appointments (i.e., as to minimize the server idle time at the beginning of a service session) individual ASRs with fixed intervals are observed in Bailey (1952), Blanco White & Pike (1964), Klassen & Rohleder (1996), and Ho & Lau (1999).

Ho & Lau (1992) introduce "variable interval" rules, also known as "offset" rules that allow for variable times in between two consecutive appointments. Many studies (a.o., Denton & Gupta, 2003; Kuiper & Mandjes, 2015; Robinson & Chen, 2003) conclude that optimal appointment intervals have a dome-shaped pattern (i.e., the length of an appointment interval gradually increases towards the middle of the session, after which it gradually decreases). When the length of the appointment intervals are integer, Klassen & Yoogalingam (2009) have found that a plateau-dome structure (i.e., the middle intervals have the same length; creating a plateau) leads to the best results. The optimal pattern of appointment intervals is shown to depend on service time variability (Chakraborty, Muthuraman, & Lawley, 2010; Erdogan & Denton, 2013), interruptions (Klassen & Yoogalingam, 2013; Luo et al., 2012), no-shows (Chakraborty et al., 2010; Erdogan & Denton, 2013), and customer unpunctuality (Klassen & Yoogalingam, 2014; Tai & Williams, 2012).

Block ASRs allow the scheduling of multiple customers at each of the discrete appointment times (i.e., during each of the "blocks"). In Blanco White & Pike (1964) and Soriano (1966), fixed block sizes (i.e., the number of appointments made at each of the discrete appointment times) as well as fixed block lengths (i.e., the time interval in between two successive discrete appointment times) are assumed. Variable block sizes and fixed intervals have been studied in Rising et al. (1973), Fries & Marathe (1981), Liao et al. (1993), and Liu & Liu (1998a,b). Fixed block sizes and variable intervals are analyzed in Pegden & Rosenshine (1990), Wang (1997), and Vanden Bosch & Dietz (2000).

As mentioned earlier, appointment systems may focus on the sequencing of customers during a single service session based on their individual characteristics. We refer to Cayirli et al. (2008) who use patient-related information for sequencing combined with interval adjustments. Deceuninck et al. (2018) introduce several types of unpunctuality, and develop an algorithm that optimizes a schedule with respect to customer waiting time, server idle time, and session overtime. Their analysis is based on a discrete-time queueing model (resulting in explicit expressions of each performance measure). Note that in our paper we also model the appointment system as a DTMC allowing us to obtain the same performance measures. The paper of Deceuninck et al. (2018) and our paper are clearly methodologically related (both use a discretization approach), but are quite different in specific modeling characteristics. Note that discretization not only allows the modeling of general service and/or arrival processes, it also solves the curse of dimensionality that plagues approaches that rely on phase-type distributions (see for instance Kuiper & Mandjes (2015); Wang (1993, 1997)). More specifically, the number of phases in a phase-time distribution can become very large, resulting in a Markov chain that can no longer be analyzed (e.g., for low-variability processes).

Only a limited number of studies allow customers to have distinct service requirements. Most of these studies do not only optimize the scheduling of customers, but also the sequence of customers to be served (Cardoen, Demeulemeester, & Beliën, 2009; Klassen & Rohleder, 1996; Rohleder & Klassen, 2000; Vanden Bosch & Dietz, 2000, 2001).

Optimization of customer appointment times usually occurs over some subset of: (1) customer waiting time, (2) server idle time and, (3) server overtime. Most of the research observes either server idle time or server overtime. Surprisingly few studies assess the trade-off between all three performance measures. Well-established multidimensional performance techniques, however, exist. DEA, for instance, provides a means to perform a multidimensional performance analysis based on mathematical optimization (see Cook & Seiford (2009) for an overview of the DEA literature). Fries & Marathe (1981) and Kaandorp & Koole (2007) take all three performance measures into account, however, they do not use an objective technique. In order to deal with multiple performance measures, Ho & Lau (1992) adopt a frontier approach that can be considered as a simplification of a DEA (Cook & Seiford, 2009).

Ho & Lau (1992, 1999) examine 50 scheduling rules under various environmental factors (such as the probability of no-shows, the number of patients per session, etc.). In this article, we extend the work of Ho and Lau by (1) examining more scheduling rules, (2) allowing more realistic operating environments (3) using analytical methods to obtain performance measures, and (4) including server overtime as a third objective next to customer waiting time and server idle time. Our operating environment is more realistic as we allow for customer unpunctuality, service process interruptions, and session start delays.

Cayirli et al. (2006) extend Ho and Lau's assessment study by incorporating a wider set of environmental factors, such as walk-ins, no-shows, and punctuality. Lian, DiStefano, Shields, Heinichen, Giampietri, & Wang (2010) also examine a large number of appointment configurations, that differ with respect to the number of total appointment requests, service time distribution, time slot size, and co-operation ratio (reflecting the degree of mutual preferences between patient and provider). In contrast to our work, their study examines the impact of avoiding schedule defragmentation while ignoring no-shows, customer unpunctuality, and server interrupts.

4 Definitions

In this section, we classify the different ASRs considered in our study. In addition, we define the basic processes that govern the AS, and introduce a discretization procedure that allows us to obtain the discrete distributions of customer service and arrival times. These discrete distributions are used in the DTMC that is used to model the AS.

4.1 Classification of appointment scheduling rules

Most ASRs may be classified in terms of:

- A_i, the scheduled arrival time of customer i,
- μ^{-1}, the mean service time of a customer,
- σ_i, the standard deviation of the service time requirement of customer i,
- N, the number of customers that require scheduling, where customer i is in $\{0, 1, \ldots, N-1\}$.

We implement a set of 314 ASRs and use an analytical model to perform an extensive computational experiment in which the performance of these rules is assessed with respect to three performance measures in a wide variety of settings. The adopted set of ASRs is an extension of the 50 ASRs selected in Ho & Lau (1992, 1999). Our set includes many ASRs that are common in practice and/or that have been shown to yield good, robust results (e.g., dome-shaped ASRs and Bailey's rule). We are aware of the fact that other sets of parameter settings are possible. This, however, may further extend the number of ASRs tested. For instance, to create VI rules, we adopted the same logic as Ho & Lau (1992, 1999). The advantage of their approach is its simplicity (only a few parameters are required to define a VI rule). A more involved logic (that uses more parameters) can be devised to capture more VI rules, however, this would result in a large number of ASRs to be tested (i.e., more parameters result in more combinations of parameter values, and hence more ASRs).

The ASRs may be summarized as variations of (1) the individual ASR, (2) the block ASR, and (3) Variable Interval ASRs (also referred to as the VI ASRs).

The individual ASR schedules the arrival times of customers as follows:

$$\begin{array}{rcll} A_i & = & ia\mu^{-1} & \forall i : i < l, \\ A_i & = & A_{i-1} + \mu^{-1} + h\sigma_i & \forall i : l \leq i < N, \end{array} \tag{1}$$

where l denotes the number of customers scheduled for arrival at the start of a service session, a is a multiplier to delay the start of the second until the l-th customer, and h is a multiplier used to adjust the impact of σ_i. We implement 91 variants of the individual ASR by allowing parameters a, l, and h to vary over set $\{0, 0.3, 0.5\}$, set $\mathbf{L} = \{1, 2, 3, 4, 5\}$, and set $\mathbf{H} = \{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$, respectively.

The block ASR may be summarized as follows:

$$\begin{array}{rcll} A_i & = & 0 & \forall i : i < b, \\ A_{jb} & = & A_{(j-1)b} + b\mu^{-1} + h\sqrt{b\sigma_{jb}} & \forall j : 1 \leq j < \frac{N}{b}, \\ A_{jb+i} & = & A_{jb} & \forall i : 1 \leq i < b, \end{array} \tag{2}$$

where b denotes the block size (i.e., the number of customers assigned to arrive at a single time instance), and j is an index iterating over all blocks. Varying parameters b and h over set $(\mathbf{L} \setminus \{1\})$ and set $\mathbf{H}$ respectively, we obtain 28 ASRs.

The VI ASRs schedule customers using intervals with varying length, forcing a dome-pattern. The dome rules of Cayirli et al. (2012) and Cayirli & Yang (2014) are examples of VI ASRs. We implement VI ASRs by speeding up/slowing down the pace of scheduled arrivals using correction factors r_1 and r_2. The computation of scheduled arrival times is performed in two steps. First, all scheduled arrival times are initialized using an individual ASR with $(l = 1)$ and $(h = 0)$. Next, a correction is applied to speed up and/or slow down the pace of scheduled customer arrivals.

$$\begin{array}{rcll} \text{Initialization:} & & & \\ A_0 & = & 0, & \\ A_i & = & A_{i-1} + \mu^{-1} & \forall i : 1 \leq i < N. \\ \text{Correction:} & & & \\ A_i & = & A_i - r_1(z-i)h\sigma_i & \forall i : 1 \leq i \leq z, \\ A_i & = & A_i - r_2(z-i)h\sigma_i & \forall i : z < i < N, \end{array} \tag{3}$$

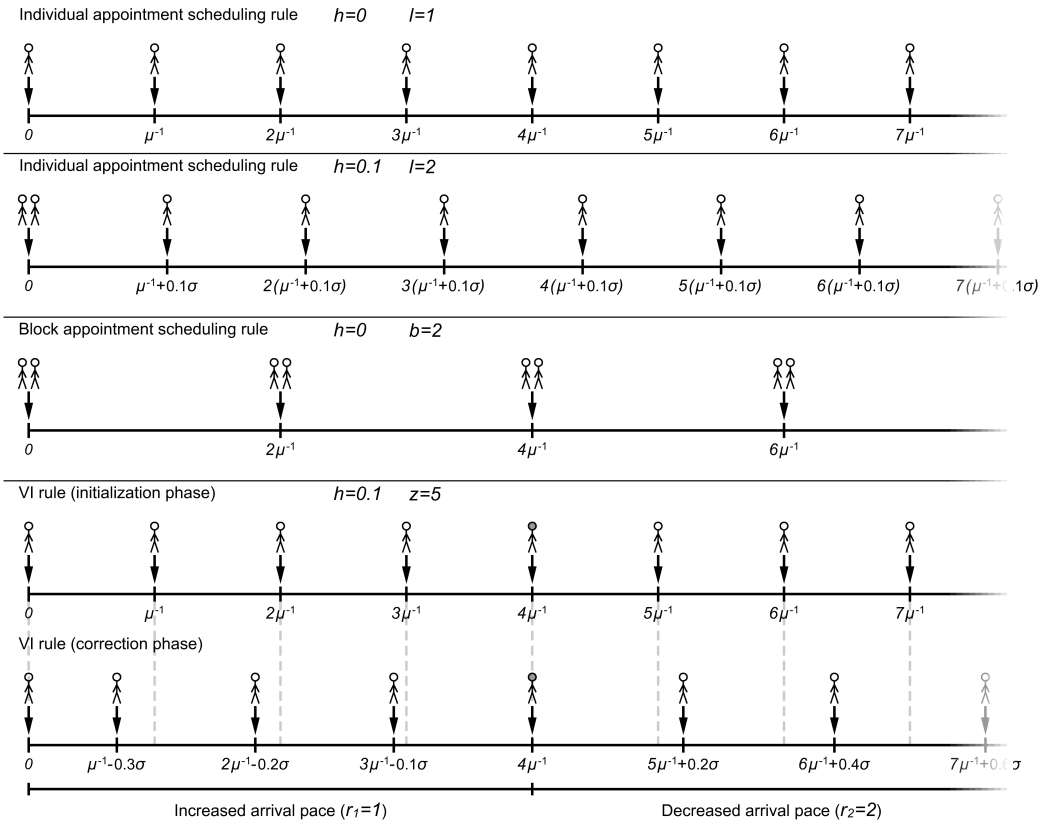

Figure 2: Illustration of different ASRs.

