Performance Evaluation 80 (2014) 5–26

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

Designing cyclic appointment schedules for outpatient clinics with scheduled and unscheduled patient arrivals

Nikky Kortbeek[a,b,c,*], Maartje E. Zonderland[a,b], Aleida Braaksma[a,b,c], Ingrid M.H. Vliegen[b,d], Richard J. Boucherie[a,b], Nelly Litvak[a,b], Erwin W. Hans[b,d]

[a] Stochastic Operations Research, University of Twente, Postbox 217, 7500 AE Enschede, The Netherlands

[b] Center for Healthcare Operations Improvement and Research, University of Twente, The Netherlands

[c] Department of Quality and Process Innovation, Academic Medical Center Amsterdam, Meibergdreef 9, 1105 AZ Amsterdam, The Netherlands

[d] Department Industrial Engineering and Business Information Systems, University of Twente, The Netherlands

ARTICLE INFO

Article history:
Available online 5 July 2014

Keywords:
Health care management
Service operations
Appointment scheduling
Queuing theory
Stochastic methods

ABSTRACT

We present a methodology to design appointment systems for outpatient clinics and diagnostic facilities that offer both walk-in and scheduled service. The developed blueprint for the appointment schedule prescribes the number of appointments to plan per day and the moment on the day to schedule the appointments. The method consists of two models; one for the day process that governs scheduled and unscheduled arrivals on the day and one for the access process of scheduled arrivals. Appointment schedules that balance the waiting time at the facility for unscheduled patients and the access time for scheduled patients are calculated iteratively using the outcomes of the two models. Two methods to calculate appointment schedules, complete enumeration and a heuristic procedure, are compared in various numerical experiments. Furthermore, an appointment schedule for the CT-scan facility at the Academic Medical Center Amsterdam, The Netherlands, is developed to demonstrate the practical merits of the methodology. The method is of general nature and can therefore also be applied to scheduling problems in other sectors than health care.

1. Introduction

Developing appointment schedules for service facilities that process both scheduled and unscheduled arrivals is challenging, as it requires planning and scheduling on different time scales. A well-designed appointment system comprises an efficient day appointment schedule and provides timely access. This article is motivated by challenges faced by hospital outpatient clinics that serve patients on a walk-in basis. Most of these clinics also have a limited number of appointment slots. There are various organizational (e.g., fixed slots for patients in a care pathway, patients with long travel time to the hospital, children) and medical (e.g., local anesthesia or contrast fluid required) reasons to give a patient an appointment. In this article, we introduce a method to design appointment schedules for such facilities. To illustrate the method, we also design an appointment schedule for the Computed Tomography (CT) scan facility at the radiology department of the Academic Medical Center (AMC) in Amsterdam, a Dutch teaching hospital. At the CT-scan facility, where approximately 11,000

[*] Corresponding author at: Stochastic Operations Research, University of Twente, Postbox 217, 7500 AE Enschede, The Netherlands. Tel.: +31 205666368.
E-mail addresses: n.kortbeek@utwente.nl, n.kortbeek@amc.nl (N. Kortbeek), m.e.zonderland@utwente.nl (M.E. Zonderland).

http://dx.doi.org/10.1016/j.peva.2014.06.003
0166-5316/

diagnostic examinations per year are performed, currently an appointment system is employed. Management considers the implementation of a mixed walk-in and appointment system.

Advantages of a walk-in system are a higher level of accessibility and more freedom for patients to choose the date and time of their hospital visit. Disadvantages are a possible highly variable demand, and as a consequence low utilization and high waiting time (the time between the physical arrival at the facility and the start of consultation and/or treatment). The advantage of an appointment system is that workload can be dispersed, while it has the disadvantage of a potentially long access time (the time between the day of the appointment request and the appointment date). Since prolonged access times result in a delay of treatment, deterioration of health condition is a serious risk [1]. Allowing patients to walk in effectively reduces access times to zero, and thus increases quality of care. In addition, health care facilities typically aim to guarantee a certain service level with respect to the access time for patients with an appointment.

The challenge in a mixed system is thus to balance access time (for appointment patients) and waiting time (on the day of service). To achieve this, we develop a methodology that schedules appointments when the expected walk-in demand is low. To smoothen the system, in periods of high demand part of the walk-in patients is offered an appointment at a later moment. Walk-in demand [2,3] and demand for appointments requests [4] are often cyclic; therefore, we develop a cyclic appointment schedule. Appointment scheduling has received considerable attention in the literature (see Section 2), in contrary to models that relate access and waiting time [5].

Our contribution is a methodology that incorporates unscheduled and scheduled arrivals and maximizes the number of unscheduled patients served on the day of arrival, while satisfying a pre-specified access time norm for scheduled patients. We model the unscheduled arrivals with a stochastic non-stationary arrival process and incorporate balking behavior. The scheduled patients have priority, may not show up, and appointment requests are assumed to arrive according to a cyclic pattern. To account for the cyclic arrivals, the appointment schemes we develop are also cyclic, where the cycle is a repeating sequence of days. The cycle length can, for instance, be a week or a month. The Cyclic Appointment Schedule (CAS) specifies a capacity cycle (the maximum number of patients that can be scheduled on each day of the cycle) and a day schedule (the maximum number of patients to be scheduled per time slot on each day). Access time and waiting time are measured on different time scales, since access time is counted between days and waiting time during a day.

To facilitate the two time scales, our approach consists of decomposing the appointment planning process and the service process during the day. For both processes we propose an analytical evaluation model. The first model determines the access time for scheduled patients for any given capacity cycle. The second model determines the expected number of unscheduled patients that cannot be seen on the day of arrival. Two methods to calculate appointment schedules, complete enumeration and a heuristic procedure, are compared in various numerical experiments. Furthermore, an appointment schedule for the CT-scan facility at the AMC is developed to demonstrate the practical merits of the methodology.

This article is organized as follows. Section 2 provides a literature review. In Section 3, we give an introduction to the methodology and provide a formal problem description. Sections 4–6 present the access and day process evaluation models and the iterative procedure. Section 7 describes the numerical experiments, followed by the discussion and conclusions in Section 8.

2. Literature

In many service facilities customers are requested to make an appointment. There is a substantial body of literature focusing on the design of appointment systems. Health care is the most prevalent application area and hence most prevalent in the literature (see the surveys [6,5,7]). Appointment systems can be regarded as a combination of two distinct queuing systems. The first queuing system concerns customers making an appointment and waiting until the day the appointment takes place. The second queuing system concerns the process of a service session during a particular day. We denote these two queuing processes as the 'access process' and the 'day process'. The remainder of this section provides an overview of the literature relevant for the present work and is structured as follows: (1) appointment scheduling, (2) access time models, and (3) integrating the access process and the day process.

2.1. Appointment scheduling

Appointment scheduling concerns designing blueprints for day-appointment schedules with typical objectives such as minimizing customer waiting time, and maximizing resource utilization or minimizing resource idle time. A large part of the literature focuses on scheduling a given number of appointments on a particular day (e.g., [8–12]). The extent to which various aspects that impact the performance of an appointment schedule are incorporated varies, such as customer punctuality (e.g., [13]), customers not showing up ('no-shows') (e.g., [14,8]), lateness of the server at the start of a service session (e.g., [11]), service interruptions (e.g., [13]) and the variance of service duration (e.g., [14]).

Research techniques employed in appointment scheduling can be divided into analytical and simulation-based approaches, of which the latter is most widely applied [6]. In the day process we aim for an analytical approach, namely finite-time Markov chain analysis. Related examples with health care applications are [15,16,8,10,17,12], although these references do not consider unscheduled customers.

Often, a homogeneous customer population is assumed [18]. Some studies however, focus on service systems with various customer types. Differentiation between customer types is identified as a consequence of distinct service requirements (e.g., [19–21,12,22]). Also, distinct priority levels may be a reason for patient type differentiation. An example can be found in [23], where service slots are earmarked for various scheduled customer classes. In this article, customer type differentiation arises from distinct arrival processes.

The effect of mixed arrival processes is studied in [24–26]. Here, scheduled outpatients, unscheduled inpatients and emergency patients are taken into account. Patients without an appointment are either emergency patients who require non-preemptive priority or inpatients available for 'call-in' at any time during the day. These unscheduled patients are assumed to arrive according to an equal arrival rate throughout the day. In our case, we consider walk-in patients without priority who cannot be called in during the day. Moreover, we consider non-stationary arrivals to incorporate the expected peak behavior of walk-in demand. Studies that do incorporate non-priority unscheduled arrivals similar to the unscheduled arrivals in this article are [2,27,19,28–31]; however, in all cases a simulation approach is employed. Also, these studies do not incorporate unscheduled customers leaving the facility when the waiting time is too long.

2.2. Access time models

As our approach consists of a decomposition, models solely focusing on access time are also of interest. The access process we consider is discrete-time and cyclical in both the arrival and service processes. Various access time models based on continuous-time queuing models are available. Examples are the $M(t)|M|s(t)$ queue [32] and the adapted $M|M|s$ queue that models time-dependent demand [33]. The latter method is also applied to a health care problem in [34]. To preserve the discrete-time nature we take as starting point the generating function approach for discrete-time queuing models by [35]. A survey on discrete-time queuing systems is presented in [36].

Models to evaluate the length of hospital waiting lists are introduced in [37], and further studied in for example [38]. In these models homogeneous appointment request arrivals are assumed. In polling models, multiple queues are served by one server in cyclic order (see [39] for an overview). However, cyclic arrival rates and cyclic service capacity have not yet been incorporated in polling models.

2.3. Linking the access and the day process

We found only a few examples that jointly consider the access and day process. In [40], the authors propose a two time scale model for the Emergency Department (ED)—Ward patient flow. The fast time scale of the ED is modeled by a continuous-time Markov chain, while the slower time scale of the wards is modeled by a discrete-time Markov chain. In [41,21], appointment schedules ranging over a horizon of several days are evaluated. The aim is to minimize the patient's waiting and the doctor's idle time, but the patient's access time is not studied in detail.

The advanced (or open) access methodology described in [1] also considers two time scales. With advanced access, a clinic leaves a fraction of appointment slots vacant for patients who request an appointment on the same day or within a couple of days. As many patients as possible are scheduled on the day they make an appointment request. One should determine the optimal ratio between the reserved capacity for long-term and same-day appointments [42]. This principle is slightly adapted in [43], where the demand for short-term appointments is distributed over several days, to smooth the daily load of the system. The aim of the advanced access methodology is to minimize access time ("do today's work today"). Note that in an advanced access clinic patients do announce themselves in advance and make a (same-day) appointment, contrary to the type of unscheduled patients we consider, who just show up. Models that study the advanced access methodology usually focus on capacity distribution (e.g., [42,44,45]). In addition, the reduced adverse effect of no-shows by introducing open access is studied [46].

Formulating a model to design an appointment schedule considering two time scales is usually done using simulation techniques (e.g., [47]). An analytic approach is presented in [48], where the effect of capacity allocation among competing patient classes on access time targets is studied using techniques from Markov Decision Modeling and Mathematical Programming. An approach related to ours, although without the presence of walk-in patients, is given in [49]. The authors consider a service facility, and first develop a vacation queuing system to determine the access time. Subsequently an appointment system is developed that calculates the waiting time at the facility.

3. Formal problem description

This section defines all modeling assumptions, defines the Cyclic Appointment Schedule (CAS), formally states the research goal and gives an overview of the proposed approach. Then, Sections 4 and 5 present two models to respectively evaluate the access time to the facility and the day schedule performance. In Section 6, the two models are connected by an iterative procedure, through which the best CAS is computed. Since our approach is generically applicable, we also present the methodology in the generic terms: a facility that serves scheduled and unscheduled jobs. Table 1 summarizes the notation introduced in this section.

Assumptions. A facility consisting of R resources is operational during T time slots of length h, during each day in a cycle of D days. Two types of jobs have to be served: scheduled and unscheduled jobs. Service takes one time slot. Scheduled jobs are given a specific date and time immediately when an appointment is requested. In addition, when the facility is temporarily

Table 1
Notation introduced in Section 3.

