Omega 57 (2015) 145–156

Contents lists available at ScienceDirect

Omega

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

Appointment scheduling in tandem-type service systems ☆

Alex Kuiper *, Michel Mandjes

IBIS UvA, University of Amsterdam, Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Received 17 July 2014
Accepted 21 April 2015
Available online 28 April 2015

Keywords:
Appointment scheduling
Tandem queue
Phase-type distribution
Healthcare

ABSTRACT

Appointment-based service systems arise in a broad variety of healthcare settings (for example an outpatient clinic or a dentist). Where most existing algorithms specifically consider the situation of the patient undergoing a single service, in many practical situations *multiple* services have to be *sequentially* performed. Modeling the service system as a tandem queue, the main objective of this paper is to generate schedules that soundly balance the interests of patients (i.e., low waiting times) and staff (i.e., low idle times). Importantly, following up on prior work for the single-node queue, we advocate a phase-type based technique that can deal with *any* service-time distribution (which may, in addition, vary across patients). Relying on a novel recursive scheme to evaluate the sojourn-time distribution of clients in such tandem systems, we show how optimal schedules can be computed. Our technique is illustrated by extensive numerical experimentation, also leading to practical guidelines that apply to a broad range of parameter settings.

1. Introduction

In healthcare appointment schedules are frequently used to overcome issues of excessive idle and waiting times. Obviously, when patients are allowed to freely choose when to arrive at practitioner, this may result in substantial waiting times (for patients) during peak moments, while there can be significant idle times (for medical staff) when there are no clients to be served.

A complicating factor is that service times are random. As a result, for any appointment schedule there is still the possibility that upon arrival a patient has to wait before being served. On the other hand, it may also happen that a practitioner finishes serving a patient earlier than that a next patient arrives, resulting in idle time. 'Good' appointment schedules strike an appropriate balance between these two undesired effects.

1.1. Setup

In more formal terms, an appointment scheduling system can be described as follows. Suppose that there are n patients, in the sequel also referred to as *clients*, to be scheduled, for instance on a specific day. Let, for each client $i \in \{1, \ldots, n\}$, W_i denote her waiting time, whereas I_i stands for the idle time *prior to* her arrival. A specific objective could be to determine the scheduled arrival epochs $(t_1, \ldots, t_n)$ of the n clients, so as to minimize the total expected idle and waiting time over the day. This means that we are to evaluate

$$\min_{t_1,\ldots,t_n} \sum_{i=1}^{n} (\mathbb{E}I_i + \mathbb{E}W_i). \tag{1}$$

Alternatively, the mean idle and waiting times can be weighted with appropriately chosen scalars, if it is felt that the interests of the practitioner and the clients should not be evenly valued. This problem fits in a queueing-theoretic framework: as soon as we have fixed the clients' arrival times, the resulting queueing system is, in Kendall's notation, a $D/G/1$ queue: deterministic (albeit not equally spaced) interarrival times, general service times, and a single server. Evaluation of the objective function evidently requires a technique to determine various queueing-related quantities; we wish to find the arrival epochs that minimize this objective function.

1.2. Literature

Problems of the sort of (1), and numerous variants, have been extensively considered in the literature; this body of work dates back to e.g. [3,23]. The history of research on finding a proper balance between the practitioner's and patients' interests (for instance by working with objective function incorporating waiting times and idle times) is extensively described in Cayirli and Veral [4]; see in particular Section 3 of that paper. We restrict ourselves to mentioning a couple of references that directly relate to our setup. Wang [21] considers the case in which the service times of the clients have a phase-type distribution, which allows efficient evaluation of the mean idle and waiting times.

☆This manuscript was processed by Associate Editor Ghate.

* Corresponding author.
E-mail address: a.kuiper@uva.nl (A. Kuiper).

http://dx.doi.org/10.1016/j.omega.2015.04.009
0305-0483/

This idea is further exploited in Kuiper et al. [12]: a systematic approach is presented in which the (generally distributed) service times are replaced by their phase-type counterparts (fitting the first two moments), which is then validated in detail for a wide variety of service-time distributions and objective functions (i.e., not necessarily the linear form featured in (1)). Where Kuiper et al. [12] and Wang [21] suggest a phase-type fit, the main idea of Lau and Lau [13] is to use a fit of the first four moments by means of a beta distribution (having four parameters); this approach turns out to have attractive computational features.