where r_1 and r_2 are correction factors used to speed up or slow down the succession of scheduled arrivals and z is any multiple of 5 smaller than N. Parameter r_1 controls the arrival pace of the first z customers; the arrival pace of these customers increases as r_1 increases. Conversely, parameter r_2 controls the arrival pace of those customers that are scheduled to arrive after customer z (i.e., z divides the set of customers in two parts). When varying parameter h over set $(\mathbf{H} \setminus \{0\})$ and parameters r_1 and r_2 over the set $\{0, 1, 2\}$ (where $(r_1 + r_2) > 0$), we obtain 39 times $\left\lfloor \frac{N-1}{5} \right\rfloor$ ASRs.

A summary of the 314 ASRs may be found in Table 1. Figure 2 illustrates how the different types of ASRs are constructed (i.e., how the appointments times of the different customers are assigned). For instance, Figure 2 illustrates how an individual ASR with $h = 0.1$ and $l = 2$ is used to schedule customers.

4.2 Basic processes

Because of notational requirements introduced in later sections, we will sometimes use the superscript $^{(2)}$ to identify some of the basic processes. For each customer i, define:

- A_i^*, the effective arrival time,
- E_i, the earliest possible arrival time,

Rule number	Conditions [1]
Individual ASR	
[1 - 7], [8 - 14], [15-21], [22-28], [29-35]	$l = 1, 2, 3, 4, 5 \wedge a = 0 \wedge h \in \mathbf{H}$
[36-42], [43-49], [50-56], [57-63]	$l = 2, 3, 4, 5 \wedge a = 0.3 \wedge h \in \mathbf{H}$,
[64-70], [71-77], [78-84], [85-91]	$l = 2, 3, 4, 5 \wedge a = 0.5 \wedge h \in \mathbf{H}$
Block ASR	
[92-98], [99-105], [106-112], [113-119]	$b = 2, 3, 4, 5 \wedge h \in \mathbf{H}$
Variable Interval ASR	
[120-125], [159-164], [198-203], [237-242], [276-281]	$z = 5, 10, 15, 20, 25 \wedge r_1 = 0 \wedge r_2 = 1 \wedge h \in (\mathbf{H} \setminus \{0\})$,
[126-128], [165-167], [204-206], [243-245], [282-284]	$z = 5, 10, 15, 20, 25 \wedge r_1 = 0 \wedge r_2 = 2 \wedge h \in \{0.2, 0.25, 0.3\}$
[129-134], [168-173], [207-212], [246-251], [285-290]	$z = 5, 10, 15, 20, 25 \wedge r_1 = 1 \wedge r_2 = 0 \wedge h \in (\mathbf{H} \setminus \{0\})$
[135-140], [174-179], [213-218], [252-257], [291-296]	$z = 5, 10, 15, 20, 25 \wedge r_1 = 1 \wedge r_2 = 1 \wedge h \in (\mathbf{H} \setminus \{0\})$
[141-146], [180-185], [219-224], [258-263], [297-302]	$z = 5, 10, 15, 20, 25 \wedge r_1 = 1 \wedge r_2 = 2 \wedge h \in (\mathbf{H} \setminus \{0\})$
[147-149], [186-188], [225-227], [264-266], [303-305]	$z = 5, 10, 15, 20, 25 \wedge r_1 = 2 \wedge r_2 = 0 \wedge h \in \{0.2, 0.25, 0.3\}$
[150-155], [189-194], [228-233], [267-272], [306-311]	$z = 5, 10, 15, 20, 25 \wedge r_1 = 2 \wedge r_2 = 1 \wedge h \in (\mathbf{H} \setminus \{0\})$
[156-158], [195-197], [234-236], [273-275], [312-314]	$z = 5, 10, 15, 20, 25 \wedge r_1 = 2 \wedge r_2 = 2 \wedge h \in \{0.2, 0.25, 0.3\}$

with:
l: the number of customers scheduled for the arrival at the start of a service session,
h: a multiplier used to adjust the impact of σ_i,
a: a multiplier to delay the start of the first arriving customers,
b: the block size
z: the first customers for variable interval ASRs
r_1 and r_2: the correction factors to speed up or slow down the pace of scheduled arrivals

[1] The order of the first parameter values corresponds to the order of the ASR rule number intervals; e.g., l=1 for ASRs 1-7, l=2 for ASRs 8-14, etc. For instance, ASR 14 reflects an individual ASR with $l = 2$, $a = 0$, and $h = 0.3$.

Table 1: Summary of the different appointment scheduling rules.

doi:10.1016/j.ejor.2018.10.036 • www.stefancreemers.be • info@stefancreemers.be

- L_i, the latest possible arrival time,
- $P[A_i^* < A_i]$, the probability of arriving early (i.e., prior to the scheduled arrival time A_i),
- $P[A_i^* > A_i]$, the probability of arriving late,
- $P[A_i^* = A_i]$, the probability of arriving on time,
- $P[\delta_i^{(2)} = 1]$, the probability of customer i not showing up (conversely, event $\left(\delta_i^{(2)} = 0\right)$ corresponds to the showing up of customer i),
- $f_i^{(E)}$, the density function of the amount of time customer i arrives early ($F_i^{(E)}$ denotes the cumulative distribution function),
- $f_i^{(L)}$, the density function of the amount of time customer i arrives late ($F_i^{(L)}$ denotes the cumulative distribution function).

The parameters of the service process of a customer may be defined as follows:

- S^{max}, the maximum service time requirement of a customer,
- $f^{(2)}$, the density function of the service time requirement of a customer ($F^{(2)}$ denotes the cumulative distribution function),
- S^*, the realized service time requirement of a customer.

Let $\mathbf{n}$ denote the set of system parameters and environmental variable settings that characterize an AS. For a given set $\mathbf{n}$ and a given schedule of customer arrivals during a service session, we obtain the following performance measures:

- $\mathcal{O}_{\mathbf{n}}$, the expected amount of overtime performed (with O being defined as the available time capacity after which overtime is performed),
- $\mathcal{I}_{\mathbf{n}}$, the expected amount of time the server resides in an idle state,
- $\mathcal{V}_{\mathbf{n}}$, the total expected customer waiting time (i.e., the expected sum of the waiting times of all customers scheduled to receive service during the service session).
- $\mathcal{W}_{\mathbf{n}}$, the expected average customer waiting time.

Note that we assume the server to be idle if: (1) the server has to wait for the first customer, (2) the server has completed serving a customer but has to wait for a new one to arrive, or (3) if service of all customers is completed early (because staff wages, equipment costs, etc. are incurred until the end of the service session).

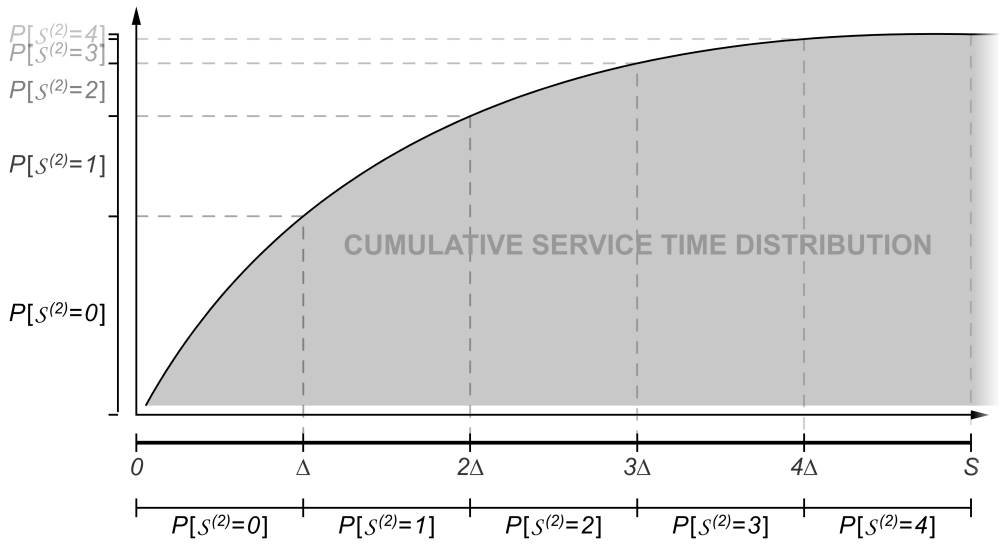

Figure 3: Discretization of the service time requirement distribution.

4.3 Discretization

We model the AS as a DTMC. Let Δ denote the unit time interval over which transitions are observed (e.g., we observe the state of the system every 5 minutes). During a time interval of length Δ, various events may take place (the completion of service of a customer, the arrival of one or more customers, etc.). State transitions (i.e., from a state at time instance $x\Delta$ towards a state at time instance $(x+1)\Delta$, where x is defined as $x := x \in \{0, 1, \ldots, \mathcal{X}\}$ and $\mathcal{X}\Delta$ is the last possible time instance at which service of all customers completes) need to take these unobserved events into account.