Symbol	Description
R	Number of resources
T	Number of time slots during a day
t	Time slot index ($t \in \{1, \dots, T\}$)
h	Length of a time slot
D	Cycle length in days
d	Day index ($d \in \{1, \dots, D\}$)
g	Patience of an unscheduled job, expressed in the number of slots a job is willing to wait
q	$\mathbb{P}$(No-show of a scheduled job)
λ^d	Initial appointment request arrival rate on day d
χ_t^d	Unscheduled job arrival rate on day d during time interval $(t-1, t]$
c_t^d	Maximum number of appointments to schedule in slot t on day d
C^d	Appointment schedule on day d, $C^d = (c_1^d, \dots, c_T^d)$
C	Cyclic appointment schedule, $C = (C^1, \dots, C^D)$
k^d	Maximum number of appointments to schedule on day d
K	Capacity cycle, $K = (k^1, \dots, k^D)$
F	$\mathbb{E}$[Fraction of unscheduled jobs served on the day of arrival during one cycle]
$S(y)$	Access time service level: fraction of jobs with access time not greater than y
$(y, S^{\text{norm}}(y))$	Access time service level requirement: fraction of jobs with access time not greater than y is at least $S(y)$
ϕ^d	Distribution of the number of deferred jobs on day d
γ^d	Total appointment request arrival distribution on day d
ν^d	Expected number of deferred jobs on day d

congested, unscheduled jobs are also offered an appointment: if the service of an unscheduled job cannot start within g time slots after arrival, it leaves the facility and an appointment is planned for another day. We refer to such jobs as *deferred* unscheduled jobs, or just deferred jobs. The first available appointment slot for scheduled and deferred jobs is always the next day at the earliest. All appointments, both scheduled jobs and deferred unscheduled jobs, are scheduled according to a First Come First Served (FCFS) principle. In addition, we allow for no-shows, that is, the probability that a scheduled job actually arrives at the facility equals $1-q$, so that q represents the probability that a job does not show up.

We assume a non-stationary Poisson process for the arrivals of appointment requests, with $\lambda^1, \dots, \lambda^D$ the arrival rates for different days in the cycle. Next, during each day in the cycle, we assume a non-stationary Poisson arrival process for unscheduled job arrivals, with slot-dependent arrival rates: χ_t^d for day $d \in \{1, \dots, D\}$ and time slot $t \in \{1, \dots, T\}$.

Cyclic appointment schedule. To effectively counterbalance the non-stationarity at both the daily and cyclic (i.e., weekly, bi-weekly or monthly) levels, we aim to design an appointment schedule that is cyclic. We introduce the CAS $C = (C^1, \dots, C^D)$, with $C^d = (c_1^d, \dots, c_T^d)$, where c_t^d specifies the maximum number of jobs that may be scheduled in slot t on day d. To avoid waiting for scheduled jobs c_t^d is maximally R.

To find an adequate appointment schedule, we propose a decomposition. First, we introduce the concept of a capacity cycle $K = (k^1, \dots, k^D)$, where k^d prescribes the maximum number of jobs to schedule for day d. Second, given the capacity cycle K, the day plan is specified. In order to match the capacity cycle K, the day plan C^d should be such that $k^d = \sum_{t=1}^{T} c_t^d$.

Goal. An effective strategy balances the opportunities (1) for unscheduled jobs to be served on the same day without long waiting time and (2) for scheduled jobs to be served within an acceptable access time. To this end, we define the best policy as the CAS in which the expected fraction of unscheduled jobs served on the day of arrival, F, is maximized, while for scheduled jobs the access time service level, $S(y)$, defined as the percentage of jobs that is served within y days, is above a pre-specified norm $S^{\text{norm}}(y)$. The value of the vector $(y, S^{\text{norm}}(y))$ is chosen by facility management.

Approach. The best CAS is determined by employing an iterative procedure that effectively utilizes our decomposition of the CAS in the capacity cycle and the day plan. Fig. 1 provides an overview of the iterative procedure.

In each iteration, first, capacity cycles are generated with at most $R \cdot T$ appointments per day, for which the access time service level norm is satisfied. All jobs requesting an appointment are taken into account—thus both scheduled jobs and deferred unscheduled jobs. We derive the distribution of the number of deferred unscheduled jobs ϕ^d, such that the distribution of the total number of appointment requests on day d is the sum of a Poisson distribution with parameter λ^d and the distribution ϕ^d. To assess whether specific capacity cycles satisfy the access time norm $S^{\text{norm}}(y)$, a discrete-time cyclic queuing model is proposed (Model I, presented in Section 4).

Next, for each capacity cycle generated in the first step, the best day schedule is determined. Given the queue length probabilities resulting from Model I and the unscheduled job arrival rates, χ_t^d, for each day the k^d appointments are distributed over the T time slots, such that the number of deferred unscheduled jobs is minimized. To achieve this, a Markov reward model is presented (Model II, Section 5), which is used to calculate the performance of a specific day schedule.

Then, the capacity cycle that achieves the lowest expected number of deferred unscheduled jobs over the entire cycle is chosen as the best cycle. If the expected numbers of deferred unscheduled jobs ν^d did not change significantly since the last iteration, the procedure stops. If not, the entire process is repeated. A detailed description of the iterative procedure is given in Section 6.

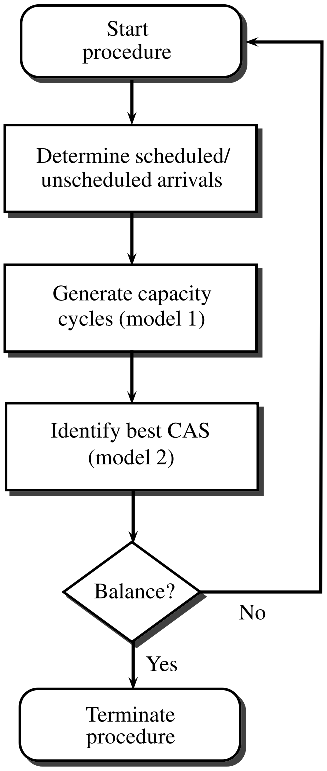

Start procedure
Determine scheduled/ unscheduled arrivals
Generate capacity cycles (model 1)
Identify best CAS (model 2)
Balance?
No
Yes
Terminate procedure

Fig. 1. The iterative procedure.

4. Model I: access time evaluation

In this section, a discrete-time cyclic queuing model is presented that allows for an evaluation of the access time for scheduled jobs, given an arbitrary capacity cycle. To this purpose, we focus on the backlog, B^d, at the start of each day d. We define the backlog as the number of jobs for which a request for an appointment has already been made, while the appointment itself has not yet taken place. We formulate a Lindley-type equation to characterize the backlog, and use a probability generating function approach to derive expressions for the distribution of the backlog at the start of each day in the cycle. From the backlog distribution, we derive the access time distribution. A summary of the notation used in this section is given in Table 2.

Lindley-type equation. Consider day d. During the day, a maximum number of jobs, k^d, is served, and a number of new jobs, A^d, arrives. At the start of day d, there is a backlog B^d. Since it is not possible to make an appointment on the day of arrival itself, the backlog at the start of the next day equals the backlog on day d minus the number of jobs served on day d plus the number of jobs that arrived on day d. This can be formalized in the following Lindley-type equation:

$$B^{d+1} = (B^d - k^d)^+ + A^d,$$

where $(x)^+ = x$ if $x > 0$, and 0 otherwise.

A generating function approach. Using an approach based on generating functions [35], we derive expressions for the distribution of the backlog at the start of each day in the cycle. The transition probabilities for going from state $B^d = i$ to state $B^{d+1} = i'$ are given by:

$$\mathbb{P}\left(B^{d+1} = i' | B^d = i\right) = \begin{cases} \mathbb{P}\left(A^d = i'\right) & \text{if } i - k^d \leq 0 \\ \mathbb{P}\left(A^d = i' - i + k^d\right) & \text{if } i - k^d > 0. \end{cases}$$

Let π_j^d denote the stationary probability that at the start of day d, the backlog equals j jobs. Furthermore, let a_j^d denote the probability that $A^d = j$. Note that the underlying probability distribution does not necessarily have to be Poisson. The stationary probabilities can be computed recursively, under the condition that the capacity for scheduled jobs is larger than the average demand, i.e., $\sum_{d=1}^{D} \mathbb{E}[A^d] < \sum_{d=1}^{D} k^d$, since otherwise we would be dealing with an unstable system. For $d \in \{1, \ldots, D\}$ and $j \geq 0$ we obtain:

$$\pi_j^{d+1} = a_j^d \sum_{i=0}^{k^d - 1} \pi_i^d + \sum_{r=0}^{j} a_{j-r}^d \pi_{k^d+r}^d. \tag{1}$$

We multiply both sides of (1) with the complex number $z^{(j)}$, where $|z| \leq 1$, and $z^{(j)}$ denotes z raised to the power j, as opposed to index d in π_j^d, a_j^d and k^d. The summation of both sides of the resulting equation over j yields the probability generating function for π^{d+1}:

$$P_{B^{d+1}}(z) = \sum_{j=0}^{\infty} \pi_j^{d+1} z^{(j)} = \sum_{j=0}^{\infty} \left(a_j^d \sum_{i=0}^{k^d-1} \pi_i^d + \sum_{r=0}^{j} a_{j-r}^d \pi_{k^d+r}^d \right) z^{(j)}.$$

From this we obtain:

$$P_{B^{d+1}}(z) = \sum_{j=0}^{\infty} \pi_j^{d+1} z^{(j)} = P_{A^d(z)} z^{(-k^d)} P_{B^d(z)} + P_{A^d(z)} z^{(-k^d)} \sum_{i=0}^{k^d-1} \pi_i^d \left(z^{(k^d)} - z^{(i)} \right).$$

Rearranging terms and changing the order of summation leads to the probability generating function of B^d:

$$P_{B^d}(z) = \frac{\sum\limits_{i=1}^{D} \sum\limits_{r=0}^{k^{d+D-i}-1} (z^{(k^{d+D-i})} - z^{(r)}) \pi_r^{d+D-i} \left[\prod\limits_{s=d}^{d+D-i-1} z^{(k^s)} \prod\limits_{r=0}^{i-1} P_{A^{d+D-r-1}}(z) \right]}{\prod\limits_{g=1}^{D} z^{(k^g)} - \prod\limits_{h=1}^{D} P_{A^h}(z)},$$

where, since we consider days in a repeating cycle, we define:

$$d := \begin{cases} D, & d \bmod D = 0 \\ d \bmod D, & \text{otherwise.} \end{cases}$$

The generating functions uniquely determine the stationary probabilities $\pi_j^d, j \in \{0, \ldots, k^d - 1\}, d \in \{1, \ldots, D\}$. To calculate these probabilities, we build upon the approach given in [50]. Define k as the total number of available appointment slots in a capacity cycle, i.e., $k = \sum_{d=1}^{D} k^d$. Then, the denominator of $P_{B^d}(z)$ has $k - 1$ zeros inside the unit disk; this can be shown by using Rouché's theorem [51]. All generating functions, including $P_{B^d}(z)$, are bounded for $|z| \leq 1$, and therefore the zeros of the denominator are also zeros of the numerator [35]. Thus we obtain $k - 1$ equations, and use $P_{B^d}(1) = 1$ to secure the last equation. The $k - 1$ zeros of the denominator of $P_{B^d}(z)$ are found by solving:

$$\prod_{r=1}^{D} z^{(k^r)} - \prod_{h=1}^{D} P_{A^h}(z) = 0. \tag{2}$$

The solutions of (2) also represent zeros of the numerator. Together with the normalizing equation $P_{B^d}(1) = 1$, $P_{B^d}(z)$ is completely defined for $d = 1, \ldots, D$. Note that now only the backlog probabilities for $j \in \{0, \ldots, k^d - 1\}$, have been derived. The remaining backlog probabilities are calculated directly using (1).

Performance measures. The access time distribution can be directly derived from the backlog probabilities, since appointment requests are served according to the FCFS principle. The FCFS service order and the impossibility of making an appointment request for the day of arrival result in an access time of at least one day. Several performance measures can be derived. Of particular interest are the probability distribution of the access time, the expected access time and the access time service level.

1. *The probability distribution of the access time.* First we derive the conditional access time probability that the access time for a client arriving on day d exceeds y days, given that the backlog at the start of day d equals b clients. As argued, for $y = 0$, we have that

$$\mathbb{P}[W^d > y | B^d = b] = 1 \quad \forall b.$$

For $y > 0$, we have that

$$\mathbb{P}[W^d > y | B^d = b] = \begin{cases} 1 & \text{if } b \geq \sum\limits_{i=0}^{y} k^{d+i} \\ \dfrac{\sum\limits_{j=s+1}^{\infty} (j - s) \cdot \mathbb{P}[A^d = j]}{\mathbb{E}[A^d]} & \text{otherwise,} \end{cases} \tag{3}$$

where s represents the number of jobs arrived on day d that is served within y days:

$$s = \min \left\{ \sum_{i=1}^{y} k^{d+i}, \sum_{i=0}^{y} k^{d+i} - b \right\}.$$

We can explain formula (3) as follows. First, when the backlog b outnumbers the available capacity in y days, the conditional probability that the access time exceeds y days equals 1. Otherwise, all arrivals beyond the number s wait for more than y

Table 2
Notation introduced in Section 4.