A fundamentally different approach is followed in De Vuyst et al. [8]: generating functions are intensively used as a tool for fast and accurate computation of the underlying objective function. In Kemper et al. [11] the arrival epochs are determined on a client-by-client basis, thus substantially simplifying the underlying optimization problem: for an objective function of the form (1) it was shown that the arrival epoch of the i-th client should equal the sum of the medians of the sojourn times of the previous $i-1$ clients; a somewhat related approach has been proposed by Weiss [22]. Luo et al. [14] propose heuristics which incorporate a number of additional features, such as cancellations and no-shows.

As pointed out above, the problem of finding appropriate schedules critically depends on the availability of the clients' service-time distributions, as these determine the idle and waiting times. Importantly, the service times are typically not exponentially distributed; depending on the application at hand they may have coefficients of variation significantly different from 1. We refer to, e.g. Appendix A of Cayirli and Veral [4] for a detailed account of the properties of typical service times; it is reported that, in the situations studied, the coefficient of variation varies roughly from 0.35 to 0.85. As a consequence, one would ideally rely on a methodology that can in principle deal with *any* service-time distribution, for instance characterized in terms of its first two moments (or, equivalently, the mean and coefficient of variation). As was mentioned above, such a procedure was proposed (and extensively validated) in Kuiper et al. [12] for the single-node appointment scheduling system; the underlying idea is that the service times are approximated by their phase-type counterparts, for which the distributions of the idle and waiting times can be explicitly determined and relatively easily numerically evaluated.

1.3. Contribution

It is important to notice that a practical limitation of the above setting is that in many situations in healthcare, patients do not necessarily undergo just one service. Instead, patients may sequentially be served at multiple service stations (or: *nodes*). There are numerous examples of this, such as a patient who first has an x-ray made and then sees a doctor, or a patient who first has an intake and is then examined by a doctor. In those contexts, representing the system as a single queue is obviously not appropriate: one should rather consider a (two-node) *tandem* network (sometimes referred to as an $D/G/1\rightarrow G/1$ queue), where the individual queues correspond to the two service stages.

Importantly, despite the relevance of multi-stage systems, the vast majority of all papers focuses on single nodes; see e.g. some remarks on this in Section 2.1 of Cayirli and Veral [4]. Notable exceptions that *do* cover multi-node situations are the case study (backed by Monte Carlo simulation) presented by Rising et al. [16] and the visual simulation-based approach due to Swisher et al. [18]. An elementary queueing model, designed for a specific multi-stage application (i.e., an ear, nose and throat outpatient clinic), has been developed by Cox et al. [6]. While there is a variety of situations in which single-stage systems are a sufficiently accurate representation of the real system, one would ideally like to have appointment scheduling algorithms that can deal with more complex structures as well, such as the ones presented by Côté and Stein [5].

It is noted that, to set up appointment schedules one needs to be able to evaluate the transient distribution of the underlying queueing model; this transient distribution facilitates the computation of an objective function, which is then to be optimized over the arrival epochs. Single queues typically have a reasonable level of tractability, but (multi-node) queueing networks are known to allow such an explicit transient analysis only in specific cases (e.g. Jackson networks, relying heavily on various restrictive exponentiality assumptions). In light of this, our paper is among the first contributions to appointment scheduling in a multi-stage context. Importantly, our framework does not impose any restrictive assumptions on the service-time distributions.

The approach proposed in this paper uses the transient distribution of the tandem queue to set up schedules. In systems in which the number of clients to be scheduled is relatively large and in which (per node) the clients service times stem from the same distribution, however, we can work with the corresponding *stationary* distributions. The second main novelty of this paper lies in the way we evaluate such steady-state distributions; it is noted that the approach we present here is significantly more efficient than the one we developed earlier in Kuiper et al. [12].

The primary application area of appointment scheduling lies in healthcare, but there is potential use in several other areas as well. In industrial applications, where jobs pass through multiple stations (e.g. machines) in a flow line, the cost function can be expressed in terms of holding cost and the (opportunity) cost of station idleness; the idea is then to schedule jobs so as to minimize this objective function (see e.g. [7]).