With respect to the service process, let $P\left[\mathcal{S}^{(2)} = x\right]$ denote the probability of finishing service during time interval $[x\Delta, (x+1)\Delta)$ (where $\mathcal{S}^{(2)}$ identifies the time interval in which service completes and equals $\left\lfloor \frac{S^*}{\Delta} \right\rfloor$). $P\left[\mathcal{S}^{(2)} = x\right]$ is computed as follows:

$$\begin{aligned} P\left[\mathcal{S}^{(2)} = x\right] &= \int_{x\Delta}^{(x+1)\Delta} f^{(2)}(t)dt \qquad \forall x : x < \left\lfloor \tfrac{S^{max}}{\Delta} \right\rfloor, \\ P\left[\mathcal{S}^{(2)} = \left\lfloor \tfrac{S^{max}}{\Delta} \right\rfloor\right] &= \int_{\left\lfloor \frac{S^{max}}{\Delta} \right\rfloor \Delta}^{S^{max}} f^{(2)}(t)dt. \end{aligned} \tag{4}$$

Note that the maximum number of service phases equals $\left(Y^{(2)} = \left(\left\lfloor \frac{S^{max}}{\Delta} \right\rfloor + 1\right)\right)$. The discretization of the service process is illustrated in Figure 3. The probability of completing service during a time interval $[x\Delta, (x+1)\Delta)$, given that service did not finish prior to time instance $x\Delta$, is defined as:

$$P\left[\mathcal{S}^{(2)} := x | \mathcal{S}^{(2)} > (x-1)\right] = \frac{P\left[\mathcal{S}^{(2)} = x\right]}{\sum_{n=x}^{\left\lfloor \frac{S^{max}}{\Delta} \right\rfloor} P\left[\mathcal{S}^{(2)} = n\right]} \quad \forall x : x \leq \left\lfloor \frac{S^{max}}{\Delta} \right\rfloor. \tag{5}$$

As such, the probability of finishing service during a time interval $[x\Delta,(x+1)\Delta)$ is weighted using the remaining probability mass at a time instance $x\Delta$. The weighted probability of not finishing service during a time interval $[x\Delta,(x+1)\Delta)$ is:

$$P\left[\mathcal{S}^{(2)}>x|\mathcal{S}^{(2)}>(x-1)\right]=1-P\left[\mathcal{S}^{(2)}=x|\mathcal{S}^{(2)}>(x-1)\right]. \tag{6}$$

For notational convenience, let $P_\omega[\mathcal{S}^{(2)}=x]$ and $P_\omega[\mathcal{S}^{(2)}>x]$ be the equivalent of $P\left[\mathcal{S}^{(2)}=x|\mathcal{S}^{(2)}>(x-1)\right]$ and $P\left[\mathcal{S}^{(2)}>x|\mathcal{S}^{(2)}>(x-1)\right]$, respectively. Note that for $(x=0)$, $P_\omega[\mathcal{S}^{(2)}=x]$ equals $P[\mathcal{S}^{(2)}=x]$ and $P_\omega[\mathcal{S}^{(2)}>x]$ equals $\left(1-P[\mathcal{S}^{(2)}=x]\right)$.

With respect to the arrival process, $P\left[\mathcal{A}_i^*=x\right]$ denotes the probability of arrival of customer i during time interval $[x\Delta,(x+1)\Delta)$ (where $\mathcal{A}_i^*$ identifies the time interval in which customer i arrives and equals $\left\lfloor\frac{A_i^*}{\Delta}\right\rfloor$). The equations that determine probability $P\left[\mathcal{A}_i^*=x\right]$ are presented below:

$$P[\mathcal{A}_i^*=x]=\begin{cases}
P[A_i^*<A_i]+P[A_i^*=A_i]+P[A_i^*>A_i]=1 & x=\left\lfloor\frac{E_i}{\Delta}\right\rfloor\wedge x=\left\lfloor\frac{A_i}{\Delta}\right\rfloor\wedge x=\left\lfloor\frac{L_i}{\Delta}\right\rfloor,\\
P[A_i^*<A_i]+P[A_i^*=A_i] & x=\left\lfloor\frac{E_i}{\Delta}\right\rfloor\wedge x=\left\lfloor\frac{A_i}{\Delta}\right\rfloor\wedge x<\left\lfloor\frac{L_i}{\Delta}\right\rfloor,\\
P[A_i^*=A_i]+P[A_i^*>A_i] & x>\left\lfloor\frac{E_i}{\Delta}\right\rfloor\wedge x=\left\lfloor\frac{A_i}{\Delta}\right\rfloor\wedge x=\left\lfloor\frac{L_i}{\Delta}\right\rfloor,\\
P[A_i^*=A_i] & x>\left\lfloor\frac{E_i}{\Delta}\right\rfloor\wedge x=\left\lfloor\frac{A_i}{\Delta}\right\rfloor\wedge x<\left\lfloor\frac{L_i}{\Delta}\right\rfloor,\\
P[A_i^*<A_i]\left(F_i^{(E)}(\infty)-F_i^{(E)}\left(\left(\left\lfloor\frac{A_i}{\Delta}\right\rfloor-\gamma_i^{(E)}\right)\Delta\right)\right) & x=\left\lfloor\frac{E_i}{\Delta}\right\rfloor\wedge x<\left\lfloor\frac{A_i}{\Delta}\right\rfloor,\\
P[A_i^*<A_i]\left(F_i^{(E)}\left(\left(\left\lfloor\frac{A_i}{\Delta}\right\rfloor-x-2\right)\Delta\right)-F_i^{(E)}\left(\left(\left\lfloor\frac{A_i}{\Delta}\right\rfloor-x-1\right)\Delta\right)\right) & x>\left\lfloor\frac{E_i}{\Delta}\right\rfloor\wedge x<\left\lfloor\frac{A_i}{\Delta}\right\rfloor,\\
P[A_i^*>A_i]\left(F_i^{(L)}(\infty)-F_i^{(L)}\left(\left(\left\lfloor\frac{L_i}{\Delta}\right\rfloor-\gamma_i\right)\Delta\right)\right) & x>\left\lfloor\frac{A_i}{\Delta}\right\rfloor\wedge x=\left\lfloor\frac{L_i}{\Delta}\right\rfloor,\\
P[A_i^*>A_i]\left(F_i^{(L)}\left(((x+1)-\gamma_i)\Delta\right)-F_i^{(L)}\left((x-\gamma_i)\Delta\right)\right) & x>\left\lfloor\frac{A_i}{\Delta}\right\rfloor\wedge x<\left\lfloor\frac{L_i}{\Delta}\right\rfloor.
\end{cases} \tag{7}$$

Where: (1) γ_i indicates the end of the time interval in which the arrival of a customer i is scheduled to take place and (2) $\gamma_i^{(E)}$ indicates the end of the first time interval in which the customer is allowed to arrive. γ_i is defined as follows ($\gamma_i^{(E)}$ is defined analogously):

$$\gamma_i:=\left\lfloor\frac{A_i}{\Delta}\right\rfloor+1. \tag{8}$$

The maximum number of arrival phases equals $\left(Y^{(A)}=\left(\left\lfloor\frac{L_i}{\Delta}\right\rfloor-\left\lfloor\frac{E_i}{\Delta}\right\rfloor+1\right)\right)$. The probability of a customer i arriving during a time interval $[x\Delta,(x+1)\Delta)$, given that customer i did not arrive prior to time instance $x\Delta$, is given by:

$$P\left[\mathcal{A}_i^*=x|\mathcal{A}_i^*>(x-1)\right]=\frac{P\left[\mathcal{A}_i^*=x\right]}{\sum_{n=x}^{\left\lfloor\frac{L_i}{\Delta}\right\rfloor}P\left[\mathcal{A}_i^*=n\right]}\;\forall x:\left\lfloor\frac{E_i}{\Delta}\right\rfloor\leq x\leq\left\lfloor\frac{L_i}{\Delta}\right\rfloor. \tag{9}$$

The corresponding weighted probability of a customer not arriving during a time interval $[x\Delta,(x+1)\Delta)$ is:

$$P\left[\mathcal{A}_i^*>x|\mathcal{A}_i^*>(x-1)\right]=1-P\left[\mathcal{A}_i^*=x|\mathcal{A}_i^*>(x-1)\right]. \tag{10}$$

For notational convenience let $P_\omega[\mathcal{A}_i^*=x]$ and $P_\omega[\mathcal{A}_i^*>x]$ be the equivalent of $P\left[\mathcal{A}_i^*=x|\mathcal{A}_i^*>(x-1)\right]$ and $P\left[\mathcal{A}_i^*>x|\mathcal{A}_i^*>(x-1)\right]$, respectively. Note that for $(x=0)$, $P_\omega[\mathcal{A}_i^*=0]$ equals $P[\mathcal{A}_i^*=0]$ and $P_\omega[\mathcal{A}_i^*>0]$ equals $(1-P[\mathcal{A}_i^*=0])$.

5 Model

In this section we discuss the DTMC that is used to model the AS and that allows us to obtain the performance measures. To efficiently compute these performance measures we use an algorithm that is also introduced here.

5.1 Discrete-Time Markov Chain

In order to illustrate the state transitions, define: (1) $\mathbf{N}$, the set of all customers that require scheduling and (2) $\mathbf{T}_x$, the set of customers allowed to arrive during the time interval $[x\Delta, (x+1)\Delta)$. Using the earliest and latest arrival time instances of a customer i, membership of $\mathbf{T}_x$ may easily be determined. The set of customers that have become eligible to arrive at a time instance $x\Delta$ is defined as $(\mathbf{E}_x := (\mathbf{T}_x \setminus \mathbf{T}_{x-1}))$ (with $(\mathbf{E}_0 \equiv \mathbf{T}_0)$). In addition, define the following state-dependent sets:

- $\mathbf{S}$, the set of customers that are eligible to arrive but that have not arrived yet (i.e., $\mathbf{S}$ is the subset of customers in $\mathbf{T}_x$ that did not yet arrive),
- $\mathbf{U}$, the set of customers that arrives (including no-shows),
- $\mathbf{V}$, the set of arriving customers that do not show up.

Note that $\mathbf{V} \subseteq \mathbf{U} \subseteq \mathbf{S} \subseteq \mathbf{T}_x \subseteq \mathbf{N}$ at any time instance $x\Delta$.

The AS may be modeled as a DTMC of four dimensions:

- $x\Delta$, the time instance at which the system is observed,
- $Q : Q \in \{0, 1, 2, \ldots\}$, the number of waiting customers in queue at time instance $x\Delta$,
- $y : y \in \left\{-2, 0, \ldots, Y^{(2)}\right\}$, the phase of the service process at time instance $x\Delta$ (where $(y = -2)$ indicates the completion of service of all customers, $(y = -1)$ indicates server idleness, and $(y \geq 0)$ indicates that a service process is ongoing),
- $\mathbf{S}$, the set of customers that are eligible to arrive at time instance $x\Delta$ but that have not arrived yet.

Because $\mathbf{S} \subseteq \mathbf{T}_x$ at any time instance $x\Delta$, the size of the state space depends heavily on the cardinality of set $\mathbf{T}_x$ (i.e., the size of the state space is mainly determined by the number of customers that is allowed to arrive in parallel during a given time interval). The state space may be divided into two sets of states: (1) transient states which are visited only once and (2) absorbing states which indicate the service completion of all customers at a given time instance (more specifically, each time instance $x\Delta$ is associated with a single absorbing state that masses all probability to complete the service process of all customers at time instance $x\Delta$). We represent the state space using quadruples $(x, Q, y, \mathbf{S})$. In addition, let $\pi[x, Q, y, \mathbf{S}]$ denote the probability to visit state $(x, Q, y, \mathbf{S})$.