Symbol	Description
B^d	Backlog at start of day d
$P_{B^d}(z)$	Generating function of B^d
A^d	Number of appointment requests arriving on day d
a_j^d	Appointment request arrival probabilities, $\mathbb{P}\left(A^d = j\right)$
$P_{A^d}(z)$	Generating function of A^d
π_j^d	Stationary backlog probabilities, $\mathbb{P}\left(B^d = j\right)$
k	Total number of available appointment slots in a capacity cycle, $k = \sum_{d=1}^{D} k^d$
$\mathbb{E}[W^d]$	$\mathbb{E}$[Access time for an appointment request arriving on day d]
$\mathbb{E}[W]$	$\mathbb{E}$[Access time for an arbitrary appointment request]

days. There are $j—s$ such arrivals. Then, the probability that the access time for a client arriving on day d exceeds y days, equals

$$\mathbb{P}[W^d > y] = \sum_{b=0}^{\infty} \mathbb{P}[W^d > y|B^d = b] \cdot \mathbb{P}[B^d = b].$$

2. *The expected access time*. Analogously, the expected access time for an appointment request that arrives on day d is computed with:

$$\mathbb{E}[W^d|B^d = b] = \sum_{y=0}^{\infty} \mathbb{P}[W^d > y|B^d = b],$$

and thus

$$\mathbb{E}[W^d] = \sum_{b=0}^{\infty} \mathbb{E}[W^d|B^d = b] \cdot \mathbb{P}[B^d = b],$$

and

$$\mathbb{E}[W] = \sum_{d=1}^{D} \mathbb{E}[W^d] \frac{\mathbb{E}[A^d]}{\sum\limits_{r=1}^{D} \mathbb{E}[A^r]}.$$

3. *The access time service level*. Using the access time probability distribution, we determine the fraction of scheduled jobs for which the access time does not exceed y. We define this as follows:

$$S(y) = \sum_{d=1}^{D} \left(1 - \mathbb{P}[W^d > y]\right) \frac{\mathbb{E}[A^d]}{\sum\limits_{r=1}^{D} \mathbb{E}[A^r]}.$$

5. Model II: day process evaluation

In this section, we present a model to evaluate the performance of a single day in the CAS. Recall that the CAS consists of a capacity cycle, $K = (k^1, \ldots, k^D)$, that prescribes the maximum number of jobs that can be scheduled for day d. Using Model I, we are able to evaluate the access time performance of a given capacity cycle. In this section, we evaluate the day process of a given appointment schedule, by formulating a Markov reward process.

Note that although day appointment schedule C^d is open for scheduling appointments, there may be less backlog than the $k^d = \sum_{t=1}^{T} c_t^d$ available appointment slots. Therefore, we introduce the notation $\widetilde{C}^d$ to represent the *realized* day planning, which is the schedule we evaluate. Now, $\widetilde{C}^d = \left(\widetilde{c}_1^d, \ldots, \widetilde{c}_T^d\right)$ expresses the actually utilized appointment slots. Since appointments are planned on a FCFS basis, the realized appointment day schedule, $\widetilde{C}^d$, is always a 'bottom-up filled' version of the day schedule, C^d. Of course, unoccupied appointment slots can be used for unscheduled jobs.

Since we consider the day performance on a day-by-day basis, in the remainder of this section we drop the superscript d for notational convenience. Table 3 provides a summary of the notation introduced in this section.

Assumptions. For clarity of presentation, some of the assumptions introduced in Section 3 are repeated. During one day the facility of R resources is operational during T intervals of length h. Two types of jobs have to be served: scheduled and unscheduled jobs. Service always takes one time slot of length h. At the beginning of each time slot, a service can start. If there are both scheduled and unscheduled jobs, scheduled jobs are given priority. Overtime is not allowed.

Scheduled jobs arrive on time, according to the schedule $\widetilde{C}$. Unscheduled jobs arrive at the facility according to an inhomogeneous Poisson process with slot-dependent arrival rate χ_t. If the service of an unscheduled job cannot start within g time slots after arriving, it leaves the facility and an appointment is planned for another day. The decision to defer an

Table 3
Notation introduced in Section 5.

Symbol	Description
$\widetilde{C}$	Realized schedule under CAS C, $\widetilde{C} = (\widetilde{C}^1, \dots, \widetilde{C}^D)$, $\widetilde{C}^d = \left(\widetilde{c}_1^d, \dots, \widetilde{c}_T^d\right)$
$e_{t,g}$	Number of slots available for unscheduled jobs in the next g intervals after time t
$p_t^s(s)$	$\mathbb{P}$(Number of scheduled jobs arriving at the start of slot $t = s$)
$p_t^u(u)$	$\mathbb{P}$(Number of unscheduled jobs arriving during interval $(t-1, t] = u$)
$\mathbb{P}\left[(s, u)_{t+1} \mid (k, l)_t\right]$	Transition probability from state (t, k, l) to state $(t+1, s, u)$
$Q_t(s, u)$	$\mathbb{P}$(Number of scheduled, unscheduled jobs waiting at the start of slot $t = s, u$)
ν_t	$\mathbb{E}$[Number of deferred jobs in time interval $(0, t]$]
ν	$\mathbb{E}$[Total number of deferred jobs]
ϕ_t	Distribution of the number of deferred jobs in time interval $(t-1, t]$
ϕ	Distribution of the total number of deferred jobs

unscheduled job is based on the anticipated number of free slots. We assume that the facility has no pre-knowledge about potential no-shows. Therefore, an unscheduled job arriving during interval $(t-1, t]$ stays if – and only if – the number of unscheduled jobs already waiting is strictly smaller than the minimum number of service slots during the upcoming g intervals that are not utilized by scheduled jobs. The number of time slots anticipated to be available for unscheduled jobs during the upcoming g intervals is denoted by $e_{t,g}$:

$$e_{t,g} = \sum_{j=t}^{\min\{t+g-1,T\}} (R - \widetilde{c}_j). \tag{4}$$

States. The state of the system is denoted by the tuple (t, s, u), which specifies that at the beginning of time slot t, s scheduled and u unscheduled jobs are present.

Transition probabilities. Let $p_t^s(s)$ denote the probability that s scheduled jobs arrive at the beginning of time slot t. Since each no-show is assumed to occur independently, these probabilities are calculated as follows (recall that q denotes the no-show probability):

$$p_t^s(s) = \begin{cases} \binom{\widetilde{c}_t}{s} (1-q)^s (q)^{\widetilde{c}_t - s}, & 0 \le s \le \widetilde{c}_t \\ 0, & s > \widetilde{c}_t. \end{cases}$$

Let $p_t^u(u)$ denote the probability that u unscheduled jobs arrive during time interval $(t-1, t]$. As specified, $p_t^u(u)$ is Poisson distributed with slot-dependent parameter χ_t. Note that χ_1 represents the arrival rate of unscheduled jobs that arrive before the opening time of the facility. Furthermore, note that any distribution function p_t^u can be used in the day process evaluation model. Therefore, for Model II the assumption of a Poisson arrival process is not strictly required.

Let $\mathbb{P}\left[(s, u)_{t+1} \mid (v, w)_t\right]$ denote the transition probability of jumping from state (t, v, w) to $(t+1, s, u)$. Below we specify these transition probabilities for all possible events. In Fig. 2, the state space for an arbitrary time slot t is displayed in which the seven different possible events (a)–(g) are indicated. The events are separated into three groups: first, cases (a)–(c) in which no scheduled job is served ($v = 0$), second, cases (d) and (e) in which both scheduled and unscheduled jobs are served ($v < R$), and third, cases (f) and (g) in which only scheduled jobs are served ($v \ge R$). As a clarification on how the system evolves when no-shows occur, recall that unscheduled jobs arrive in the time interval $(t-1, t]$ and the decision of acceptance is made directly at the time of arrival based on the policy described above. So at time t no-shows can be observed, and only the unscheduled jobs that are still in the queue can be served in a slot which was at first reserved for an appointment slot but now released due to a no-show. In the expressions below, $\mathbb{1}_A$ represents the indicator function; $\mathbb{1}_A = 1$ if condition A is satisfied, and 0 otherwise.

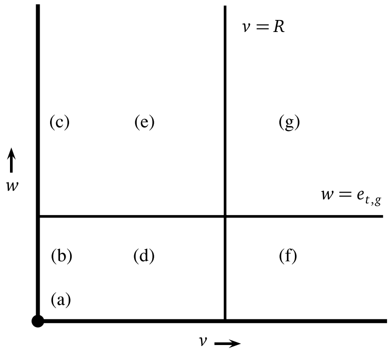


Fig. 2. Day process state space and events.

Case(a) $v = w = 0$; no job served:

$$\mathbb{P}\left[(s,u)_{t+1} \mid (v,w)_t\right] = p^s_{t+1}(s)p^u_{t+1}(u).$$

Case(b) $v = 0,\ 0 < w \leq e_{t,g}$; unscheduled job(s) served:

$$\mathbb{P}\left[(s,u)_{t+1} \mid (v,w)_t\right] = p^s_{t+1}(s)p^u_{t+1}(u - w + \min\{R, w\})\mathbb{1}_{(u\geq w-\min\{R,w\})}.$$

Case(c) $v = 0,\ w > e_{t,g}$; unscheduled job(s) served, unscheduled job(s) deferred:

$$\mathbb{P}\left[(s,u)_{t+1} \mid (v,w)_t\right] = p^s_{t+1}(s)p^u_{t+1}(u - e_{t,g} + R)\mathbb{1}_{(u\geq e_{t,g}-R)}.$$

Case(d) $0 < v < R,\ w \leq e_{t,g}$; scheduled and unscheduled job(s) served:

$$\mathbb{P}\left[(s,u)_{t+1} \mid (v,w)_t\right] = p^s_{t+1}(s)p^u_{t+1}(u - w + \min\{(R-v), w\})\mathbb{1}_{(u\geq w-\min\{(R-v),w\})}.$$

Case(e) $0 < v < R,\ w > e_{t,g}$; scheduled and unscheduled job(s) served, unscheduled job(s) deferred:

$$\mathbb{P}\left[(s,u)_{t+1} \mid (v,w)_t\right] = p^s_{t+1}(s)p^u_{t+1}(u - e_{t,g} + R - v)\mathbb{1}_{(u\geq e_{t,g}-R+v)}.$$

Case(f) $v = R,\ w \leq e_{t,g}$; scheduled job(s) served:

$$\mathbb{P}\left[(s,u)_{t+1} \mid (v,w)_t\right] = p^s_{t+1}(s - v + R)p^u_{t+1}(u - w)\mathbb{1}_{(s\geq v-R)}\mathbb{1}_{(u\geq w)}.$$

Case(g) $v = R,\ w > e_{t,g}$; scheduled job(s) served, unscheduled job(s) deferred:

$$\mathbb{P}\left[(s,u)_{t+1} \mid (v,w)_t\right] = p^s_{t+1}(s - v + R)p^u_{t+1}(u - e_{t,g})\mathbb{1}_{(s\geq v-R)}\mathbb{1}_{(u\geq e_{t,g})}.$$

Performance measures. Let $Q_t(s, u)$ denote the probability that at the start of slot t there are s scheduled and u unscheduled jobs present. $Q_t(s, u)$ can be calculated as follows:

$$Q_1(s,u) = p^s_1(s) \cdot p^u_1(u).$$

For $t = 2, \dots, T$:

$$Q_{t+1}(s,u) = \sum_{v=0}^{\infty}\sum_{w=0}^{\infty} Q_t(v,w)\mathbb{P}\left[(s,u)_{t+1} \mid (v,w)_t\right].$$

The expected number of deferred jobs $\nu = \nu_T$ is calculated accordingly (recall that ν is the total number of deferred jobs that is accumulated at the end of the day and that need an appointment during one of the upcoming days):

$$\nu_1 = \sum_{s=0}^{\infty}\sum_{u=e_{1,g}+1}^{\infty}(u - e_{1,g}) \cdot Q_1(s,u).$$

For $t = 2, \dots, T$:

$$\nu_t = \nu_{t-1} + \sum_{s=0}^{\infty}\sum_{u=e_{t,g}+1}^{\infty}(u - e_{t,g}) \cdot Q_t(s,u).$$

The distribution of the number of deferred jobs, ϕ, can be calculated as follows. For $t = 1, \dots, T$:

$$\phi_t(j) = \begin{cases} \displaystyle\sum_{s=0}^{\infty}\sum_{u=0}^{e_{t,g}} Q_t(s,u), & j = 0 \\ \displaystyle\sum_{s=0}^{\infty} Q_t(s, e_{t,g} + j), & j > 0, \end{cases}$$

and

$$\phi = \phi_1 * \cdots * \phi_T,$$

where $*$ denotes the discrete convolution operator.

6. Balancing scheduled and unscheduled arrivals

In this section, we link the access and day process in order to maximize the number of unscheduled jobs served during the day of arrival, given the pre-specified access time service level norm for scheduled jobs. Since unscheduled jobs that cannot be served within g time slots receive an appointment, in order for a certain CAS to satisfy the access time service level norm, the deferred jobs ϕ^d resulting from that CAS should be accounted for in the appointment request arrival distribution γ^d. Therefore, we present an iterative procedure that uses Models I and II to find a candidate CAS in each iteration, which adapts the number of jobs to schedule by adding the deferred jobs from the previous iteration. The iterative procedure approximates the optimal value of F, the expected fraction of unscheduled jobs served on the day of arrival.

In the remainder of this section we first present the iterative procedure, followed by two different procedures for finding a candidate CAS within each iteration. The first procedure, complete enumeration, finds the optimal CAS within each iteration, but is computationally intensive. The second, heuristic, procedure, is not guaranteed to find the optimal CAS in each iteration, but is very fast and thus applicable to real-life instances. Table 4 summarizes the notation introduced in this section.