1.4. Organization

The structure of the paper is as follows. In Section 2 we state the scheduling problem for the two-node tandem in terms of idle and waiting times, which will be addressed in the rest of this paper. Section 3 explains in detail how one can exploit phase-type characterizations of the service-time distributions to compute idle and waiting times; various extensions of the 'base model' (viz. heterogeneous service-time distributions, the situation in which the second node may 'block' the first node) are dealt with in Section 4. Then, in Section 5, we use the developed methodology to numerically compute optimal schedules and study the effect of various parameters on the optimal schedule. We also see that, for the special case that all service times are equally distributed, schedules in this transient setting rapidly approach steady-state, and hence one could approximate the transient schedule by its stationary counterpart (which has the evident advantage of being easier to evaluate). In Section 6 we demonstrate an efficient technique to compute the optimal steady-state schedule, and we use this procedure to evaluate such schedules (thus showing the impact of the various model parameters). The paper is concluded by a brief discussion in Section 7.

2. Problem description

As argued in Introduction, appointment schedules are intended to properly balance the 'disutilities' experienced by both the server (i.e., the practitioner) and the clients (i.e., the patients). More concretely, the schedules should be such that the server's idle time is kept sufficiently low, while at the same time controlling the clients' waiting times. In this section, we first recapitulate how a mathematical framework can be set up in the single-server setting, and then extend this to the two-node tandem.

A central role is played by the notion of a *risk function*, measuring the system's (aggregate) disutility, which captures the effect of idle times and waiting times in a single expression. A common choice (see e.g. [20], [9], [12]) is the (potentially weighted) sum of the mean idle times and the mean waiting times, i.e.,

$$R(t_1,\ldots,t_n)=\sum_{i=1}^{n}\left(\beta\mathbb{E}[I_i]+(1-\beta)\mathbb{E}[W_i]\right)\quad\text{with }\beta\in(0,1);\tag{2}$$

here the t_is (for $i\in\{1,\ldots,n\}$) denote the arrival epochs of the n clients, W_i is the waiting time of the i-th client, and I_i is the idle time prior to the arrival of the i-th client. Observe that shifting the value of β amounts to trading off the interests of the server and the clients.

The idea is to balance idle and waiting times by optimizing the risk function (2) over all arrivals epochs $0\le t_1\le\ldots\le t_n$:

$$\min_{t_1,\ldots,t_n}R(t_1,\ldots,t_n)=\min_{t_1,\ldots,t_n}\sum_{i=1}^{n}\left(\beta\mathbb{E}[I_i]+(1-\beta)\mathbb{E}[W_i]\right).\tag{3}$$

As pointed out in Kemper et al. [11], this optimization problem, which phrased in terms of mean idle and waiting times, can also be expressed in terms of the clients' sojourn times. More precisely, it turns out that, as a direct consequence of the Lindley recursion,

$$\min_{t_1,\ldots,t_n}\sum_{i=1}^{n}\left(\beta\mathbb{E}[I_i]+(1-\beta)\mathbb{E}[W_i]\right)=\min_{x_1,\ldots,x_{n-1}}\sum_{i=1}^{n-1}\mathbb{E}[\ell(S_i-x_i)],\tag{4}$$

where $\ell(\cdot)$, evaluated in S_i-x_i denotes the so-called *loss function*, defined for any $x\in\mathbb{R}$ by

$$\ell(x):=-\beta x\mathbf{1}_{\{x<0\}}+(1-\beta)x\mathbf{1}_{\{x>0\}},$$

the variable S_i is the sojourn time of the i-th client (i.e., waiting time W_i plus service time B_i), and the non-negative numbers $x_i:=t_{i+1}-t_i$ (with $t_1=0$) correspond to the *interarrival times*.