A state transition (from a state at time instance $x\Delta$ towards a state at time instance $(x+1)\Delta$) may result in one (or multiple) events occurring. The probability of arrival of a

set of customers $\mathbf{U}$ at a state $(x, Q, y, \mathbf{S})$ is defined as $P[\mathbf{U}|x, \mathbf{S}]$. The equations determining probability $P[\mathbf{U}|x, \mathbf{S}]$ are given below:

$$P[\mathbf{U}|x,\mathbf{S}] = \begin{cases} 1 & \text{for} \quad \mathbf{U} = \emptyset \wedge \mathbf{U}^{\mathsf{c}} = \emptyset, \\ \prod\limits_{i\in\mathbf{S}} P_\omega[\mathcal{A}_i^* = x] & \text{for} \quad \mathbf{U}^{\mathsf{c}} = \emptyset, \\ \prod\limits_{n\in\mathbf{S}} P_\omega[\mathcal{A}_n^* > x] & \text{for} \quad \mathbf{U} = \emptyset, \\ \prod\limits_{i\in\mathbf{U}} P_\omega[\mathcal{A}_i^* = x] \prod\limits_{n\in\mathbf{U}^{\mathsf{c}}} P_\omega[\mathcal{A}_n^* > x] & \text{for} \quad \mathbf{U} \neq \emptyset \wedge \mathbf{U}^{\mathsf{c}} \neq \emptyset, \end{cases} \tag{11}$$

where $\mathbf{U}^{\mathsf{c}}$ is the set of customers that do not arrive. Analogously, the probability of having a set of customers $\mathbf{V}$ not showing up, when a set of customers $\mathbf{U}$ is supposed to arrive, is defined as $P[\mathbf{V}|\mathbf{U}]$. Probabilities $P[\mathbf{V}|\mathbf{U}]$ are computed as follows:

$$P[\mathbf{V}|\mathbf{U}] = \begin{cases} 1 & \text{for} \quad \mathbf{V} = \emptyset \wedge \mathbf{V}^{\mathsf{c}} = \emptyset, \\ \prod\limits_{i\in\mathbf{U}} P\left[\delta_i^{(2)} = 1\right] & \text{for} \quad \mathbf{V}^{\mathsf{c}} = \emptyset, \\ \prod\limits_{n\in\mathbf{U}} P\left[\delta_n^{(2)} = 0\right] & \text{for} \quad \mathbf{V} = \emptyset, \\ \prod\limits_{i\in\mathbf{V}} P\left[\delta_i^{(2)} = 1\right] \prod\limits_{n\in\mathbf{V}^{\mathsf{c}}} P\left[\delta_n^{(2)} = 0\right] & \text{for} \quad \mathbf{V} \neq \emptyset \wedge \mathbf{V}^{\mathsf{c}} \neq \emptyset, \end{cases} \tag{12}$$

where $\mathbf{V}^{\mathsf{c}}$ is the set of customers that do show up.

Seven transitions are possible at a time instance $x\Delta$:

- service is ongoing and does not finish during $[x\Delta, (x+1)\Delta)$,
- service is ongoing, finishes and at least one customer is present in the queue at time instance $(x+1)\Delta$,
- service is ongoing, finishes and although no customers are left in the queue at time instance $(x+1)\Delta$, there are still customers that have to arrive,
- service is ongoing, finishes and all customers have arrived or have failed to show up (i.e., an absorbing state has been entered; service has finished at time instance $x\Delta$).
- the server is idle and at least one customer arrives during $[x\Delta, (x+1)\Delta)$,
- the server is idle, no customer arrives during $[x\Delta, (x+1)\Delta)$ and some customers have yet to arrive,
- the server is idle, no more customers are present in the queue and all customers have arrived (i.e., an absorbing state has been entered; service has finished at time instance $x\Delta$).

5.2 Performance measures

The transition probabilities may be used to calculate $\pi[x, Q, y, \mathbf{S}]$, the probability of visiting a state $(x, Q, y, \mathbf{S})$. Using the probabilities to visit each of these states, the performance measures may easily be obtained. More specifically, a state $(x, Q, y, \mathbf{S})$ (with corresponding probability $\pi[x, Q, y, \mathbf{S}]$) is associated with:

- a total customer waiting time of $Q\Delta$ time units (i.e., Q customers are waiting during time interval $[x\Delta, (x+1)\Delta)$),
- a server idle time of Δ time units if $(y = -1)$,
- a server idle time of $(O - x\Delta)$ time units if: (1) $(x\Delta < O)$ or (2) $(y = -2)$ (i.e., $(x, Q, y, \mathbf{S})$ is an absorbing state),
- a server overtime of $(x\Delta - O)$ time units if: (1) $(x\Delta > O)$ or (2) $(y = -2)$.

General performance measures may be obtained as the weighted sum of the performance measures corresponding to each of the states (where the probabilities of visiting a state serve as weights). More formally, for a given set $\mathbf{n}$ and a given schedule of customer arrivals, the expected amount of overtime performed is given by:

$$\mathcal{O}_{\mathbf{n}} = \sum_{x > \lfloor \frac{O}{\Delta} \rfloor}^{\mathcal{X}} \pi[x, 0, -2, \emptyset] \left(x\Delta - O\right). \tag{13}$$

With respect to the expected server idle time, we obtain the following result:

$$\mathcal{I}_i = \left(\sum_{x=0}^{\mathcal{X}} \sum_{\mathbf{S} \subseteq \mathbf{T}_x} \pi[x, 0, -1, \mathbf{S}]\Delta \right) + \left(\sum_{x=0}^{\lfloor \frac{O}{\Delta} \rfloor - 1} \pi[x, 0, -2, \emptyset] \left(O - x\Delta\right) \right). \tag{14}$$

The total expected average customer waiting time may be expressed as:

$$\mathcal{V}_{\mathbf{n}} = \sum_{x=0}^{\mathcal{X}} \sum_{Q=1}^{N} \sum_{y=0}^{Y^{(2)}} \sum_{\mathbf{S} \subseteq \mathbf{T}_x} \pi\left[x, Q, y, \mathbf{S}\right] Q\Delta. \tag{15}$$

Conversely, the expected average customer waiting time is given by:

$$\mathcal{W}_{\mathbf{n}} = \frac{\mathcal{V}_{\mathbf{n}}}{\sum_{i=0}^{N} P[\delta_i^{(2)} = 0]}. \tag{16}$$

Where $\left(\sum_{i=0}^{N} P[\delta_i^{(2)} = 0] \right)$ denotes the expected number of customers to show up.

5.3 Algorithm and implementation

The algorithm consists of two main steps: (1) initialization and selection of the ASR and (2) iterative computation of probabilities $\pi[x, Q, y, \mathbf{S}]$ and the assessment of performance measures. During the initialization, the ASR is selected. The selected rule determines the arrival process. The service process does not depend on the ASR. The iterative procedure uses probabilities $\pi[x, Q, y, \mathbf{S}]$ (associated with a time instance $x\Delta$) to compute probabilities $\pi[(x+1), Q, y, \mathbf{S}]$ (associated with a time instance $(x+1)\Delta$). Performance measures are

computed simultaneously. After computation of all probabilities $\pi[(x+1), Q, y, \mathbf{S}]$, probabilities $\pi[x, Q, y, \mathbf{S}]$ are no longer needed. As such, the memory occupied by these latter probabilities may be released. The iterations continue until all probability mass is gathered in the absorbing states. Next, performance measures corresponding to the selected ASR are stored. The process is repeated until all adopted ASRs have been assessed. A general outline of the algorithm is presented in algorithm 1.

Algorithm 1 Algorithm for computing performance measures.

```
for all x do
  Compute P_ω[S^(2) = x] and P_ω[S^(2) > x]
end for
for all Appointment scheduling rules do
  Compute P_ω[A_i^* = x] and P_ω[A_i^* > x]
  Set x = 0
  for all Q, y, S do
    Compute π[x, Q, y, S]
    Update performance measures
  end for
  while x < X do
    for all Q, y, S do
      Compute π[(x+1), Q, y, S] using π[x, Q, y, S]
      Update performance measures
    end for
    Free memory used by states (x, Q, y, S)
    Increment x
  end while
  Store performance measures
end for
```

The algorithm is implemented in Visual C++. The main inputs of the application are: (1) the size of the unit time interval, (2) the number of customers that require scheduling, (3) the parameters of the service process, and (4) the parameters of the arrival process of each of the customers.

5.4 Model extensions

In this section, we discuss three model extensions: (1) the delayed start of a service session, (2) interruptions that take place during the service process of a customer (i.e., preemptive interrupts), and (3) interruptions that take place in between the service process of two subsequent customers (i.e., non-preemptive interrupts, the delayed start of service itself). In order to take these extensions into account, we allow for an additional Markov chain dimension that captures the type of service process currently in progress. As such the resulting DTMC holds five dimensions. Its state space may be represented by quintuples $(x, Q, y, w, \mathbf{S})$, where w indicates the type of service currently in progress. By convention we have:

- $(w = -1)$ if the server is idle,
- $(w = 0)$ if the service process cannot start because the start of the service session is delayed,
- $(w = 1)$ if the ongoing service process is subject to non-preemptive interrupts,
- $(w = 2)$ if a regular service process is ongoing (i.e., as defined in the previous sections),
- $(w = 3)$ if the ongoing service process is subject to preemptive interrupts.

With the exception of $(w = -1)$, these service outages are modeled as "special" customers, each associated with a unique service process characterization. More specifically, each type of service has unique parameters: $f^{(w)}$, $P[\mathcal{S}^{(w)} = x]$, $P[\mathcal{S}^{(w)} = x|\mathcal{S}^{(w)} > (x-1)]$, $P[\mathcal{S}^{(w)} > x|\mathcal{S}^{(w)} > (x-1)]$, $Y^{(w)}$ and $P[\delta^{(w)} = 1]$ (note that index i is discarded for the non-regular types of service processes). The type of the ongoing service process is decided at: (1) the start of a service session (for the delayed start of a services session), (2) the start of a service process (for the delayed and the regular start of a service process), and (3) the end of a service process (for a service process subject to preemptive interrupts). A detailed discussion of how to implement these extensions is given in Creemers (2009a).

6 Computational experiment

The model has been verified by means of an elaborate simulation study in which each of the operating environments is simulated using 5,000,000 simulation iterations (Creemers, 2009a). Various values of Δ were tested, and a value of $(\Delta = 5)$ was shown to provide a sufficient level of accuracy while maintaining computational performance. As such, in the upcoming experiment we let $(\Delta = 5)$. Note that the 5-minute intervals have also been used by other researchers, such as Klassen & Yoogalingam (2009). It should be noted, however, that restricting our experiments to the more practical setting of 5-minute intervals does not allow to simulate a true ASR VI environment. This restriction might explain the results presented in Section 6.2. In what follows, we discuss the design and the results of the computational experiment.