Table 4
Notation introduced in Section 6.

Symbol	Description
n	Iteration counter
$\phi^d(n)$	Distribution of the number of deferred jobs on day d in iteration n
$\nu^d(n)$	Expected number of deferred jobs on day d in iteration n
$\gamma^d(n)$	Total appointment request arrival distribution on day d in iteration n
ϵ	Precision of the iterative procedure's stop criterion
$K(n_f)$	Capacity cycle option f consisting of $(k^1(n_f), \dots, k^D(n_f))$ in iteration n
$C(n_f)$	The best CAS given capacity cycle $K(n_f)$
$\bar{\pi}_j^d(n_f)$	The probability that in iteration n under capacity cycle $K(n_f) j$ appointment reservations are utilized by appointments on day d
$\nu_C^*(n_f)$	$\mathbb{E}$[Total number of deferred jobs in iteration n under capacity cycle $K(n_f)$ and CAS C]
$\nu_{C^d\|j}^d(n_f)$	$\mathbb{E}$[Number of deferred jobs on day d in iteration n under capacity cycle $K(n_f)$ and CAS C when j appointment slots are utilized by scheduled jobs]
$k(n)$	Total number of appointment slots to allocate in iteration n when heuristically constructing a capacity cycle
$\psi^d(n)$	Estimated 'excess capacity' on day d in iteration n
b	Maximum number of appointment slots to swap between days within the capacity cycle in local search procedure
$\theta_t^d(n_f)$	Value indicating the attractiveness of planning appointments on day d in time slot t under capacity cycle $K(n_f)$
r	Number of neighboring day schedules evaluated in local search procedure

6.1. Iterative procedure

At the start of the iterative procedure, the expected number of deferred jobs is set to zero. Then, a candidate capacity cycle (using Model I) with accompanying appointment schedule (using Model II) is determined, given the appointment request arrival processes with rate λ^d and those of unscheduled job arrivals with rate χ_t^d. The distribution of the number of deferred jobs on day d in iteration n is denoted by $\phi^d(n)$, and the expected number by $\nu^d(n)$. If the expected number of jobs that has to be deferred under the resulting CAS is significantly larger than in the previous iteration, then apparently the reserved capacity for appointments was not sufficient. In this case, a new iteration starts.

In the subsequent iteration, to account for the jobs that were deferred, the distribution of appointment request arrivals $\gamma^d(n)$ is set to:

$$\gamma^d(n) = \text{Poisson}(\lambda^d) * \phi^d(n-1),$$

where Poisson(λ^d) denotes the Poisson distribution with parameter λ^d. As such, the appointment requests generated by deferred jobs are taken into account on the day of occurrence in the previous iteration. Then, a new candidate CAS is calculated. As more appointment slots are reserved, this may result in more deferred jobs than in the previous iteration. This iterative procedure is repeated until on each day in the cycle, a balance is found between the anticipated extra demand for appointments from deferred unscheduled jobs (which was $\nu^d(n-1)$) and the realized deferred unscheduled jobs (which is $\nu^d(n)$). The iterative procedure terminates if, for some small ϵ,

$$|\nu^d(n) - \nu^d(n-1)| < \epsilon, \quad d \in \{1, \dots, D\}.$$

It is important to note that we aim for balance on a day-by-day basis. Balance just on a cycle basis ($|\sum_{d=1}^{D} \nu^d(n) - \nu^d(n-1)| < \epsilon$) is not sufficient, since only in the case that $\left|\nu^d(n) - \nu^d(n-1)\right| < \epsilon,\ d \in \{1, \dots, D\}$, it is guaranteed that the appointment requests of deferred jobs occur in the way that was anticipated. Only then we can assure that in the access time calculations, we account for the deferred jobs on the day they occur since the access time calculations that use $\phi^d(n-1)$, based upon which the capacity cycle is designed, are still valid for $\phi^d(n)$ in this case. Fig. 3 displays the iterative procedure in pseudocode.

6.2. Complete enumeration

The first method to determine a candidate CAS within an iteration is to apply complete enumeration, which yields an optimal CAS within each iteration.

Generating capacity cycles. Using Model I, all capacity cycles fulfilling the specified access time service level norm are generated. Thus, given $\gamma^d(n)$, the set of capacity cycles $K = (k^1, \dots, k^D)$ that satisfy $(y, S^{\text{norm}}(y))$ is generated. Suppose that this set consist of m elements, then denote these elements for iteration n by $K(n_f) = (k^1(n_f), \dots, k^D(n_f)),\ f \in \{1, \dots, m\}$. From these elements, the best capacity cycle is selected, which is the capacity cycle that minimizes the expected number of deferred jobs. To do this, for each element $K(n_f)$, the best CAS $C(n_f)$ is determined.

Determining day schedules. The best CAS's are determined by applying Model II as follows. First, observe that although in a capacity cycle $K(n_f)$ there are $k^d(n_f)$ appointment slots reserved on day d, not all of these reserved slots are necessarily utilized by scheduled jobs. Since appointments are planned according to the FCFS principle, we know from the queue length probability vectors $\pi^d(n_f)$ of Model I, the probabilities of utilizing the first j out of the $k^d(n_f)$ reservations under capacity

Step 1: *specify input*	*Specify:* $R, T, D, g, q, S^{\text{norm}}(y), \epsilon$; $\forall d: \lambda^d; \forall d, t: \chi_t^d$.
Step 2: *initialize iterative procedure*	$n := 1; \forall d: \nu^d(1) := 0, \gamma^d(1) := \text{Poisson}(\lambda^d)$.
Step 3: *find candidate CAS*	*Execute complete enumeration (see Section 6.2)* *or heuristic procedure (see Section 6.3).*
Step 4: *assess current solution*	*If* $\forall d: \lvert\nu^d(n) - \nu^d(n-1)\rvert < \epsilon$, *then stop,* *else proceed to Step 5.*
Step 5: *adjust deferrals*	$\forall d: \nu^d(n+1) := \nu^d(n), \phi^d(n+1) := \phi^d(n)$, $\gamma^d(n+1) := \text{Poisson}(\lambda^d) * \phi^d(n+1)$; $n := n+1$ *and return to Step 3.*

Fig. 3. The iterative procedure.

cycle $K(n_f)$. Let us denote these probabilities by $\bar{\pi}_j^d(n_f)$:

$$\bar{\pi}_j^d(n_f) = \begin{cases} \pi_j^d(n_f), & j \in \{0, \ldots, k^d(n_f) - 1\} \\ \displaystyle\sum_{r=k^d(n_f)}^{\infty} \pi_r^d(n_f), & j = k^d(n_f). \end{cases}$$

By evaluating each day appointment schedule for $d \in \{1, \ldots, D\}$, $f \in \{1, \ldots, m\}$ and $j \in \{0, \ldots, k^d(n_f)\}$, the best CAS is determined for each capacity cycle $K(n_f)$ (i.e., by complete enumeration). Let $\nu_C(n_f)$ denote the expected total number of deferred jobs in cycle $K(n_f)$ under appointment schedule C, and let $\nu^*(n_f)$ denote the expected total number of deferred jobs in cycle $K(n_f)$ under the best appointment schedule. Then, for the best CAS, the best CASs are those that minimize:

$$\nu^*(n_f) = \min_C \nu_C(n_f) = \min_C \sum_{d=1}^{D} \sum_{j=0}^{k^d(n_f)} \bar{\pi}_j^d(n_f)\, \nu_{C^d|j}^d(n_f),$$

where $\nu_{C^d|j}^d(n_f)$ denotes the expected number of deferred jobs on day d under capacity cycle $K(n_f)$ and CAS C, if j appointment slots are utilized by scheduled jobs. Note that $C^d|j$ is a truncated version of C^d, in exactly the same way that $\widetilde{C}^d$ was defined in Section 5.

Selecting the best CAS. Now, the final step is to select the capacity cycle $K(n_f)$ and accompanying CAS, which is the CAS with the lowest expected number of deferred jobs, namely:

$$\nu^*(n) = \min_f \nu^*(n_f), \qquad f^*(n) = \arg\min_f \nu^*(n_f), \qquad C^*(n) = \arg\min_C \nu_C(n_{f*}).$$

6.3. *Heuristic procedure*

The heuristic procedure aims at finding a CAS quickly. In each iteration, the heuristic generates a limited number of capacity cycles fulfilling the specified access time service level norm (using Model I), and for each capacity cycle constructs an appointment schedule (using Model II).

Generating capacity cycles. The first step is to determine k, the total number of appointment slots to distribute over the days in the cycle. It is set as small as possible in order to minimize the number of deferred jobs, but larger than the expected demand for appointment slots:

$$k(n) := \left\lceil \sum_{d=1}^{D} \gamma^d(n) \right\rceil.$$

Second, a constructive heuristic generates a capacity cycle by distributing these $k(n)$ appointment slots over the days in the cycle, while aiming at minimizing the number of deferred jobs. Let ψ^d be the estimated 'excess capacity' on day d, i.e., capacity neither reserved for scheduled jobs nor, in expectation, needed to serve unscheduled jobs:

$$\psi^d(n) = R \cdot T - k^d(n) - \sum_{t=1}^{T} \chi_t^d.$$

The constructive heuristic starts with $k^d(n) = 0$ for $d \in \{1, \ldots, D\}$, and consecutively assigns an appointment slot to the day $\hat{d} := \arg\max_d \psi^d(n)$, until all appointment slots $k(n)$ have been assigned.

Third, based on the cycle just generated, a local search procedure increases the number of capacity cycles by, for all possible combinations of a day d_1 and another day d_2 in the cycle, constructing all capacity cycles in which $\hat{b}$ appointment slots from day d_1 are reassigned to day d_2 (where $\hat{b} \in \{1, \ldots, b\}$, with b a parameter, $b \geq 1$). This local search procedure thus generates at most $b \cdot D \cdot (D-1)$ additional capacity cycles. Finally, all generated capacity cycles are evaluated using Model I, and the m capacity cycles satisfying the access time service level norm are taken along to the second phase of the heuristic procedure. Note that it could happen that $m = 0$, in which case we set $k(n) := k(n)+1$ and repeat the constructive procedure.

Determining day schedules. In the second phase, for each capacity cycle $K(n_f)$, a constructive heuristic generates an initial day schedule whereupon a local search procedure improves it. For each day d in the cycle, the constructive heuristic aims at minimizing the number of deferred jobs. Let θ_t^d be the estimated 'excess capacity' in time slot t on day d, i.e., capacity neither reserved for scheduled jobs nor needed to serve unscheduled jobs in time slots $t-g+1$ to t:

$$\theta_t^d(n_f) = \sum_{\hat{t}=\max\{t-g+1,1\}}^{t} R - c_{\hat{t}}^d(n_f) - \chi_{\hat{t}}^d.$$

The constructive heuristic starts with $c_t^d(n_f) = 0$, $t \in \{1, \ldots, T\}$, and consecutively assigns an appointment slot to the time slot $\hat{t} := \arg\max_t \theta_t^d(n_f)$, until all appointment slots $k^d(n_f)$ have been assigned. If the number of appointment slots to allocate on day d is the same as in the previous iteration, i.e., $k^d(n_f) = k^d((n-1)_{f^*})$, we set $C^d(n_f) := C^d((n-1)_{f^*})$ and do not execute this constructive heuristic. Analogous to Section 6.2, we evaluate the resulting schedule using the probabilities $\bar{\pi}_j^d(n_f)$. Next, we generate a neighboring schedule $C^d(n_{f'})$ by randomly selecting a slot with and a slot without a reservation, and interchanging these. If $\nu^d(n_{f'}) < \nu^d(n_f)$, we set $C^d(n_{f'})$ as our new schedule and proceed generating a new neighbor from there; otherwise we generate a new neighbor from $C^d(n_f)$. This random search procedure terminates when r neighbor schedules have been evaluated. Note that our random search procedure is similar to the neighborhood search heuristic in [26]. Like [26], we also experimented with several local search variants and concluded that random search is best-performing with respect to the combination of solution quality and computation time.

Selecting the best CAS. Now, each capacity cycle $K(n_f)$ has an accompanying CAS, and the final step is to select the CAS with the lowest expected number of deferred jobs $\nu^*(n)$.