In the tandem setting clients have to be sequentially served by two servers. It is throughout assumed that the service times of client i at node r, for $i\in\{1,\ldots,n\}$ and $r\in\{1,2\}$, are independent non-negative random variables $B_{r,i}$. As before, appointment schedules are sequences of epochs $t_1,\ldots,t_n$ at which the n clients are supposed to arrive at the first node. However, now both servers generate their own risk, so that the problem we are faced with is to find $t_1,\ldots,t_n$ that minimize a risk function that incorporates idle times and waiting times at *both* nodes. In self-evident notation, we are therefore to evaluate

$$\min_{t_1,\ldots,t_n}\sum_{i=1}^{n}\left\{w\left(\beta\mathbb{E}[I_{1,i}]+(1-\beta)\mathbb{E}[W_{1,i}]\right)+(1-w)\left(\delta\mathbb{E}[I_{2,i}]+(1-\delta)\mathbb{E}[W_{2,i}]\right)\right\},\tag{5}$$

with $\beta,\delta,w\in(0,1)$. At each node, we balance the loss incurred by idle and waiting times, as before, reflected by β and δ respectively, whereas w weighs the disutilities corresponding to both nodes. Observe that setting w equal to 1 in (5) reduces to the familiar $D/G/1$ queue, i.e., the single-node appointment scheduling problem.

3. Methodology

The goal of this paper is to devise techniques to solve the optimization problem in (5) for general service times $B_{r,i}$. In this generality this is problematic, as evaluating the objective function essentially requires us to determine the sojourn-time distributions of the clients in our tandem system of the $D/G/1\to G/1$ type, for which no closed-form solution is available. We remedy this by relying on the approach proposed and validated in Kuiper et al. [12]: as we explain in Section 3.1, we approximate the service times by their *phase-type* counterparts, for which computations turn out to be feasible.

The second subsection points out how the mean waiting time and idle times, to be used in (5), can be computed from the mean sojourn times; it is thus sufficient to be able to determine the clients' sojourn-time distributions. Where Kuiper et al. [12] focused on the single $D/G/1$ queue, we demonstrate how to extend this procedure to its tandem counterpart. This tandem case turns out to be substantially more involved; we present in Section 3.3 in detail the (recursive) method that yields the sojourn-time distribution of each of the clients.

3.1. Phase-type distribution

In our study we use the idea, advocated in Tijms [19], to match the first and second moment of the service-time distribution by a so-called phase-type distribution. Observe that it is equivalent to fitting the mean and the *squared coefficient of variation* SCV; the SCV of a random variable is defined as its variance divided by the square of the mean. In line with Kuiper et al. [12], we choose to use a mixture of two Erlang distributions ($E_{K-1,K}(\mu;p)$) in case the service-time distribution has an SCV smaller than 1; an exponential distribution in case SCV equals one; and a hyperexponential distribution ($H_2(\boldsymbol{\mu};p)$) in case of an SCV larger than 1.

Next, we point out how to express the mixture of Erlang distributions, the exponential distribution, and hyperexponential distribution as a phase-type distribution. A phase-type distribution is characterized by a 'dimension' $m\in\mathbb{N}$, an m-dimensional row vector $\boldsymbol{\alpha}$ with nonnegative entries adding up to 1, and an $(m\times m)$-dimensional matrix $\boldsymbol{S}=(s_{ij})_{i,j=1}^{m}$ such that $s_{ii}<0$, $s_{ij}\ge 0$, and $\sum_{j=1}^{m}s_{ij}\le 0$ for any $i\in\{1,\ldots,m\}$. If B has a phase-type distribution with representation $(\boldsymbol{\alpha},\boldsymbol{S})$—which we denote by $B=_{\mathrm{d}}\mathbb{P}\mathrm{h}(\boldsymbol{\alpha},\boldsymbol{S})$,—then its first moment equals

$$\mathbb{E}[B]=-\boldsymbol{\alpha}\boldsymbol{S}^{-1}\mathbf{1}_m,\tag{6}$$

where $\mathbf{1}_m$ is an all-one column vector of dimension m; higher moments can be given in closed form as well, as can be found in e.g. ([2], SectionIII.4).

As indicated above, the following three types of phase-type distributions cover all values of the mean and SCV.