6.1 Experimental design

We consider 243 operating environments that are generated by all combinations of the experimental design parameters given in Table 2. The parameter values of the computational experiment are based on the studies listed in Cayirli et al. (2008). Note that, if the probability for customers to arrive early or late is zero, we do not have to consider the variability of the early or late arrival times.

Let $\mathbf{P} = \{\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_{243}\}$ denote the set of operating environments. In addition, define $(O := N\mu^{-1})$ as the time capacity after which overtime is performed. We evaluate 158, 236, and 314 ASRs for each value of N, respectively (resulting in a total of 57,348 instances

Table 2: Experimental design: environmental parameters.

Environmental parameter	Settings
Number of customers N	$\{10, 20, 30\}$
Squared coefficient of variation of service times (service SCV) (the mean service requirement equals 300 time units)	$\{0.2, 0.5, 1.0\}$
Squared coefficient of variation of early and late arrival times (the mean early/late arrival time amounts to 150 time units)	$\{0.5, 1.0\}$
Probability of early arrival	$\{0, 0.1\}$
Probability of late arrival	$\{0, 0.1\}$
Probability of no-show	$\{0, 0.1, 0.2\}$

analyzed). The performance measures of an ASR $r : r \in \{1, 2, \ldots, 314\}$ are:

$$\mathcal{O}_r = \sum_{i=1}^{243} \mathcal{O}_{\mathbf{n}_i,r}, \tag{17}$$

$$\mathcal{I}_r = \sum_{i=1}^{243} \mathcal{I}_{\mathbf{n}_i,r}, \tag{18}$$

$$\mathcal{W}_r = \sum_{i=1}^{243} \mathcal{W}_{\mathbf{n}_i,r}, \tag{19}$$

where $\mathcal{O}_{\mathbf{n}_i,r}$, $\mathcal{I}_{\mathbf{n}_i,r}$, and $\mathcal{W}_{\mathbf{n}_i,r}$ denote the expected server overtime, the expected server idle time, and the expected customer waiting time when ASR r is used to schedule the arrival of customers at an AS that operates in environment $\mathbf{n}_i$.

While the implementation of an ASR might yield good results in terms of a single performance measure (e.g., server idle time), its impact on another performance measure (e.g., customer waiting time) can be detrimental. Hence, the need to consider multiple performance measures when evaluating ASRs. To conduct a multidimensional performance evaluation, we use a composite indicator (CI):

$$CI_r = v_o\mathcal{O}_r + v_i\mathcal{I}_r + v_w\mathcal{W}_r, \tag{20}$$

where CI_r is the weighted sum of the performance of an ASR r and $v_{(\cdot)}$ is the weight allocated to performance measure $(\cdot)$.

An ASR r performs well if the score over all performance measures is low (i.e., the lower the value of the CI, the better the ASR performs). Although practical and intuitive, CIs have several drawbacks, among which the need to normalize the performance measures and the inherent difficulty of determining appropriate weights (Cherchye, Moesen, Rogge, Van Puyenbroeck, Saisana, Saltelli, Liska, & Tarantola, 2008). Cherchye et al. (2008) have demonstrated the applicability of DEA to objectively set weights. In order to avoid the subjective fixing of weights, we use DEA to identify the optimal set of weights for each ASR individually.

The resulting CIs are conservative (i.e., allow high weights to be set on strong performance measures and low weights on measures for which performance is bad). This can be

a welcome feature as best-practice ASRs can be identified. Nevertheless, zero weights are often allocated, which is problematic as every performance measure included is by definition relevant. Extensive research has been conducted in order to identify methods that allow to avoid zero weights while maintaining the statistical properties of the DEA method (e.g., Cooper, Ruiz, & Sirvent, 2007; Portela & Thanassoulis, 2006).We opt to avoid zero weights by adding relative weight restrictions, thereby forcing each weight to have a relative weight of at least 5% of the total weights (see also constraints (26), (27), and (28) below). We implement a super-efficiency DEA approach, which eliminates the Decision Making Unit (DMU) under evaluation, referred to as r', from the reference set S (Chen & Du, 2015). We solve the following model for each ASR:

$$\min v_o\mathcal{O}_{r'} + v_i\mathcal{I}_{r'} + v_w\mathcal{W}_{r'} \tag{21}$$

subject to

$$\begin{aligned}
(v_o\mathcal{O}_r) + (v_i\mathcal{I}_r) + (v_w\mathcal{W}_r) &\geq 1 \qquad \forall r \in S\backslash\{r'\} && (22)\\
v_o &\leq 1 && (23)\\
v_i &\leq 1 && (24)\\
v_w &\leq 1 && (25)\\
v_o &\geq 0.05(v_o + v_i + v_w) && (26)\\
v_i &\geq 0.05(v_o + v_i + v_w) && (27)\\
v_w &\geq 0.05(v_o + v_i + v_w) && (28)
\end{aligned}$$

Model (21) minimizes the value of the CI by selecting weights, that need to be smaller than 1 and larger than 5% of the sum of all weights. Consequently, the selected weights do not contain any zero values. Model (21) is solved for each of the ASRs, resulting in 314 CI values and 314 sets of weights. The model yields high CI values for ASRs that are less attractive. In order to make the CI more intuitive, we use the inverse value. As such, higher CI values indicate a better performing ASR.

Although the objectivity of a DEA-based performance evaluation is a merit, decision makers can have good reasons to value some performance measures more than others. Different possibilities exist to incorporate such a valuation into the DEA (Cook & Seiford, 2009). We incorporate the valuation of different performance measures by adding constraints:

$$\frac{v_{(\cdot)}}{v_{(\cdot)}} \leq 1, \tag{29}$$

where $(\cdot)$ is a performance measure. Constraint $\left(\frac{v_o}{v_i} \leq 1\right)$ for example, imposes the restriction that the expected server idle time is considered to be more important than the expected server overtime. We obtain results for the following four scenarios: (1) server overtime is more important ($v_o \geq v_i$ and $v_o \geq v_w$), (2) server idle time is more important ($v_i \geq v_o$ and $v_i \geq v_w$), (3) customer waiting time is more important ($v_w \geq v_o$ and $v_w \geq v_i$), and (4) there is no preference between the three outputs.

Some ASRs will perform strongly across a wide range of possible weight sets, while others may have a CI value that depends heavily on the choice of a particular set of weights.

Table 3: Ranking of ASRs based on average efficiency across environments.

	N=10		N=20		N=30		Average			
Rank	**ASR**	**Type**	**ASR**	**Type**	**ASR**	**Type**	**ASR**	**Type**	**CI (%)**	**Maverick**
1	7	IND	7	IND	144	VI	7	IND	100.0%	1.39
2	8	IND	8	IND	8	IND	8	IND	97.9%	1.32
3	70	IND	70	IND	70	IND	70	IND	97.6%	1.41
4	6	IND	128	VI	228	VI	138	VI	97.3%	1.26
5	36	IND	175	VI	145	VI	228	VI	97.3%	1.19
6	37	IND	9	IND	128	VI	247	VI	97.3%	1.27
7	151	VI	208	VI	247	VI	69	IND	97.2%	1.32
8	138	VI	151	VI	152	VI	175	VI	97.2%	1.15
9	69	IND	138	VI	176	VI	151	VI	97.2%	1.15
10	133	VI	69	IND	138	VI	208	VI	97.2%	1.15
11	38	IND	152	VI	139	VI	9	IND	97.1%	1.18
12	9	IND	189	VI	214	VI	36	IND	97.1%	1.18
13	132	VI	145	VI	190	VI	176	VI	97.1%	1.19
14	137	VI	139	VI	69	IND	214	VI	97.1%	1.18
15	143	VI	228	VI	39	IND	139	VI	97.1%	1.27

To measure the sensitivity of the CI value to the selected set of weights, we calculate the maverick index (Doyle & Green, 1994; Markovits-Somogyi, 2011). If the maverick index is high, the CI score of the ASRs is sensitive to the choice of weights. A low maverick index indicates a robust performance across different sets of weights.

6.2 Experimental results

Next to the development of a new analytical model to study the impact of ASRs, we also show some interesting findings from our experiments using DEA. First, we discuss the performance of all ASRs over all environments. Next, we assess the impact of environmental variables and discuss the influence of subjective valuation of the different performance measures. Although our method is suitable to select the best ASR for any given setting and preference structure, we focus on general insights with respect to the three types of ASRs (i.e., individual ASRs, block ASRs, and VI ASRs).

Based on the results of the DEA, Table 3 provides an overview of the best-performing ASRs across all environments, and over all performance measures. The table reports: (1) the 15 best-ranked ASRs for a different number of customers ($N = \{10, 20, 30\}$) and (2) the best-ranked ASR on average along with their CI value and maverick index. Note that we represent the CI of the ASR under evaluation as the CI of the best performing ASR on average over the CI of the ASR under evaluation, and that we include all possible ASRs (from 1 to 314) in the average ranking (while for $N = 10$, $ASR > 158$, and for $N = 20$, $ASR > 236$ are not feasible).

It is striking that the individual ASRs are among the best performing ASRs, even more so when the number of customers is small. The simple Bailey-Welch rule (ASR 8) performs very good in terms of the efficiency score. Rule 7 allows for a maximum adjustment for the service time standard deviation. When $N = 30$ (see Table 3), rule 7 is no longer among the best-performing ASRs. Since this rule benefits from the fact that two initial customers are present at the start of the service session, this means that, the more customers, the longer the service session takes, the less impact these initial customers have.

We also note high efficiency scores for the VI ASRs (dome-shaped rules). Indeed, these conclusions are largely confirmed by the literature. It is, however, also highly recommended

to look at the Maverick scores (as a measure of robustness). Here we observe that the individual ASRs perform less in terms of robustness compared to the VI-based rules. The individual ASRs are in other words more sensitive to the weight selection. Since our paper values robustness, we opt for VI ASRs in complex, dynamic environments. In the remaining of this section, we will further refine our results.

Block ASRs are the least attractive type of ASR. The best-performing block ASRs (ASR 92, 93, and 94) have a block size of two with a small (or even zero) adjustment for service time variance (based on the CI over all environments, of the 314 possible ASRs, they are ranked at positions 210, 218, and 231, respectively). Their maverick index is rather low. The dominance within the block ASRs of rules with a block size of two (i.e., customers arrive in groups of two) confirms the conclusions of earlier research (Blanco White & Pike, 1964; Ho & Lau, 1992).

The efficiency results reported in Table 3 are based on the average performances for the three objectives over all settings listed in Table 2. The setting in which the probability of early arrivals is higher than the probability of late arrivals, i.e., probability of early arrival = 0.1 and probability of late arrival = 0 in Table 2, is however much more realistic than the other three combinations of both probabilities. When only this most realistic setting is considered, the rankings listed in Table 3 do not change much; i.e., ASR 7 and 8 are still the best performing ASRs, the individual ASRs and VI ASRs are outperforming the block ASRs, and ASR 92 and 93 are still the best performing block ASRs.