Remark 1 (*Convergence*)**.** The system is stable when $\left(\sum_{d=1}^{D} \lambda^d + \sum_{d=1}^{D} \sum_{t=1}^{T} \chi_t^d\right) < R \cdot T$, so that total demand does not exceed capacity. In addition, we would like to determine the conditions under which the iterative procedure converges. Therefore, first observe that since the unscheduled job arrival rate χ_t^d is fixed and the first iteration starts with no deferred jobs, i.e., $\nu^d(0) = 0$, in each iteration it is not possible to choose a CAS for which $\sum_{d=1}^{D} \nu^d(n) < \sum_{d=1}^{D} \nu^d(n-1)$. The total expected number of deferred jobs $\sum_{d=1}^{D} \nu^d(n)$ is thus monotonically non-decreasing. Also, if the access time norm $S^{\text{norm}}(y)$ is set such that it can be satisfied if all jobs are planned, we ensure that in each iteration it is possible to find feasible capacity cycles, i.e., capacity cycles for which $S(y) \geq S^{\text{norm}}(y)$. However, convergence of the iterative procedure is not assured. Although not likely for practical instances, it theoretically cannot be guaranteed that the iterative procedure does not keep jumping between points for which the total expected number of deferred jobs does not change, but without day-by-day balance, i.e., $\left|\sum_{d=1}^{D} \nu^d(n) - \nu^d(n-1)\right| < \epsilon$, and not $|\nu^d(n) - \nu^d(n-1)| < \epsilon$, $\forall d$. If such a case occurs, an additional rule to act as a tie-breaker is required. We extensively tested the iterative procedure by evaluating numerous different instances (see Section 7). Convergence was obtained for all instances.

7. Numerical experiments

This section presents the experimental results. All methods were coded with the CodeGear Delphi programming language and tested on an Intel 2.5 GHz PC with 4 GB of RAM. We test our methodology on a variety of 36 test instances, each with different characteristics, and perform a case study. Section 7.1 describes the input for both the test instances and the case study. The test instances (Section 7.2) provide insight in the execution of our method, and demonstrate the performance of the iterative procedure both with Complete Enumeration (in this section referred to by CE) and with the Heuristic Procedure (HP). We present the numerical results for the case study in Section 7.3, where we exhibit the practical potential of our methodology by presenting an appointment schedule for the mixed system of walk-in and appointment patients at the CT-scan facility of the AMC.

7.1. Input parameters

This section describes the input parameters for the 36 test instances and for the case study.

Test instances. We consider a facility with one resource, which operates in a cycle of length $D = 5$ days, where each day consists of $T = 8$ slots. We vary over three different arrival patterns for scheduled and unscheduled jobs. The initial demand per day for appointment requests is given by $(\lambda^1, \ldots, \lambda^5) = (5, 0, 2, 0, 7)$ for *Pattern* 1, $(2, 3, 4, 3, 2)$ for *Pattern* 2, and

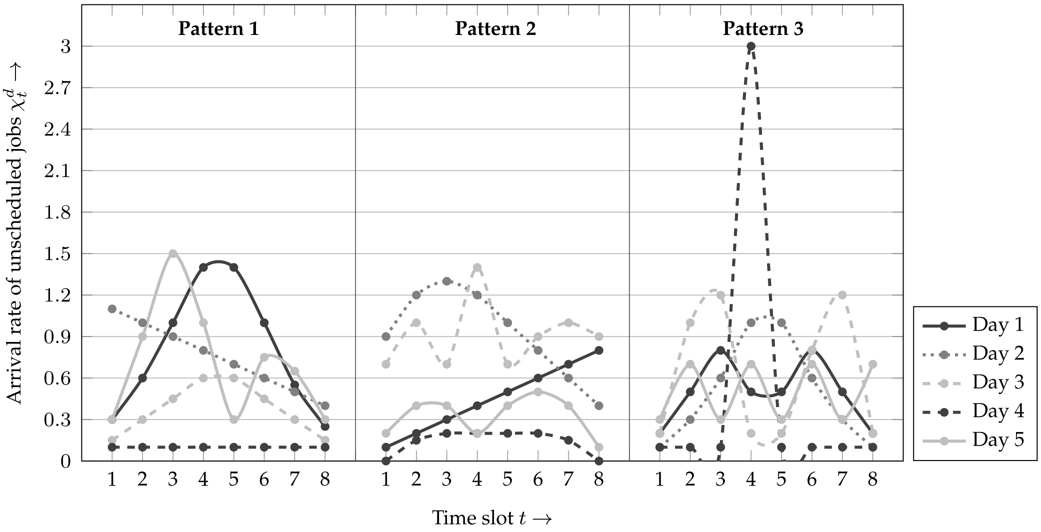


Fig. 4. Unscheduled job arrival rates per slot per day for the test instances.

(3, 3, 3, 3, 3) for *Pattern* 3. The arrival rates of unscheduled jobs χ_t^d in each pattern are displayed in Fig. 4, and are chosen such that different days in the cycle represent different unscheduled arrival patterns. Note that the total expected demand for scheduled jobs per cycle is 14, 14, and 15 for *Patterns* 1, 2, and 3 resp., and the total expected demand for unscheduled jobs per cycle is 22, 22, and 20.7 resp. Since there are $D \cdot T = 40$ time slots available within a cycle, the load of the system is 90%, 90%, and 89.25% resp. The access time service level norm is also varied; it is set such that 95% of the scheduled jobs are served within one, two, or three cycles, i.e., $(y, S^{\text{norm}}(y)) \in \{(5, 0.95), (10, 0.95), (15, 0.95)\}$. Furthermore, unscheduled jobs are willing to wait for a maximum of two or four time slots, i.e., $g \in \{2, 4\}$. We also vary the no-show probability of scheduled jobs: all scheduled jobs show up, or 15% does not show up, i.e., $q \in \{0, 0.15\}$. The stop criterion of the iterative method applies the threshold $\epsilon = 0.0001$. For HP, the maximum number of appointment slots to swap is set to $b = 2$ and the number of neighboring day schedules generated is set to $r = 10$. Table 5 provides an overview of the input parameters. By taking all possible combinations over three different arrival patterns and service level norms, and two values for both the unscheduled job patience and the no-show probability, we obtain 36 test instances.

Case study. The AMC has two CT-scanners for elective patients, i.e., $R = 2$, both available from 8:00 to 16:30 on each weekday, with time divided in 15-min slots, so $T = 34$ time slots per day. In the current situation all patients are served on appointment basis. Based on the expert opinions of the health care professionals who studied all scanning protocols of the various patient types, 72% of patients are eligible to be served on walk-in basis. To estimate the appointment request and walk-in arrival rates, one year of data of the CT-scan facility was combined with information on appointment schedules and referral rates from all outpatient clinics. Both arrival processes followed a weekly cycle, i.e., $D = 5$. The initial demand per day for appointment requests is given by $(\lambda^1, \ldots, \lambda^5) = (12.0, 11.9, 11.6, 13.5, 10.3)$; Fig. 5 displays the estimated walk-in arrival rates. These arrival rates result in a load of 62.3%, equivalent to the utilization rate in the data. In line with AMC policy, the access time service level norm is set such that 95% of the patients who are eventually scheduled are served within 10 days, i.e., $(y, S^{\text{norm}}(y)) = (10, 0.95)$. A patient survey revealed that walk-in patients are willing to wait for a maximum of

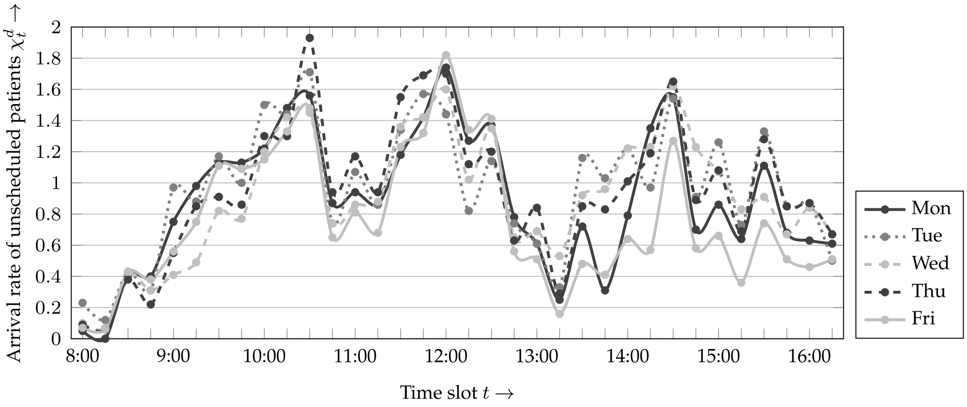


Fig. 5. Unscheduled patient arrival rates per slot per day for the case study.

Table 5
Input parameter settings of the test instances.

Parameter	Description	Value
Fixed		
R	Number of resources	1
D	Cycle length	5
T	Number of time slots	8
ϵ	Iterative method's precision	0.0001
b	Maximum number of slots to swap	2
r	Stop criterion random search	10
To be varied		
λ^d, χ_t^d	Arrival patterns	{*Pattern* 1, *Pattern* 2, *Pattern* 3}
$(y, S^{\text{norm}}(y))$	Service level norm	$\{(5, 0.95), (10, 0.95), (15, 0.95)\}$
g	Unscheduled job patience	$\{2, 4\}$
q	No-show probability	$\{0, 0.15\}$

30 min, i.e., $g = 2$. The no-show rate in the data is 5%, i.e., $q = 0.05$. As before, the stop criterion of the iterative procedure applies the threshold $\epsilon = 0.0001$. For HP, the maximum number of appointment slots to swap is set to $b = 2$. With the number of neighboring day schedules generated set to $r = 10$ as for the test instances, the iterative procedure does not converge within 48 h. When r is small compared to the size of the instance, subsequent iterations may yield considerably different day schedules, impeding convergence. We gradually increased the value of r until we reached $r = 75$, for which the iterative procedure does again converge within reasonable time. Table 6 provides an overview of the input parameters.

7.2. Performance of the iterative procedure

In this section, we first illustrate the execution of the iterative procedure by discussing one of the test instances in detail. This test instance, *Instance* 13, has arrival *Pattern* 1, service level norm $(y, S^{\text{norm}}(y)) = (10, 0.95)$, unscheduled job patience $g = 2$, and no-show probability $q = 0$. After illustrating the execution of the iterative procedure based on this instance, we discuss its results in detail. Finally, we present the overall results on all 36 test instances, and compare the performance of CE and HP.

Execution of the iterative procedure. Since the evolution of CE and HP is similar, with minor differences only in the path followed, we illustrate the execution using CE. CE was executed for *Instance* 13 and the results obtained from each iteration are displayed in Table 7. In the first iteration the number of deferred unscheduled jobs is positive on each day of the cycle, $\nu^d(1) > 0$, $d \in \{1, \dots, D\}$. The total number of deferred jobs is $\sum_{d=1}^{D} \nu^d(1) = 4.055$. Therefore, the deferred jobs are added to the scheduled arrival stream and a new iteration is started. This procedure repeats until in iteration 14 balance is obtained for each day, i.e., $|\nu^d(n) - \nu^d(n-1)| < \epsilon$, $d \in \{1, \dots, D\}$. From Table 7 it can be seen that (as described in Remark 1, Section 6) the total number of deferred jobs is monotonically non-decreasing, while deferrals on the day level are both increasing and decreasing. The fluctuations are substantial in the first iterations and the system stabilizes after six iterations.

This behavior is also reflected by the dynamics of the capacity cycles found. The total number of slots reserved for appointments develops as follows: $(16, 19, 21, 21, 21, 22, \dots, 22)$. Again, although the total number of reserved slots $\sum_d k^d$ is monotonically non-decreasing, for a specific day k^d may decrease. For example, the capacity cycles of iterations 3–5 all have a total capacity of 21, but the one obtained in the third iteration is changed in iteration 4 so that one appointment is shifted from day 5 to day 3. This change is reversed in iteration 5. The final capacity cycle is obtained in iteration 6. The only purpose of iterations 7–14 is to obtain the desired balance in the daily deferrals. Note that this is a direct result of the magnitude of ϵ. If ϵ were >0.0001, the iterative procedure would have stopped earlier.

Results Instance 13. Table 8 presents the final results for *Instance* 13, for CE and HP. When both procedures have the same result for a given indicator, the table only presents the result once. The percentage of unscheduled jobs served on the day of

Table 6
Input parameter settings of the case study.