- In case SCV <1, we use an $E_{K-1,K}(\mu;p)$ distribution, which corresponds to an Erlang distribution of $K-1$ phases and mean $(K-1)/\mu$ with probability p, and an Erlang distribution with K phases and mean K/μ with probability $1-p$. Then $m=K$, and the vector $\boldsymbol{\alpha}$ is such that $\alpha_1=1$ and $\alpha_i=0$ for $i=2,\ldots,K$. In addition $s_{ii}=-\mu$ for $i=1,\ldots,K$ and $s_{i,i+1}=-s_{ii}=\mu$ for $i=1,\ldots,K-2$, while $s_{K-1,K}=(1-p)\mu$; all other entries are 0. The corresponding SCV equals
$$\frac{K-p^2}{(K-p)^2},$$
which lies between $1/(K-1)$ and $1/K$ for $K\in\{2,3,\ldots\}$. We can thus uniquely identify an $E_{K-1,K}(\mu;p)$ distribution matching the first two moments of the target distribution, as long as SCV <1.
- In case SCV $=1$, we use an Exp(μ) distribution. Then $m=1$, $\alpha_1=1$ and $\boldsymbol{S}=s_{11}=-\mu$.
- In case SCV >1, we use a $H_2(\boldsymbol{\mu};p)$ distribution: we sample from Exp(μ_1) distribution with probability p, and from an Exp(μ_2) distribution with probability $1-p$. Then $m=2$, and $\alpha_1=p=1-\alpha_2$. Also, $s_{ii}=-\mu_i$, for $i=1,2$, while the other two entries of $\boldsymbol{S}$ equal 0. Notice that we have three parameters that we can freely choose to make sure that the first two moments match; to reduce the number of degrees of freedom by 1, we impose the additional condition of *balanced means*, i.e.,

$\mu_1 = 2p\mu$ and $\mu_2 = 2(1-p)\mu$ for some $\mu > 0$. The corresponding SCV then equals

$$\frac{1}{2p(1-p)} - 1,$$

which is larger than or equal to 1 (where we remark that it is obviously equal to 1 only if $p = 1/2$, corresponding to the exponential distribution).

3.2. Computing expected idle and waiting times

Section 3.3 presents an algorithm to compute the clients' sojourn-time distributions in both queues (where it is recalled that the sojourn time is the sum of the waiting time and the service time). Above we pointed out for the single-server queue that, with S_i denoting the sojourn time of the i-th client, our objective function can be expressed in terms of the loss function $\ell(\cdot)$, evaluated in $S_i - x_i$. This suggests that we need to know the full distribution of the sojourn times to be able to evaluate the objective function. Perhaps counter-intuitively, this is *not* the case, as we explain in this section: as it turns out, one only needs to know the *mean* sojourn times.

We show that the expected idle and waiting times at both nodes can be expressed in terms of the expected sojourn times. Let us first consider the first node. Realize that for all $i \in \{1, \ldots, n\}$, in self-evident notation,

$$\mathbb{E}[S_{1,i}] = \mathbb{E}[W_{1,i}] + \mathbb{E}[B_{1,i}]. \tag{7}$$

$\mathbb{E}[B_{1,i}]$ being known, we have found $\mathbb{E}[W_{1,i}]$ (in terms of the expected sojourn times, that is).

Now notice that the time the i-th client leaves the system can be expressed in two ways. In the first place it is the sum of the idle and service times of the first i clients, but in the second place also the arrival epoch of the i-th client plus her sojourn time. As a consequence, we have, for any client i,

$$\sum_{j=1}^{i} \left(\mathbb{E}[B_{1,j}] + \mathbb{E}[I_{1,j}]\right) = t_i + \mathbb{E}[S_{1,i}] = \sum_{j=1}^{i-1} x_j + \mathbb{E}[S_{1,i}]. \tag{8}$$

Hence we can recursively compute the expected idle time at the first server, prior to the arrival of the i-th client through

$$\mathbb{E}[I_{1,i}] = \mathbb{E}[S_{1,i}] - \mathbb{E}[B_{1,i}] + \sum_{j=1}^{i-1} \left(x_j - \mathbb{E}[B_{1,j}] - \mathbb{E}[I_{1,j}]\right).$$