Based on both prior research and the results discussed above, we select eight ASRs for further discussion. We select (1) good performing ASRs for different number of customers according to Table 3, (2) ASRs that are sufficiently different from one another (i.e., IND, VI, as well as block ASRs), and (3) ASRs that can be implemented for all values of N. Table 4 presents the selected ASRs and their characteristics.

We select four individual ASRs that differ in terms of number of customers at session start, first arrival delay, and maximum adjustment for service time standard deviation. The customers at session start equals 1 or 2. The two selected block ASRs both have a block size of two and a very small (or zero) adjustment of the arrival times. With respect to the VI ASRs, we select two rules that postpone the arrival rate of customers after the arrival of the first 5 customers (the arrival rate of the first five customers is not corrected). They differ in terms of the adjustment for service time standard deviation. In what follows, starting from the 8 selected ASRs, we first illustrate the effect of the different environmental parameters. Next, we analyze the impact of subjective valuation of the different performance measures.

6.2.1 Impact of environmental variables

In Figure 4, we examine the impact of (1) the probability of no-shows, (2) the variability of the service times (service SCV), and (3) the number of customers during a service session, on the performance (in terms of customer waiting time, server idle time, and server overtime) of the ASRs in Table 4.

From Figure 4 it is clear that the customer waiting time increases with: (1) the number of customers N, (2) the level of service time variability, and (3) a decreasing probability of no-shows. The impact of service SCV and no-shows on server overtime is similar to their impact on waiting time. A larger number of customers, service time variability, and probability of

Table 4: Overview of the selected ASRs.

ASR	Characteristics
7	Individual ASR, one customer at session start, no first arrival delay, maximum adjustment for service time standard deviation (h=0.3)
8	Individual ASR, two customers at session start, no first arrival delay, no adjustment for service time standard deviation (h=0)
36	Individual ASR, two customers at session start, medium first arrival delay (a=0.3), no adjustment for service time standard deviation (h=0)
70	Individual ASR, two customers at session start, maximum first arrival delay (a=0.5), maximum adjustment for service time standard deviation (h=0.3)
92	Block ASR, block size of two customers, no adjustment for service time standard deviation (h=0)
93	Block ASR, block size of two customers, no adjustment for service time standard deviation (h=0.05)
138	VI ASR, postpone arrivals after first 5 customers (z=5), maximum adjustment for service time standard deviation (h=0.2)
151	VI ASR, postpone arrivals after first 5 customers (z=5), small adjustment for service time standard deviation (h=0.1)

no-shows all lead to an increase in server idle time.

ASR 7 (IND ASR with $l = 1$ and $a = 0$) and 151 (VI ASR with $z = 5$) have a strong performance in customer waiting time, however they have the highest server idle and overtime. The block ASRs 92 and 93 are characterized by lower performance (i.e., higher waiting, idle, and overtime).

We find that the individual ASR 70 (with $l = 2$ and $a = 0.5$), which allows for a high first arrival delay, performs worse on server idle and overtime. Compared to the block ASRs 92 and 93, the VI ASRs also perform worse on server idle and overtime.

Table 5 shows the impact of a change in number of customers, service SCV, and no-show probability, on customer waiting time, server idle time, and server overtime for the different ASRs. A positive value means the time increased (i.e., got worse) due to the change. The table confirms that an increase of the service time variability always leads to a worse performance in all three performance criteria, which is an intuitive result. An increase in the number of customers also results in longer waiting times, more overtime, and more sever idle time. The latter is counter-intuitive at first sight, but can be explained by the fact that in our experiments the time capacity increases with a growing number of customers (see Section 6.1). More customers, combined with a corresponding larger time capacity, leads to an increased probability of server idle time. This can be explained by the fact that with a growing number of customers who are processed in a longer time horizon, there is a higher probability that things go seriously wrong during specific moments (higher customer waiting times) while other moments are calm (higher service idle time). Finally, an increase of no-shows is negative for idle time, but not for waiting time, nor for overtime; which is again an intuitive result.

Table 6 shows which of the 314 ASRs performs best under environments with high no-shows, high variability in service time, or a high number of customers. While the individual ASRs perform best under high no-show probability and high service time variability, the

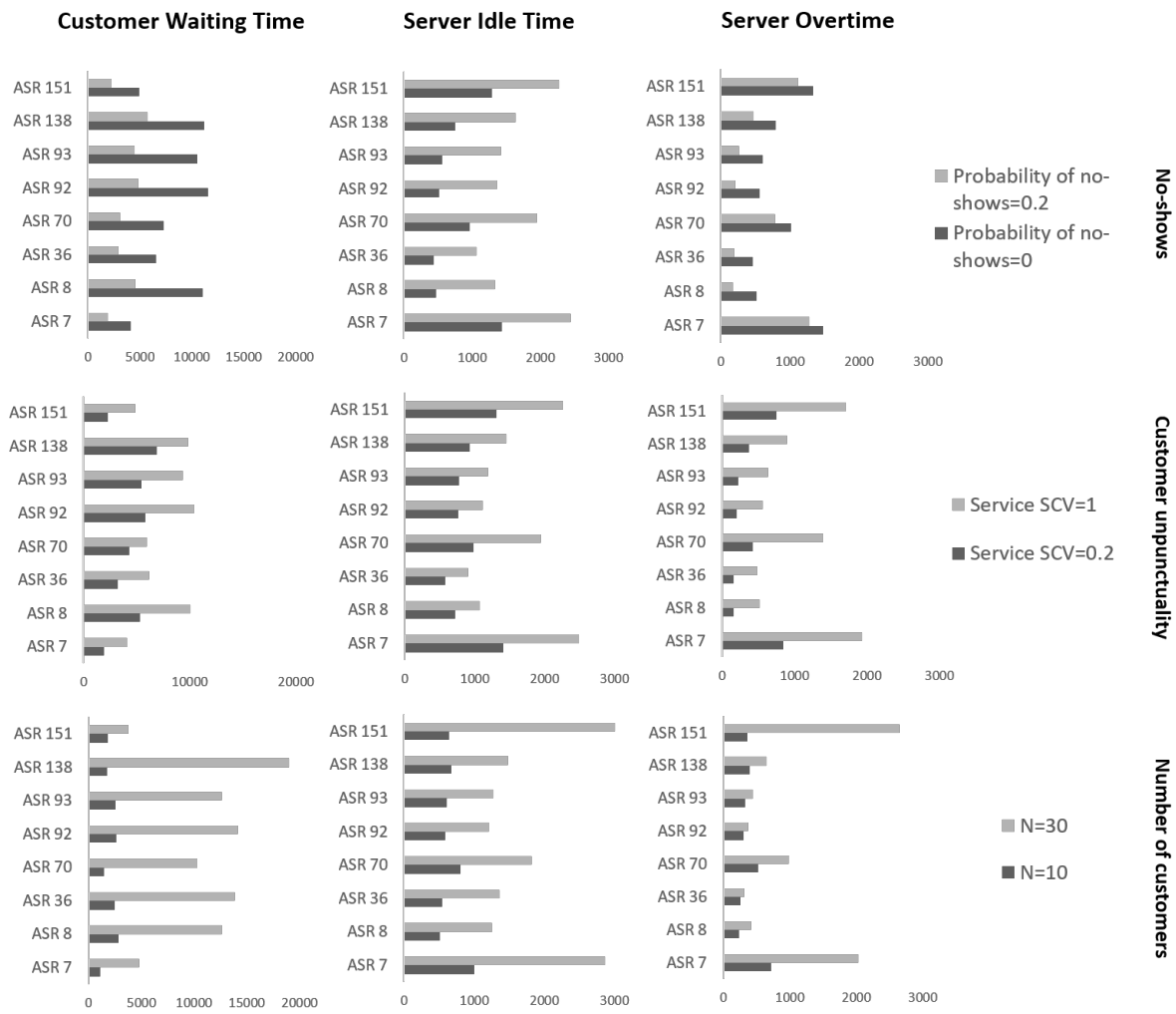

Figure 4: Impact of environmental variables on the selected ASR CI score.

variable ASRs are often the best performing ASRs under a high number of customers. Once more, the simple Bailey-Welch rule (ASR 8) performs very well, however, it is not in the top 10 best-ranked ASRs when service time variability is high. The strong performance of the Bailey-Welch rule was also noted in Ho & Lau (1992).

From our results we also found that customer unpunctuality only has a minor impact on the performance of an ASR. For instance, for ASR 8, while the impact on waiting time, idle time, and overtime of a change in number of customers, no shows, and service variability lies between -58.9% and 349.5% (see Table 5), a change in the probability of being late/early from 0 to 0.1 has only a minor impact ranging between -3.88% and 0.81%. While customers being too late is negative for both waiting time, idle time, and overtime, customers arriving too early is only negative for waiting time.

Based on the results discussed above, we observe that the impact of environmental variables also differs depending on the selected ASR.

Table 5: Change in performance due to a change in environmental factors for the selected ASR.

ASR	**Change in performance when N increases from 10 to 30**			**Change in performance when probability of no-shows increases from 0 to 0.20**			**Change in performance when service time variability increases from 0.2 to 1**		
	Waiting time	**Idle time**	**Over time**	**Waiting time**	**Idle time**	**Over time**	**Waiting time**	**Idle time**	**Over time**
7	321.2%	187.5%	183.9%	-53.6%	70.8%	-13.3%	113.4%	77.0%	128.4%
8	349.5%	144.3%	81.1%	-58.9%	183.9%	-66.4%	88.4%	49.5%	226.1%
36	454.6%	150.6%	23.5%	-54.8%	146.9%	-58.9%	92.8%	56.3%	206.2%
70	608.2%	126.4%	88.5%	-57.9%	102.2%	-22.3%	38.7%	98.5%	230.2%
92	436.4%	106.3%	22.6%	-58.0%	166.4%	-62.8%	78.6%	46.8%	180.7%
93	398.9%	110.9%	36.3%	-57.7%	154.3%	-57.4%	73.1%	53.0%	190.6%
138	993.2%	119.6%	64.7%	-49.1%	118.3%	-40.4%	41.8%	56.0%	141.4%
151	105.3%	441.1%	630.3%	-53.4%	77.0%	-16.4%	117.4%	72.9%	128.0%

Table 6: Ranking of ASRs based on their ability to cope with a high amount of no-shows, service time variability, or number of customers.