Parameter	Description	Value
R	Number of available resources	2
D	Cycle length	5
T	Number of time slots	34
ϵ	Iterative procedure's precision	0.0001
b	Maximum number of slots to swap	2
r	Stop criterion random search	75
$(\lambda^1, \dots, \lambda^5)$	Appointment request arrival rates	(12.0, 11.9, 11.6, 13.5, 10.3)
$(y, S^{\text{norm}}(y))$	Service level norm	(10, 95%)
g	Unscheduled job patience	2 (i.e., 30 min)
q	No-show probability	0.05

Table 7
Results per iteration step of the iterative procedure using CE.

n	Day	Tot. app. req. rate	Deferral rate		Difference	Capacity cycle	CAS
	d	γ^d	$\nu^d(n-1)$	$\nu^d(n)$	$\lvert\nu^d(n-1)-\nu^d(n-1)\rvert$	k^d	C^d
1	1	5	0	1.133	1.133	1	(1, 0, 0, 0, 0, 0, 0, 0)
	2	0	0	0.865	0.865	1	(1, 0, 0, 0, 0, 0, 0, 0)
	3	2	0	0.547	0.547	4	(1, 1, 0, 1, 0, 0, 1, 0)
	4	0	0	0.637	0.637	8	(1, 1, 1, 1, 1, 1, 1, 1)
	5	7	0	0.873	0.873	2	(1, 1, 0, 0, 0, 0, 0, 0)
2	1	6.133	1.133	1.456	0.323	2	(1, 1, 0, 0, 0, 0, 0, 0)
	2	0.865	0.865	1.296	0.431	2	(1, 0, 0, 0, 0, 0, 1, 0)
	3	2.547	0.547	0.549	0.002	4	(1, 1, 0, 1, 0, 0, 1, 0)
	4	0.637	0.637	0.736	0.099	8	(1, 1, 1, 1, 1, 1, 1, 1)
	5	7.873	0.873	1.371	0.498	3	(1, 1, 0, 0, 0, 0, 1, 0)
3	1	6.456	1.456	1.456	0.000	2	(1, 1, 0, 0, 0, 0, 0, 0)
	2	1.296	1.296	1.296	0.000	2	(1, 0, 0, 0, 0, 0, 1, 0)
	3	2.549	0.549	0.952	0.403	5	(1, 1, 1, 0, 0, 1, 0, 1)
	4	0.736	0.736	0.715	0.021	8	(1, 1, 1, 1, 1, 1, 1, 1)
	5	8.371	1.371	1.752	0.381	4	(1, 1, 0, 0, 1, 1, 0)
4	1	6.456	1.456	1.456	0.000	2	(1, 1, 0, 0, 0, 0, 0, 0)
	2	1.296	1.296	1.296	0.000	2	(1, 0, 0, 0, 0, 0, 1, 0)
	2	1.296	1.296	1.296	0.000	2	(1, 0, 0, 0, 0, 0, 1, 0)
	3	2.952	0.952	1.498	0.546	6	(1, 1, 1, 0, 1, 0, 1, 1)
	4	0.715	0.715	0.742	0.027	8	(1, 1, 1, 1, 1, 1, 1, 1)
	5	8.752	1.752	1.402	0.350	3	(1, 1, 0, 0, 0, 0, 1, 0)
5	1	6.456	1.456	1.456	0.000	2	(1, 1, 0, 0, 0, 0, 0, 0)
	2	1.296	1.296	1.296	0.000	2	(1, 0, 0, 0, 0, 0, 1, 0)
	3	3.498	1.498	0.954	0.544	5	(1, 1, 1, 0, 0, 1, 0, 1)
	4	0.742	0.742	0.771	0.029	8	(1, 1, 1, 1, 1, 1, 1, 1)
	5	8.402	1.402	2.049	0.647	4	(1, 1, 0, 0, 1, 0, 1, 0)
6	1	6.456	1.456	1.456	0.000	2	(1, 1, 0, 0, 0, 0, 0, 0)
	2	1.296	1.296	1.296	0.000	2	(1, 0, 0, 0, 0, 0, 1, 0)
	3	2.954	0.954	1.495	0.541	6	(1, 1, 1, 0, 1, 0, 1, 1)
	4	0.771	0.771	0.721	0.050	8	(1, 1, 1, 1, 1, 1, 1, 1)
	5	9.049	2.049	1.794	0.255	4	(1, 1, 0, 0, 0, 1, 1, 0)
		⋮				⋮	
14	1	6.456	1.456	1.456	0.000	2	(1, 1, 0, 0, 0, 0, 0, 0)
	2	1.296	1.296	1.296	0.000	2	(1, 0, 0, 0, 0, 0, 1, 0)
	3	3.497	1.497	1.497	0.000	6	(1, 1, 1, 0, 1, 0, 1, 1)
	4	0.743	0.743	0.743	0.000	8	(1, 1, 1, 1, 1, 1, 1, 1)
	5	8.897	1.897	1.897	0.000	4	(1, 1, 0, 0, 1, 1, 0)

arrival is 69%, so $F = 0.69$. This fraction is composed by fractions $F^1, \dots, F^D$ that differ from day to day ($F^d = (\sum_t \chi_t^d - \nu^d)/\sum_t \chi_t^d$). For example, since day 4 is a quiet day with respect to unscheduled job arrivals, it is completely filled with appointments. Only if no appointment request is made in one of the reserved slots, an unscheduled job can be served. Apparently, it pays off to serve on average only 7% of the unscheduled jobs directly on day 4 in the cycle. This is a result of the fact that only 3.6% of the unscheduled jobs arrive on day 4, and that accordingly appointments are preferably planned on this day. The deferred unscheduled jobs stream per day and the expected number of unscheduled jobs served on the day of arrival are displayed in Table 8, which also reflects that on day 4 a small amount of unscheduled jobs is directly served but also relatively few jobs are deferred. The realized service level $S(10) = 0.99$ is well above the defined service level norm of 0.95.

The resulting capacity cycle is $K = (2, 2, 6, 8, 4)$, with corresponding day schedules as shown in Table 8. Note that to achieve the service level norm it is required to reserve a buffer capacity of 1.11 to account for variability in appointment request arrivals, since 22 appointment slots are reserved while the average total number of jobs to schedule within a cycle is $\sum_d(\lambda^d + \nu^d) = 14 + 6.89 = 20.89$. The service level norm is achieved with only 5% buffer capacity, thus in this instance reserved capacity for appointments can be used efficiently.

For day 3, HP finds another day schedule than CE. Because HP employs random search, it may generate different results when executed multiple times with the same input parameters. Therefore, we ran HP twenty times for each of the test instances, and kept track of the results for each run. For *Instance* 13, this procedure found the same capacity cycle as CE in all runs. In twelve of the twenty runs, however, a different day schedule was found for day 3. Table 8 contains the results for such a run. Although minor differences in performance measures were found due to the different CAS, these differences are so small that they do not show up in the other values in Table 8.

Table 8
End results for *Instance* 13.

Indicator	Description	Value
F	Fraction unscheduled directly served	0.69
$F^1, \ldots, F^5$	Daily fraction unscheduled directly served	0.78, 0.78, 0.50, 0.07, 0.67
$S(10)$	Service level scheduled jobs	0.99
$\nu^1, \ldots, \nu^5$	Deferral rate per day	1.46, 1.30, 1.50, 0.74, 1.90
$\sum_t \chi_t^1 - \nu^1, \ldots, \sum_t \chi_t^5 - \nu^5$	Unscheduled job service rate per day	5.04, 4.70, 1.50, 0.06, 3.80
$L^1, \ldots, L^5$	Realized utilization per day	0.88, 0.84, 0.94, 0.96, 0.88
K	Capacity cycle	(2, 2, 6, 8, 4)
C^1	CAS day 1	(1, 1, 0, 0, 0, 0, 0, 0)
C^2	CAS day 2	(1, 0, 0, 0, 0, 0, 1, 0)
C^3	CAS day 3	(1, 1, 1, 0, 1, 0, 1, 1) (CE) (1, 1, 1, 1, 0, 1, 0, 1) (HP)
C^4	CAS day 4	(1, 1, 1, 1, 1, 1, 1, 1)
C^5	CAS day 5	(1, 1, 0, 0, 0, 1, 1, 0)

The realized expected load per day, denoted by $L^1, \ldots, L^D$, is a result of the capacity cycle, the probabilities that the reserved appointment slots are utilized by appointment requests and the expected number of unscheduled jobs served on day of arrival $\left(\sum_t \chi_t^d - \nu^d\right)$. Each day's load lies between 84 and 96%.

Results all test instances. Table 9 shows the overall results for all 36 test instances. For HP, we present the minimum ($F_{\min}$), average (F_{av}), and maximum ($F_{\max}$) fraction of unscheduled jobs directly served over twenty runs.

The fraction of unscheduled jobs served on the day of arrival, F, ranges from 63% to 83%. As expected, this fraction increases when unscheduled jobs are willing to wait longer, or when the no-show probability of scheduled jobs is higher. Also, when y is set less tight, F increases, since there is more flexibility to spread the appointments. However, Table 9 indicates that this effect is bounded. This is most apparent for arrival *Pattern* 3; F increases when going from $y = 5$ to $y = 10$, but F remains stable when going from $y = 10$ to $y = 15$. This is due to the fact that, regardless of the access time service level norm, a certain minimum number of appointments is required for the system to be stable.

The conclusion is that the resulting CAS and its performance is the outcome of the complex interaction between the scheduled job arrival rates λ^d, the unscheduled jobs arrival patterns χ_t^d, and the service level requirement $S^{\text{norm}}(y)$.

Comparison complete enumeration and heuristic procedure. Table 9 compares the performance of CE and HP. CE outperforms HP in 35 out of 36 instances when considering F, the fraction of unscheduled jobs served on the day of arrival. For *Instance* 3, HP outperforms CE in two runs, while five runs find the same solution, and the remaining 13 runs perform worse. The differences between the two procedures are small. In 13 of the 36 instances the maximum deviation between the two procedures is less than 0.01%, and the maximum deviation over all runs of all instances is 3.19%. The average deviation over all runs and instances is 0.19%. There is no consistency with regard to the influence of the different parameter settings on how well the two procedures perform compared to each other.

Regarding run time, HP outperforms CE. The average run time for HP is 0.69 min, while the average run time for CE is 481.55 min. This difference will further increase when increasing the problem size, due to the non-linear increase in the total number of possible CAS's.

We conclude that CE finds slightly better solutions than HP. However, the run time of CE makes this method not applicable to analyze large problem instances. Hence, HP is applied to analyze the case study in Section 7.3.

7.3. Case study results

In this section, we present the results for the case study introduced in Section 7.1. We first apply HP to the data obtained from the CT-scan facility in the AMC with a system load of 62.3%, the so-called base case, and discuss the results. Subsequently we evaluate a scenario with increased load, to investigate the performance of our approach under higher loads and the facility's growth potential when implementing a mixed system.

Results base case. Table 10 presents the results for the base case. The percentage of unscheduled jobs served on the day of arrival is 99%. This fraction is composed by fractions $F^1, \ldots, F^D$ that are similar on each weekday, such that an unscheduled patient arriving at the CT-scan facility on a particular weekday has a similar probability of being served directly each day. This is an advantage for patient equity, and also simplifies communication about the mixed system to patients and referring physicians. Again, the realized service level for scheduled jobs, $S(10) = 1.00$, is well above the defined service level norm of 0.95. The resulting capacity cycle is $K = (14, 10, 12, 10, 15)$, with corresponding day schedules which we discuss one-by-one below.

It turns out that the realized expected load per day, denoted by $L^1, \ldots, L^D$, is balanced throughout the cycle where each day has a realized load between 60% and 64%. This is an advantage in terms of work load for laboratory workers, and therefore increases the acceptability of the new mixed system for them. Moreover, although a balanced load tends to be viewed as one of the advantages of an appointment system, and one of the things being at risk in a mixed system, these results indicate that a balanced load can also be achieved in a mixed system.

Table 9
Results for all instances.

Instance	Input settings				Results CE		Results HP				Max. deviation
	$(y, S^{\text{norm}}(y))$	g	q	Arrival pattern	F	Time (min)	$F_{\min}$	F_{av}	$F_{\max}$	Time (min)	$\lvert(F_{\min} - F)/F\rvert$ (%)
1	(05, 95)	2	0.00	1	0.66	938.08	0.65	0.65	0.66	1.30	0.44
2	(05, 95)	2	0.00	2	0.65	467.45	0.63	0.65	0.65	0.57	3.19
3	(05, 95)	2	0.00	3	0.64	337.29	0.64	0.64	0.64	1.14	0.20
4	(05, 95)	2	0.15	1	0.70	911.00	0.70	0.70	0.70	0.97	0.33
5	(05, 95)	2	0.15	2	0.69	503.50	0.68	0.68	0.68	0.67	2.22
6	(05, 95)	2	0.15	3	0.70	272.99	0.70	0.70	0.70	0.99	0.03
7	(05, 95)	4	0.00	1	0.75	967.98	0.75	0.75	0.75	0.93	0.05
8	(05, 95)	4	0.00	2	0.72	781.58	0.72	0.72	0.72	0.69	0.00
9	(05, 95)	4	0.00	3	0.76	303.96	0.75	0.76	0.76	0.97	1.70
10	(05, 95)	4	0.15	1	0.79	612.31	0.79	0.79	0.79	0.76	0.00
11	(05, 95)	4	0.15	2	0.76	434.35	0.76	0.76	0.76	0.49	0.00
12	(05, 95)	4	0.15	3	0.81	287.67	0.81	0.81	0.81	0.83	0.23
13	(10, 95)	2	0.00	1	0.69	760.00	0.69	0.69	0.69	0.97	0.01
14	(10, 95)	2	0.00	2	0.69	467.00	0.69	0.69	0.69	0.46	0.00
15	(10, 95)	2	0.00	3	0.69	287.95	0.68	0.69	0.69	0.89	0.70
16	(10, 95)	2	0.15	1	0.73	537.00	0.73	0.73	0.73	0.76	0.25
17	(10, 95)	2	0.15	2	0.71	474.00	0.71	0.71	0.71	0.40	0.00
18	(10, 95)	2	0.15	3	0.73	264.49	0.73	0.73	0.73	0.72	0.31
19	(10, 95)	4	0.00	1	0.78	802.66	0.78	0.78	0.78	0.66	0.00
20	(10, 95)	4	0.00	2	0.76	372.57	0.75	0.76	0.76	0.48	0.33
21	(10, 95)	4	0.00	3	0.79	292.91	0.79	0.79	0.79	0.80	0.00
22	(10, 95)	4	0.15	1	0.80	527.00	0.80	0.80	0.80	0.60	0.00
23	(10, 95)	4	0.15	2	0.79	384.00	0.78	0.79	0.79	0.35	1.23
24	(10, 95)	4	0.15	3	0.83	219.61	0.83	0.83	0.83	0.57	0.03
25	(15, 95)	2	0.00	1	0.71	722.00	0.71	0.71	0.71	0.72	0.01
26	(15, 95)	2	0.00	2	0.69	456.74	0.69	0.69	0.69	0.41	0.00
27	(15, 95)	2	0.00	3	0.69	321.62	0.68	0.69	0.69	0.87	0.74
28	(15, 95)	2	0.15	1	0.74	530.00	0.74	0.74	0.74	0.56	0.00
29	(15, 95)	2	0.15	2	0.71	472.00	0.71	0.71	0.71	0.40	0.00
30	(15, 95)	2	0.15	3	0.73	291.35	0.73	0.73	0.73	0.78	0.30
31	(15, 95)	4	0.00	1	0.79	525.46	0.78	0.78	0.78	0.62	0.55
32	(15, 95)	4	0.00	2	0.76	368.98	0.76	0.76	0.76	0.42	0.10
33	(15, 95)	4	0.00	3	0.79	323.77	0.79	0.79	0.79	0.80	0.00
34	(15, 95)	4	0.15	1	0.82	524.46	0.81	0.81	0.82	0.49	1.25
35	(15, 95)	4	0.15	2	0.79	358.79	0.79	0.79	0.79	0.35	0.00
36	(15, 95)	4	0.15	3	0.83	233.10	0.83	0.83	0.83	0.58	0.06