Rank	High probability of no shows (0.2)		High service time variability (service SCV=1)		High number of customer (N=30)	
1	7	IND	7	IND	144	VI
2	**8**	IND	10	IND	**8**	IND
3	36	IND	9	IND	70	IND
4	43	IND	11	IND	228	VI
5	125	VI	16	IND	145	VI
6	16	IND	12	IND	128	VI
7	71	IND	13	IND	247	VI
8	64	IND	70	IND	152	VI
9	15	IND	14	IND	176	VI
10	44	IND	17	IND	138	VI

6.2.2 Impact of subjective valuation of performance measures

We now discuss how the suitability of the ASRs changes when either the customer waiting time, the server idle time, or the server overtime is pivotal for scheduling purposes. Table 7 lists the 15 best-performing ASRs across different performance measure preferences (e.g., when waiting time is more important the weights assigned by the DEA are higher than the weights of idle time and overtime) While ASRs 7 and 70 (both IND ASR with h=0.3) perform well under different performance preferences, ASR 8 (the Bailey-Welch rule) is only ranked 200th of the 314 ASRs when waiting time is more important. When comparing the overall ranking in Table 3, and the ranking in Table 7 when idle time is the most important performance measure, they are exactly the same. Moreover, the ranking where idle time is most important, looks very similar to what we observe for a high number of customers ($N = 30$); in Table 3, we find that VI ASRs 144 and 145 (both with $z = 5$) perform well when waiting time is important, whereas they perform worse when idle or overtime is important. As expected, block ASRs perform badly. However, we find that they perform better when waiting time is important (the first block ASR in the ranking (i.e., ASR 98) is ranked 104th) compared to the situation where idle time or overtime is more important (in which the first block ASR (i.e., ASR 92) is ranked 210th and 212th, respectively). In block schedules, groups of customers are scheduled at fixed moments in time in order to exploit a pooling effect that leads to a decreased total service time variability that reduces the expected waiting time and idle time. It must be clear that overtime cannot take advantage of this, because the total processing time will not be impacted. On the contrary, with block ASRs the server has a larger probability to be idle just before the block start times, thereby extending the total session duration and thus having a negative impact on overtime. Furthermore, similar to block ASRs, individual ASRs allow to schedule multiple customers at the start of a session, but they are not restricted in the remainder of the session to a number of fixed moments to schedule groups of customers. Consequently, an individual ASR that schedules a sufficiently large number of customers at the start of the session will often protect better against idle time, when compared to a block ASR.

7 Conclusion

Appointment scheduling rules are used to determine the point in time at which a customer is to receive service during a service session. ASRs are commonly applied in service and manufacturing industries (e.g., healthcare or after-sales service).

We develop an analytical model that uses a DTMC and an efficient algorithm to assess the performance (in terms of customer waiting time, server idle time, and server overtime) of ASRs in a wide variety of settings. More specifically, the model takes into account the following environmental factors: (1) customer unpunctuality, (2) no-shows, (3) service interruptions, and (4) delay in the session start time. In addition, the model allows the characterization of the arrival and service process for each individual customer. The model has been verified using a simulation study. We use the model to assess the performance of 314 ASRs and use DEA to compare the results.

This paper focuses on the development of good appointment rules, and does not fo-

Table 7: Ranking of ASRs across environments when performance measures are not considered to be equally important.

Rank	**Waiting Time is most important**			**Idle Time is most important**			**Overtime is most important**		
	ASR	**CI**	**Type**	**ASR**	**CI**	**Type**	**ASR**	**CI**	**Type**
1	7	1	IND	7	1	IND	**8**	1	IND
2	70	0.9744	IND	**8**	0.9994	IND	7	0.9993	IND
3	144	0.9672	VI	70	0.9981	IND	70	0.9970	IND
4	6	0.9638	IND	138	0.9951	VI	138	0.9938	VI
5	143	0.9629	VI	228	0.9949	VI	228	0.9937	VI
6	145	0.9626	VI	247	0.9946	VI	247	0.9936	VI
7	42	0.9617	IND	69	0.9944	IND	69	0.9930	IND
8	69	0.9605	IND	175	0.9940	VI	175	0.9927	VI
9	128	0.9596	VI	151	0.9938	VI	151	0.9924	VI
10	140	0.9579	VI	208	0.9934	VI	208	0.9920	VI
...									
104	98	0.8049	**Block**	287	0.9751	VI	3	0.9711	IND
...									
200	**8**	0.7008	IND	55	0.9349	IND	251	0.9233	VI
...									
207	92	0.6865	**Block**	91	0.9270	IND	243	0.9164	VI
...									
210	54	0.6833	IND	92	0.9250	**Block**	56	0.9146	IND
...									
212	256	0.6779	VI	222	0.9242	VI	92	0.9107	**Block**

cus on the sequencing issue. We opted for this approach because we believe that in most real-life cases the dynamic nature of the appointment process dominates. In such complex environments it is of vital importance to focus on robustness.

In general, we find that the well-known Bailey-Welch rule (an individual ASR with 2 customers scheduled at the start) performs well over many environments. This good performance is weaker in case of high service time variability or if waiting time becomes more important. It is interesting to see that simple rules like these can perform very well. The other individual ASRs also perform quite well, even more so for a small number of customers. Block ASRs perform poorly in all cases. When considering the robustness of ASRs over various environments, VI (or dome-shaped) ASRs are among the best performing ASRs. This indicates that VI ASRs are to be recommended in AS where environmental variables are prone to change.

When idle time or overtime become more important, the average ranking of ASRs hardly changes when compared to the ranking where there is no preference of measures. When waiting time becomes more important, however, the ranking changes drastically. This is a crucial insight. The optimal ASR in an environment where waiting time is highly important may perform poorly in an environment where server idle time and/or overtime are important (and vice versa). In addition, we show that the three performance measures (customer waiting time, server idle time, and overtime) are always negatively impacted by an increase in the number of customers (with corresponding increase in time capacity) and in the level of service time variability. An increase in no-show probability, on the other hand, increases server idle time, but decreases customer waiting times and server overtime. Note that generally, customer unpunctuality does not seem to have a large impact on the different performance measures. An important managerial insight that follows, is that good performing ASRs do not necessarily need to be updated when customer punctuality changes.

There are several interesting avenues for future research. First of all, we would like to examine to what extent our approach can efficiently cope with a multiple-server setting. Second, allowing for cancellations would enable us to include a fourth objective, (i.e., the minimizations of the number of cancellations), and to study different heuristics to decide on a cancellation in order to obtain an optimal trade-off with the other objectives. These heuristics might take advantage of particular appointment schedules that allow for a faster cancellation decision, reducing the expected waiting time and overtime. The continuously growing collection of data in health care (both medical data and operational data about internal processes) combined with a continuously increasing computing power will enable the use of complex ASRs which go beyond the human capability of understanding. The question arises whether the resulting "optimized" appointment schedules based on such "black box" algorithms will be generally accepted. Simple ASRs have the important advantage that implementation is easy and schedules based on understandable rules are more easily accepted. Nevertheless, we do believe that big data analysis and artificial intelligence including machine learning techniques will have a big impact on appointment systems. Rather than in developing complex ASRs, the surplus value will mainly lie in the discovery of "hidden patterns" that allow for the development of better simple ASRs. For instance, patient groups could be identified based on medical data and patient characteristics as age, work status, etc. Relatively simple ASRs could be derived that exploit this information; e.g., a patient of group A has a higher chance of arriving late and should be scheduled earlier. In

contrast to "black box" derived appointment schedules, the discovery of such hidden, but understandable patterns can also be published and consequently be used to improve other appointment systems than the ones that discovered the pattern. With this in mind, we believe that the importance of efficient algorithms to implement and evaluate (simple) ASRs will only increase in the near future, when big data analysis and machine learning will allow to detect hidden patterns and relations that could be exploited in simple ASRs.

References

Ahmadi-Javid, A., Jalali, Z., & Klassen, K. J. (2017). Outpatient appointment systems in healthcare: A review of optimization studies. *European Journal of Operational Research*, *258*(1), 3–34.

Alaeddini, A., Yang, K. K., Reeves, P., & Chandan, K. R. (2015). A hybrid prediction model for no-shows and cancellations of outpatient appointments. *IIE Transactions on Healthcare Systems Engineering*, *5*(1), 14–32.

Babes, M., & Sarma, G. V. (1991). Out-patient queues at the Ibn-Rochd health centre. *The Journal of the Operational Research Society*, *42*(10), 845–855.

Bailey, N.T. (1952). A study of queues and appointment systems in hospital outpatient departments, with special reference to waiting-times. *Journal of the Royal Statistical Society, Series B*, *14*(2), 185–199.

Bhattacharjee, P., Ray, P. K. (2016). Simulation modelling and analysis of appointment system performance for multiple classes of patients in a hospital: A case study. *Operations Research for Health Care*, *8*, 71–84.

Biskup, D., Herrmann, J., & Gupta, J. N. D. (2008). Scheduling identical parallel machines to minimize total tardiness. *International Journal of Production Economics*, *115*(1), 134–142.

Blanco White, M. J., & Pike, M. C. Appointment systems in out-patients' clinics and the effect of patients' unpunctuality. *Medical Care*, *2*(3), 133–145.

Cardoen, B., Demeulemeester, E., & Beliën, J. (2009). Sequencing surgical cases in a day-care environment: An exact branch-and-price approach. *Computers & Operations Research*, *36*(9), 2660–2669.

Cayirli, T., & Veral, E. (2003). Outpatient scheduling in health care: A review of literature. *Production and Operations Management*, *12*(4), 519–549.

Cayirli, T., Veral, E., & Rosen, H. (2006). Designing appointment scheduling systems for ambulatory care services. *Health Care Management Science*, *9*(1), 47–58.

Cayirli, T., Veral, E., & Rosen, H. (2008). Assessment of patient classification in appointment system design. *Production and Operations Management*, *17*(3), 338–353.

Cayirli, T., Yang, K. K., & Quek, S. A. (2012). A universal appointment rule in the presence of no-shows and walk-ins. *Production and Operations Management*, *21*(4), 682–697.

Cayirli, T., & Yang, K. K. (2014). A universal appointment rule with patient classification for service times, no-shows, and walk-ins. *Service Science*, *6*(4), 274–295.

Chakraborty, S., Muthuraman, K., & Lawley, M. (2010). Sequential clinical scheduling with patient no-shows and general service time distributions. *IIE Transactions*, *42*(5), 354–366.

Chen, Y., & Du, J. (2015). Super-Efficiency in Data Envelopment Analysis. In J. Zhu (Eds.), *Data Envelopment Analysis* (pp. 381–414). Springer, Boston.

Cherchye, L., Moesen, W., Rogge, N., Van Puyenbroeck, T., Saisana, M., Saltelli, A., Liska, R., & Tarantola, S. (2008). Creating composite indicators with DEA and robustness analysis: The case of the technology achievement index. *Journal of the Operational Research Society*, *59*(2), 239–251.

Cook, W. D., & Seiford, L. M. (2009). Data envelopment analysis (DEA) - Thirty years on. *European Journal of Operational Research*, *192*(1), 1–17.

Cooper, W. W., Ruiz, J. L., & Sirvent, I. (2007). Choosing weights from alternative optimal solutions of dual multiplier models in DEA. *European Journal of Operational Research*, *180*(1), 443–458.