Finally, we discuss the resulting day schedules, to explain the moments on which the appointments are planned (see also Fig. 6).

Monday. Only in the first two time slots, both CT-scanners are reserved for appointments. This is due to the very low unscheduled arrival rates in these time slots, but also to the waiting behavior of unscheduled patients. Since unscheduled patients are willing to wait for two time slots, a peak in arrivals has an impact until two slots afterwards. If appointments were planned at the end of the day, there is no possibility to serve arriving unscheduled patients, while when planning appointments at slots at the beginning of the day, early unscheduled arrivals can be served in the third time slot.

Tuesday. The appointments are planned around the unscheduled arrival peaks. It is remarkable that the last appointment does not occur exactly during the off-peak hours but later, which can also be explained by the aforementioned delayed impact of unscheduled arrival peaks.

Wednesday. Again, the tendency to plan appointments early on the day is evident. Although the latest time slot has an unscheduled arrival rate that is lower than some of the time slots reserved for an appointment, the latest time slot is not reserved for an appointment, to be able to serve unscheduled patients arriving during interval (15:45, 16:15].

Thursday. As described before, only in the first two time slots, both CT-scanners are reserved for appointments. During the rest of the day always one scanner is kept free, so to spread the possibilities for unscheduled job service. This spreading is further promoted by the fact that the two later appointment slots are not planned consecutively.

Friday. The demand for unscheduled jobs is relatively low. Therefore, more slots are reserved for scheduled jobs on Friday, compared to other days. Again, the appointments are planned around the unscheduled arrival peaks, and the tendency to plan appointments with a two time slot delay after an unscheduled arrival peak can be clearly witnessed from the appointments planned at 13:15 and 15:15.

The results presented here were found within a run time of 38.5 min, which seems to be a very reasonable amount of time to generate an appointment schedule that will be updated at most a few times per year.

Results increased load. We constructed the scenario with higher load by scaling the original appointment request and walk-in arrival rates such that the resulting load is 85.0%. Table 11 displays the results for this scenario. These results were found in a

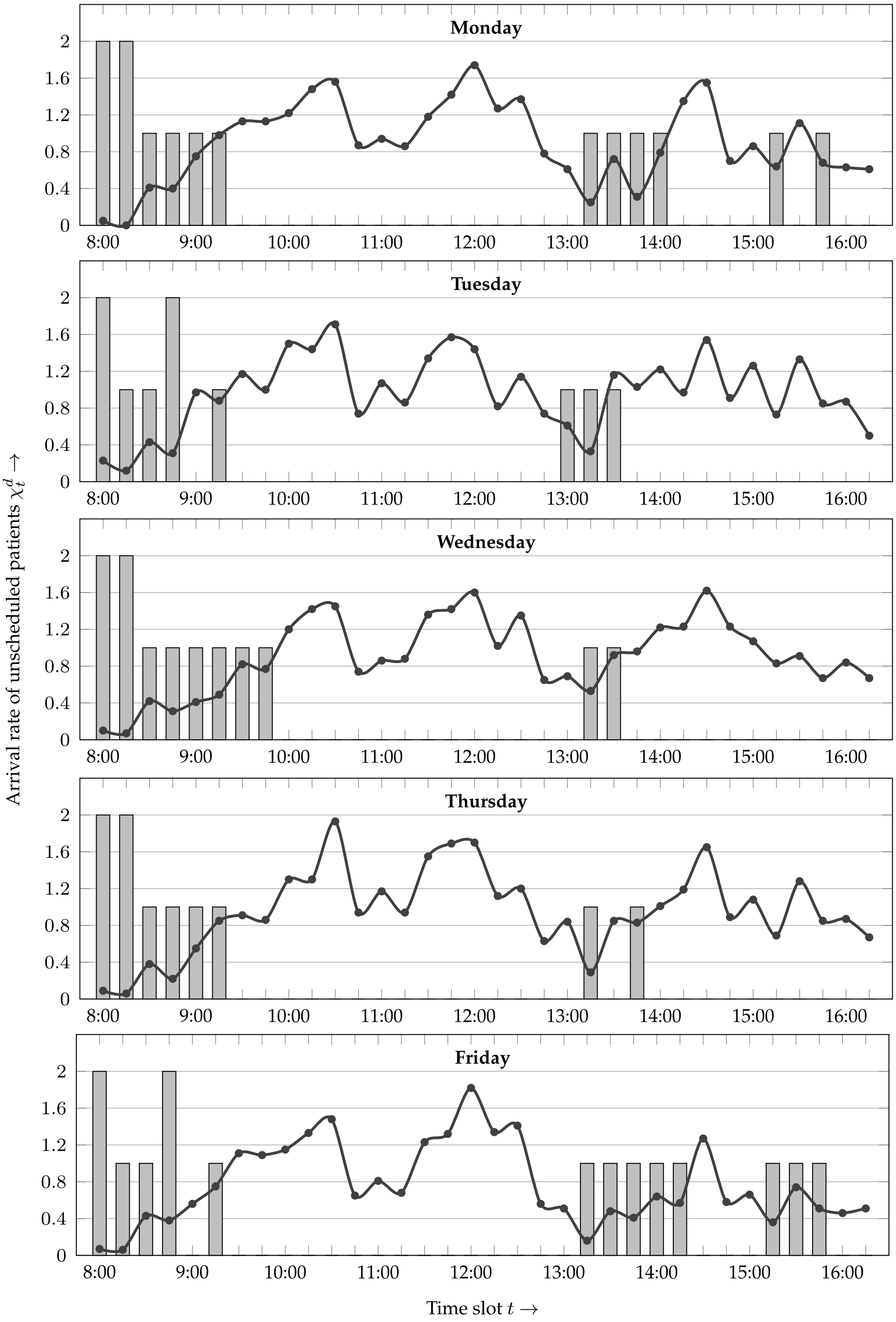


Fig. 6. The CAS versus the unscheduled patient arrival rates.

run time of 161.5 min, which indicates that the run time remains acceptable under higher loads. Similar to the base case, we find that appointments are planned in the beginning of the day, and during times when the walk-in rates are relatively low. Furthermore, we see that also for this case with increased load, the procedure is able to find an appointment schedule that allows 94% of walk-in patients to be served within 30 min of their arrival. As before, the realized service level is well above the pre-specified norm, and both the daily fraction of unscheduled patients directly served and the realized utilizations per day are equally spread over the days.

Concluding, we see that the developed methodology finds a good appointment schedule in a reasonable amount of time for a real-life instance.

Table 10
End results for the case study (base case).

Indicator	Value
F	0.99
$F^1, \dots, F^5$	0.99, 0.99, 0.99, 0.99, 1.00
$S(10)$	1.00
$\nu^1, \dots, \nu^5$	0.20, 0.18, 0.17, 0.21, 0.11
$\sum_t \chi_t^1 - \nu^1, \dots, \sum_t \chi_t^5 - \nu^5$	30.16, 32.58, 30.58, 32.17, 25.99
$L^1, \dots, L^5$	0.64, 0.63, 0.62, 0.62, 0.60
K	(14, 10, 12, 10, 15)
C^1	(2, 2, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0)
C^2	(2, 1, 1, 2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
C^3	(2, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
C^4	(2, 2, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
C^5	(2, 1, 1, 2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0)

Table 11
End results for the case study (increased load).

Indicator	Value
F	0.94
$F^1, \dots, F^5$	0.94, 0.94, 0.94, 0.94, 0.94
$S(10)$	1.00
$\nu^1, \dots, \nu^5$	2.68, 2.64, 2.44, 2.62, 2.08
$\sum_t \chi_t^1 - \nu^1, \dots, \sum_t \chi_t^5 - \nu^5$	38.76, 42.09, 39.54, 41.59, 33.57
$L^1, \dots, L^5$	0.86, 0.85, 0.84, 0.85, 0.84
K	(20, 16, 18, 16, 24)
C^1	(2, 2, 1, 2, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0)
C^2	(2, 2, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0)
C^3	(2, 2, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0)
C^4	(2, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0)
C^5	(2, 2, 1, 2, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0)

8. Discussion and conclusion

In this article we have outlined a methodology to develop an appointment schedule for facilities with scheduled and unscheduled arrival streams. The methodology consists of two separate models, one to evaluate the access and the other to evaluate the day process. The two models are linked by an iterative algorithm. An advantage of this modular approach is that the models and the algorithm can be updated separately, so that a high level of flexibility is obtained.

In order to generate suitable appointment schedules, two methods, complete enumeration and a heuristic procedure, were employed and tested on a considerable number of test instances. Both methods perform well, with complete enumeration slightly outperforming the heuristic procedure on solution quality, while the latter is significantly faster. The heuristic procedure was then used to develop appointment schedules for the CT-facility at the AMC, thereby demonstrating its practical value. For the instances considered, our methodology balances both the workload and the percentage of unscheduled jobs served on the day of arrival throughout the cycle, while realizing service levels for scheduled jobs well above the defined service level norms. The fraction of unscheduled jobs served on the day of arrival is well above 90% for the considered practical instances.

Some extensions can readily be incorporated in our approach. For instance, the management of a facility may want to incorporate service levels on more than one of the percentiles of the access time distribution. Also, different choices for the time jobs are willing to wait (job patience) could be studied, just as overbooking to anticipate for no-shows. Furthermore, the access time for scheduled jobs and the fraction of unscheduled jobs that cannot be served on the day of arrival are outcomes of Model I and Model II respectively, and serve as input for the iterative procedure. Of course, other model outcomes could be chosen as well. Another direct extension would be to incorporate planned maintenance of a service facility: the number of available slots in the day process can easily be amended by closing slots.

The practical contribution of our methodology is that it supports the realization of one-stop shopping at outpatient care facilities. In many settings one-stop shopping is highly valuable to patients to offer the combination of consultations, diagnostics, and treatments during a single visit. By one-stop shopping the number of hospital visits can be reduced, and required treatments can earlier be commenced and better be coordinated.

To conclude, the case study for the AMC showed the advantages of offering combined walk-in and scheduled service at its CT-scan facility. The research project was performed in close cooperation with healthcare professionals of the Radiology department of the AMC. Based on our findings the AMC decided to start offering walk-in service by implementing a mixed appointment/walk-in system.

Acknowledgments

This research is supported by the Dutch Technology Foundation STW, applied science division of NWO (Grant No. 08140) and the Technology Program of the Ministry of Economic Affairs.

The authors are grateful to the health care managers and medical doctors of Academic Medical Center Amsterdam and Leiden University Medical Center, who inspired us to take up this research topic, the many fruitful discussions and their involvement in further development and implementation of the methodology.

References

[1] M. Murray, D.M. Berwick, Advanced access: reducing waiting and delays in primary care, J. Am. Med. Assoc. 289 (8) (2003) 1035–1040.
[2] R. Ashton, L. Hague, M. Brandreth, D. Worthington, S. Cropper, A simulation-based study of a NHS walk-in centre, J. Oper. Res. Soc. 56 (2) (2004) 153–161.
[3] J.K. Cochran, K.T. Roche, A multi-class queuing network analysis methodology for improving hospital emergency department performance, Comput. Oper. Res. 36 (5) (2009) 1497–1512.
[4] P. Williams, G. Tai, Y. Lei, Simulation based analysis of patient arrival to health care systems and evaluation of an operations improvement scheme, Ann. Oper. Res. 178 (1) (2010) 263–279.
[5] D. Gupta, B. Denton, Appointment scheduling in health care: challenges and opportunities, IIE Trans. 40 (9) (2008) 800–819.
[6] T. Cayirli, E. Veral, Outpatient scheduling in health care: a review of literature, Prod. Oper. Manage. 12 (4) (2003) 519–549.
[7] P.J.H. Hulshof, N. Kortbeek, R.J. Boucherie, E.W. Hans, P.J.M. Bakker, Taxonomic classification of planning decisions in health care: a structured review of the state of the art in OR/MS, Health Syst. 1 (2) (2012) 129–175.
[8] G.C. Kaandorp, G. Koole, Optimal outpatient appointment scheduling, Health Care Manage. Sci. 10 (3) (2007) 217–229.
[9] K.J. Klassen, R. Yoogalingam, Improving performance in outpatient appointment services with a simulation optimization approach, Prod. Oper. Manage. 18 (4) (2009) 447–458.
[10] C.J. Liao, C.D. Pegden, M. Rosenshine, Planning timely arrivals to a stochastic production or service system, IIE Trans. 25 (5) (1993) 63–73.
[11] L. Liu, X. Liu, Dynamic and static job allocation for multi-server systems, IIE Trans. 30 (9) (1998) 845–854.
[12] P.M.V. Vanden Bosch, D.C. Dietz, J.R. Simeoni, Scheduling customer arrivals to a stochastic service system, Nav. Res. Logist. 46 (5) (1999) 549–559.
[13] B. Lehaney, S.A. Clarke, R.J. Paul, A case of an intervention in an outpatients department, J. Oper. Res. Soc. 50 (9) (1999) 877–891.
[14] C.J. Ho, H.S. Lau, Minimizing total cost in scheduling outpatient appointments, Manag. Sci. 38 (12) (1992) 1750–1764.
[15] S. Creemers, J. Beliën, M. Lambrecht, The optimal allocation of server time slots over different classes of patients, European J. Oper. Res. 219 (3) (2012) 508–521.
[16] R. Hassin, S. Mendel, Scheduling arrivals to queues: a single-server model with no-shows, Manag. Sci. 54 (3) (2008) 565–572.
[17] C.D. Pegden, M. Rosenshine, Scheduling arrivals to queues, Comput. Oper. Res. 17 (4) (1990) 343–348.
[18] S. Creemers, Appointment-driven queueing systems (Ph.D. thesis), Katholieke Universiteit Leuven, 2009.
[19] T. Cayirli, E. Veral, H. Rosen, Assessment of patient classification in appointment system design, Prod. Oper. Manage. 17 (3) (2008) 338–353.
[20] K.J. Klassen, T.R. Rohleder, Scheduling outpatient appointments in a dynamic environment, J. Oper. Manage. 14 (2) (1996) 83–101.
[21] P.M.V. Vanden Bosch, D.C. Dietz, Minimizing expected waiting in a medical appointment system, IIE Trans. 32 (9) (2000) 841–848.
[22] P.P. Wang, Sequencing and scheduling N customers for a stochastic server, European J. Oper. Res. 119 (3) (1999) 729–738.
[23] J. Patrick, M.L. Puterman, Improving resource utilization for diagnostic services through flexible inpatient scheduling: a method for improving resource utilization, J. Oper. Res. Soc. 58 (2) (2007) 235–245.
[24] L.V. Green, S. Savin, B. Wang, Managing patient service in a diagnostic medical facility, Oper. Res. 54 (1) (2006) 11–25.
[25] R. Kolisch, S. Sickinger, Providing radiology health care services to stochastic demand of different customer classes, OR Spectrum 30 (2) (2008) 375–395.
[26] S. Sickinger, R. Kolisch, The performance of a generalized Bailey–Welch rule for outpatient appointment scheduling under inpatient and emergency demand, Health Care Manage. Sci. 12 (4) (2009) 408–419.
[27] T. Cayirli, E. Veral, H. Rosen, Designing appointment scheduling systems for ambulatory care services, Health Care Manage. Sci. 9 (1) (2006) 47–58.
[28] L.R. LaGanga, S.R. Lawrence, Clinic overbooking to improve patient access and increase provider productivity*, Decis. Sci. 38 (2) (2007) 251–276.
[29] T.A. Reilly, V.P. Marathe, B.E. Fries, A delay-scheduling model for patients using a walk-in clinic, J. Med. Syst. 2 (4) (1978) 303–313.
[30] S. Su, C.L. Shih, Managing a mixed-registration-type appointment system in outpatient clinics, Int. J. Med. Inform. 70 (1) (2003) 31–40.
[31] J.R. Swisher, S.H. Jacobson, J.B. Jun, O. Balci, Modeling and analyzing a physician clinic environment using discrete-event (visual) simulation, Comput. Oper. Res. 28 (2) (2001) 105–125.
[32] L.V. Green, J. Soares, Computing time-dependent waiting time probabilities in $M(t)/M/s(t)$ queueing systems, Manuf. Serv. Oper. Manage. 9 (1) (2007) 54–61.
[33] L.V. Green, P.J. Kolesar, J. Soares, Improving the SIPP approach for staffing service systems that have cyclic demands, Oper. Res. 49 (4) (2001) 549–564.
[34] L.V. Green, J. Soares, J.F. Giglio, R.A. Green, Using queueing theory to increase the effectiveness of emergency department provider staffing, Acad. Emerg. Med. 13 (1) (2006) 61–68.
[35] H. Bruneel, I. Wuyts, Analysis of discrete-time multiserver queueing models with constant service times, Oper. Res. Lett. 15 (5) (1994) 231–236.
[36] H. Bruneel, Performance of discrete-time queueing systems, Comput. Oper. Res. 20 (3) (1993) 303–320.
[37] D.J. Worthington, Queueing models for hospital waiting lists, J. Oper. Res. Soc. 38 (5) (1987) 413–422.
[38] J. Goddard, M. Tavakoli, Efficiency and welfare implications of managed public sector hospital waiting lists, European J. Oper. Res. 184 (2) (2008) 778–792.
[39] H. Takagi, Queuing analysis of polling models, ACM Comput. Surv. (CSUR) 20 (1) (1988) 5–28.
[40] M. Ramakrishnan, D. Sier, P.G. Taylor, A two-time-scale model for hospital patient flow, IMA J. Manag. Math. 16 (3) (2005) 197.
[41] K.J. Klassen, T.R. Rohleder, Outpatient appointment scheduling with urgent clients in a dynamic, multi-period environment, Int. J. Serv. Ind. Manage. 15 (2) (2004) 167–186.
[42] G. Dobson, S. Hasija, E.J. Pinker, Reserving capacity for urgent patients in primary care, Prod. Oper. Manage. 20 (3) (2011) 456–473.
[43] N. Liu, S. Ziya, V.G. Kulkarni, Dynamic scheduling of outpatient appointments under patient no-shows and cancellations, Manuf. Serv. Oper. Manage. 12 (2) (2010) 347–364.
[44] X. Qu, R.L. Rardin, J.A.S. Williams, D.R. Willis, Matching daily healthcare provider capacity to demand in advanced access scheduling systems, European J. Oper. Res. 183 (2) (2007) 812–826.
[45] X. Qu, J. Shi, Effect of two-level provider capacities on the performance of open access clinics, Health Care Manage. Sci. 12 (1) (2009) 99–114.
[46] L.W. Robinson, R.R. Chen, A comparison of traditional and open-access policies for appointment scheduling, Manuf. Serv. Oper. Manage. 12 (2) (2010) 330–346.
[47] R. Kopach, P.C. DeLaurentis, M. Lawley, K. Muthuraman, L. Ozsen, R. Rardin, H. Wan, P. Intrevado, X. Qu, D. Willis, Effects of clinical characteristics on successful open access scheduling, Health Care Manage. Sci. 10 (2) (2007) 111–124.
[48] J. Patrick, M.L. Puterman, M. Queyranne, Dynamic multi-priority patient scheduling for a diagnostic resource, Oper. Res. 56 (6) (2008) 1507–1525.
[49] S. Creemers, M. Lambrecht, Queueing models for appointment-driven systems, Ann. Oper. Res. 178 (1) (2010) 155–172.
[50] I. Adan, J.S.H. van Leeuwaarden, E.M.M. Winands, On the application of Rouché's theorem in queueing theory, Oper. Res. Lett. 34 (3) (2006) 355–360.
[51] L. Kleinrock, Queueing Systems, Volume 1: Theory, John Wiley & Sons, London, UK, 1975.

Nikky Kortbeek (1983) received his M.Sc. degree in 2008 in Operations Research & Management from the University of Amsterdam. After being a research fellow for one year at the University of Amsterdam, he joined the Department of Applied Mathematics of the University of Twente (UT), and received the Ph.D. degree with distinction for the dissertation "Quality-driven efficiency in healthcare" in 2012. He is a Program Leader of healthcare logistics at the Academic Medical Center Amsterdam (AMC) and a Postdoctoral Researcher of the UT research center CHOIR (Center for Healthcare Operations Improvement and Research). His research interest is to develop mathematical techniques that help and guide healthcare professionals in making their organizations more effective and efficient.

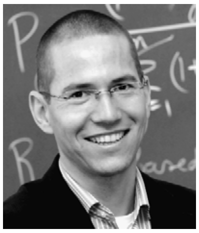
Maartje E. Zonderland (1982) received B.Sc. degrees in Industrial Engineering (2003) and Applied Mathematics (2006), and an M.Sc. degree in Applied Mathematics (2007) from the University of Twente. In 2012, she received her Ph.D. degree for the dissertation "Curing the Queue". During her Ph.D., she also worked as a Staff Consultant for Leiden University Medical Center, which is one of the eight academic hospitals in the Netherlands. Currently she has her own consulting firm, *Zonderland ZorgLogistiek*. The focus of most of her projects is quantitative support of decision making processes in healthcare organizations.

Aleida Braaksma received her M.Sc. degree in Applied Mathematics from the University of Twente in 2010. She is currently a Ph.D. candidate at the Academic Medical Center (AMC) in Amsterdam and at the Center for Healthcare Operations Improvement and Research (CHOIR) at the University of Twente. Next to this position, she is working as a consultant for process optimization in healthcare in the AMC. Her research interests are in planning and scheduling in healthcare, focusing on integral improvements for multiple care providers and patient types.

Ingrid M.H. Vliegen received her Ph.D. in Industrial Engineering from Eindhoven University of Technology, The Netherlands, in 2009. Since May 2010, she is working as an Assistant Professor within the research group CHOIR and at the Department of Industrial Engineering and Business Information Systems of the University of Twente, The Netherlands. Her research focuses on capacity modeling in health care, specifically for good flows and maintenance.

Richard J. Boucherie (1964) received M.Sc. degrees in 1988 in Applied Mathematics (Stochastic Operations Research) and Theoretical Physics (Statistical Physics) from the Universiteit Leiden, and received the Ph.D. degree in Econometrics in 1992 for a thesis on product-form in queuing networks from the Vrije Universiteit, Amsterdam. Following Post docs at INRIA Sophia Antipolis, CWI Amsterdam, and Universiteit van Amsterdam, since 2000 he is in the Department of Applied Mathematics of the University of Twente, where he was appointed in 2003 as Full Professor of Stochastic Operations Research. His research interests are in queuing theory and Petri nets with application areas including sensor networks and healthcare. He is Chair of Industrial Engineering and Operations Research graduate program and Co-founder of the UT research center CHOIR (Center for Healthcare Operations Improvement and Research) in the area of healthcare logistics.

Nelly Litvak obtained her M.Sc. in Applied Mathematics from Nizhny Novgorod State University, Russia, in 1995. She received her Ph.D. in Stochastic Operations Research from Eindhoven University of Technology (EURANDOM) in 2002. From 2002 she has been appointed as an Assistant Professor, and from 2012 as an Associate Professor in the Stochastic Operations Research Group at the University of Twente. Her research interests are in complex networks, queuing theory, and healthcare logistics. She is a Managing Editor of the Internet Mathematics journal.



Erwin W. Hans (1974) received his M.Sc. degree in 1996 in Applied Mathematics from the University of Twente, and received the Ph.D. degree in Applied Mathematics in 2001 for a thesis on mathematical models for tactical capacity planning in discrete manufacturing, from the University of Twente. Since 2001 he has continued his academic career within the School of Management and Governance, where he was appointed as Full Professor of Operations Management in Healthcare in 2013. His research interests are in the application of operations research techniques, and operations management within the healthcare domain. He is Program Director of the Industrial Engineering & Management B.Sc. and M.Sc. programs, and Co-founder of the UT research center CHOIR (Center for Healthcare Operations Improvement and Research) in the area of healthcare logistics.