Creemers, S. (2009a). *Appointment-driven queueing systems*. (Unpublished doctoral dissertation). KU Leuven, Leuven, Belgium.

Creemers, S., & Lambrecht, M. R. (2009b). An advanced queueing model to analyze appointment-driven service systems. *Computers & Operations Research*, *36*(10), 2773–2785.

Creemers, S., & Lambrecht, M. R. (2010). Queueing models for appointment-driven systems. *Annals of Operations Research*, *178*(1), 155–172.

Deceuninck, M., Fiems, D., & De Vuyst, S. (2018). Outpatient scheduling with unpunctual patients and no-shows. *European Journal of Operational Research*, *165*(1), 195–207.

Dellaert, N. P., & Melo, M. T. (1998). Make-to-order policies for a stochastic lot-sizing problem using overtime. *International Journal of Production Economics*, *56–57*, 79–97.

Denton, B., & Gupta, D. (2003). A sequential bounding approach for optimal appointment scheduling. *IIE Transactions*, *35*(11), 1003–1016.

Doyle, J., & Green, R. (1994). Efficient and cross-efficiency in DEA - Derivations, meanings and uses. *Journal of the Operational Research Society*, *45*(5), 567–578.

Erdogan, S. A., & Denton, B. (2013). Dynamic appointment scheduling of a stochastic server with uncertain demand. *INFORMS Journal on Computing*, *25*(1), 116–132.

Fetter, R. B., & Thompson, J. D. (1966). Patients' waiting time and doctors' idle time in the outpatient setting. *Health Services Research*, *1*(1), 66–90.

Fries, B. E., & Marathe, V. P. (1981). Determination of optimal variable-sized multiple-block appointment systems. *Operations Research*, *29*(2), 324–345.

Giuliano, G., & O'Brien, T. (2007). Reducing port-related truck emissions: the terminal gate appointment system at the ports of Los Angeles and Long Beach. *Transportation Research Part D*, *12*(7), 460–473.

Green, V. L. (2014). Using operations research to reduce delays for healthcare. *INFORMS Tutorials in Operations Research*, 1–16.

Grote, K. D., Newman, J. R., & Sutaria, S. S. (2007). A better hospital experience. *The McKinsey Quarterly*, *11*, 1–11.

Gupta, D., & Denton, B. (2008). Appointment scheduling in health care: Challenges and opportunities. *IIE Transactions*, *40*(9), 800–819.

Hall, R. W. (2012). *Handbook of healthcare system scheduling*. Springer.

Hall, R., & Partyka, J. (2016). Scheduling for better healthcare. *OR/MS Today*, emph39(3). Retrieved from https://www.informs.org/ORMS-Today/Public-Articles/June-Volume-39-Number-3/Scheduling-for-better-healthcare2

Ho, C. J., & Lau, H. S. (1992). Minimizing total-cost in scheduling outpatient appointments. *Management Science*, *38*(12), 1750–1764.

Ho, C. J., & Lau, H. S. (1998). Evaluating the impact of operating conditions on the performance of appointment scheduling rules in service systems. *European Journal of Operational Research*, *112*(3), 542–553.

Jerbi, B., & Kamoun, H. (2011). Multiobjective study to implement outpatient appointment system at Hedi Chaker Hospital. *Simulation Modelling Practice and Theory*, *19*(5), 1363–1370.

Kaandorp, G. C., & Koole, G. (2007). Optimal outpatient appointment scheduling. *Health Care Management Science*, *10*(3), 217–229.

Klassen, K. J., & Rohleder, T. R. (1996). Scheduling outpatient appointments in a dynamic environment. *Journal of Operations Management*, *14*(2), 83–101.

Klassen, K. J., & Yoogalingam, R. (2009). Improving performance in outpatient services with a simulation optimization approach. *Production and Operations Management*, *18*(4), 447–458.

Klassen, K. J., & Yoogalingam, R. (2013). Appointment system design with interruptions and physician lateness. *International Journal of Operations & Production Management*, *33*(4), 394–414.

Klassen, K. J., & Yoogalingam, R. (2014). Strategies for appointment policy design with patient unpunctuality. *Decision Sciences*, *45*(5), 881–911.

Kuiper, A., & Mandjes, M. (2015). Appointment scheduling in tandem-type service systems. *Omega*, *57*(B), 145–156.

Lawley, M., Parmeshwaran, V., Richard, J. P., Turkcan, A., Dalal, M., & Ramcharan, D. (2008). A time-space scheduling model for optimizing recurring bulk railcar deliveries. *Transportation Research Part B*, *42*(5), 438–454.

Lee, S., Min, D., Ryu, J. H., & Yih, Y. (2013). A simulation study of appointment scheduling in outpatient clinics: Open access and overbooking. *Simulation*, *89*(12), 1459–1473.

Lian, J., DiStefano, K., Shields, S., Heinichen, C., Giampietri, M., & Wang, L. (2010). Clinical appointment process, improvement through schedule defragmentation. *IEEE Engineering in Medicine and Biology Magazine*, *29*(2), 127–134.

Liao, C. J., Pegden, C. D., & Rosenshine, M. (1993). Planning timely arrivals to a stochastic production or service system. *IIE Transactions*, *25*(5), 63–73.

Liu, L., & Liu, X. (1998a). Block appointment systems for outpatient clinics with multiple doctors. *The Journal of the Operational Research Society*, *49*(12), 1254–1259.

Liu, L., & Liu, X. (1998b). Dynamic and static job allocation for multi-server systems. *IIE Transactions*, *30*(9), 845–854.

Luo, J., Kulkarni, V. G., & Serhan, Z. (2012). Appointment scheduling under patient no-shows and service interruptions. *Manufacturing & Service Operations Management*, *14*(4), 670–684.

Madas, M. A., & Zografos, K. G. (2006). Airport slot allocation: From instruments to strategies. *Journal of Air Transport Management*, *12*(2), 53–62.

Madas, M. A., & Zografos, K. G. (2008). Airport capcity vs. demand: Mismatch or mismanagement? *Transportation Research Part A*, *42*(1), 203–226.

Markovits-Somogyi, R. (2011). Ranking efficient and inefficient decision making units in data envelopment analysis. *International Journal for Traffic and Transport Engineering*, *1*(4), 245–256.

Mercer, A. (1960). A queueing problem in which the arrival times of the customers are scheduled. *Journal of the Royal Statistical Society, Series B*, *20*(1), 108–113.

Mercer, A. (1973). Queues with scheduled arrivals: A corretion, simplification and extension. *Journal of the Royal Statistical Society, Series B*, *35*(1), 104–116.

Mondschein, S., & Weintraub, G. Y. (2003). Appointment policies in service operations: A critical analysis of the economic framework. *Production and Operations Management*, *12*(2), 266–286.

Namboothiri, R., & Erera, A. L. (2008). Planning local container drayage operations given a port access appointment system. *Transportation Research Part E*, *44*(2), 185–202.

Pegden, C. D., & Rosenshine, M. (1990). Scheduling arrivals to queues. *Computers & Operations Research*, *17*(4), 343–348.

Portela, M., & Thanassoulis, E. (2006). Zero weights and non-zero slacks: Different solutions to the same problem. *Annals of Operations Research*, *145*(1), 129–147.

Rising, E. J., Baron, R., & Averill, B. (1973). A systems analysis of a university-health-service outpatient clinic. *Operations Research*, *21*(5), 1030–147.

Robinson, L. W., & Chen, R. R. (2003). Scheduling doctors' appointments: Optimal and empirically-based heuristic policies. *IIE Transactions*, *35*(3), 295–307.

Rohleder, T. R., & Klassen, K. J. (2000). Using client-variance information to improve dynamic appointment scheduling performance. *Journal of Operations Management*, *28*(3), 293–305.

Rose, C., & Yates, R. (1995). Scheduling arrivals to queues for minimum average blocking: The $S(n)/M/C/C$ system. *Computers & Operations Research*, *22*(8), 793–806.

Sabria, F., & Daganzo, C. F. (1989). Approximate expressions for queueing systems with scheduled arrivals and established service order. *Transportation Science*, *23*(3), 159–165.

Samorani, M., & Ganguly, S. (2016). Optimal sequencing of unpunctual patients in high-service-level clinics. *Production and Operations Management*, *25*(2), 330–346.

Schuetz, H. J., & Kolisch, R. (2012). Approximate dynamic programming for capacity allocation in the service industry. *European Journal of Operational Research*, *218*(1), 239–250.

Sickinger, S., & Kolisch, R. (2009). The performance of a generalized Bailey-Welch rule for outpatient appointment scheduling under inpatient and emergency demand. *Health care Management Science*, *12*(4), 408–419.

Smart, N. A., & Titus, T. T. (2011). Outcomes of early versus late nephrology referral in chronic kidney disease: A systematic review. *American Journal of Medicine*, *124*(11), 1073–1080.

Soriano, A. (1966). Comparison of two scheduling systems. *Operations Research*, *14*(3), 388–397.

Tai, G., & Williams, P. (2012). Optimization of scheduling patient appointments in clinics using a novel modelling technique of patient arrival. *Computer Methods and Programs Biomedicine*, *108*(2), 467-476.

Vanden Bosch, P. M., & Dietz, D. C. (2000). Minimizing expected waiting in a medical appointment system. *IIE Transactions*, *32*(9), 841–848.

Vanden Bosch, P. M., & Dietz, D. C. (2001). Scheduling and sequencing arrivals to an appointment system. *Journal of Service Research*, *4*(1), 15–25.

van Leeuwaarden, J., Denteneer, D., & Resing, J. (2006). A discrete-time queueing model with periodically scheduled arrival and departure slots. *Performance Evaluation*, *63*(4–5), 278–294.

Vissers, J. M. H. (1979). Selecting a suitable appointment system in an outpatient setting. *Medical Care*, *17*(12), 1207–1220.

Wang, P. P. (1993). Static and dynamic scheduling of customer arrivals to a single-server system. *Naval Research Logistics*, *40*(3), 345–360.

Wang, P. P. (1997). Optimally scheduling N customer arrival times for a single-server system. *Computers & Operations Research*, *24*(8), 703–716.

Welch, J., & Bailey, N. (1952). Appointment systems in hospital outpatient departments. *The Lancet*, *259*(6718), 1105–1108.

Welch, J. D. (1964). Appointment systems in hospital outpatient departments. *Operations Research Quarterly*, *15*(3), 224–232.

Wendler, E. (2007). The scheduled waiting time on railway lines. *Transportation Research Part B*, *41*, 148–158.

Yan, S., & Lai, W. (2007). An optimal scheduling model for ready mixed concrete supply with overtime considerations. *Automation in Construction*, *16*(6), 734–744.

Zacharias, C., & Pinedo, M. (2012). Appointment scheduling with no-shows and overbooking. *Production and Operations Management*, *23*(5), 788–801.