A similar procedure works for the second node. Instead of looking at the first node only, we now consider S_i, i.e., the client-specific sojourn time when traversing both nodes:

$$\mathbb{E}[S_i] = \mathbb{E}[W_{1,i}] + \mathbb{E}[B_{1,i}] + \mathbb{E}[W_{2,i}] + \mathbb{E}[B_{2,i}]. \tag{9}$$

Noting that $\mathbb{E}[W_{1,i}]$ follows from Eq. (7), we are left to compute $\mathbb{E}[W_{2,i}]$. We now have, similar to (8),

$$\sum_{j=1}^{i} \left(\mathbb{E}[B_{2,j}] + \mathbb{E}[I_{2,j}]\right) = \sum_{j=1}^{i-1} x_j + \mathbb{E}[S_i] \tag{10}$$

(notice that $\mathbb{E}[I_{2,1}] > 0$ whereas $\mathbb{E}[I_{1,1}] = 0$). From the above we conclude that by knowing the clients' expected sojourn time at the first server and in the total system, we are able to compute all expected idle and waiting times by the above formulas. In the next subsection we show how we can recursively generate the sojourn-time distributions.

3.3. Recursive procedure to compute the sojourn-time distribution

In this subsection we describe an algorithm that determines the sojourn-time distributions, assuming that the service times at both nodes have phase-type distributions. For the first node, the procedure relies on the principles developed in Wang [21]; the derivation of the sojourn-time distribution for the *entire* system (i.e., for each client i the time spent at the first node plus the time spent at the second node), however, is novel and more involved. More specifically, there are various ways to represent the jobs flowing through the tandem network, each having its own probabilistic description (and associated state space); the one we have chosen to work with in this paper keeps the dimensionality relatively low. It is noted that, when setting up such a description, there are various additional subtleties to be dealt with; see the way we introduce the 'idle states' † (single node), and $\dagger_1$ and $\dagger_2$ (tandem case) below.

In the sequel we assume that, for each server, the service times are independent and identically distributed, and that there is an independence between these two sequences of random variables. Let the service time at the first node follow a phase-type distribution in $\mathbb{P}\mathrm{h}(\boldsymbol{\alpha}^{(1)}, \boldsymbol{S}^{(1)})$, whereas for the service time at the second node we have the representation $\mathbb{P}\mathrm{h}(\boldsymbol{\alpha}^{(2)}, \boldsymbol{S}^{(2)})$; the dimensions of both phase-type distributions are m_1 and m_2, respectively.

It is noted, however, that the procedure we developed extends to independent, *non-identically* distributed service times, albeit at the expense of rather 'heavy' notation. This explains why we restrict ourselves to the case of (per node) identically distributed service times in this section; presenting the procedure directly in full generality obscures the reasoning behind it (but we point out how to deal with the 'heterogeneous case' in the next section).

3.3.1. Recursive procedure for the first node

To compute the sojourn-time distribution at the first server, we aim to derive the phase-type representation of the sojourn-time distribution of each client i at this node, that is, $S_{1,i} =_{\mathrm{d}} \mathbb{P}\mathrm{h}(\boldsymbol{\alpha}_i^{(1)}, \boldsymbol{S}_i^{(1)})$, where the subscript '1' is added to denote that for the moment we are only considering the first server. We first define the following bivariate process:

$$\{N_i(t), K_i(t), t \geq 0\} \tag{11}$$

for client $i = 1, \ldots, n$. Here $N_i(t)$ is the number of clients present in the system, t time units after the arrival of the i-th client; obviously $N_i(t) \in \{1, \ldots, i\}$. The second component, $K_i(t) \in \{1, \ldots, m_1\}$, represents the phase of the client in service t time units after the arrival of the i-th client. Observe that one state needs to be added to the state space $\{1, \ldots, i\} \times \{1, \ldots, m_1\}$, corresponding to the situation that at time t all i clients have left. We associate the symbol '†' with this state.

In the sequel the probabilities

$$p_{j,k}^{(i)}(t) := \mathbb{P}(N_i(t) = j, \quad K_i(t) = k)$$

play a crucial role, with $t \geq 0$, $i = 1, \ldots, n$, $j = 1, \ldots, i$, and $k = 1, \ldots, m_1$. It is evident that

$$\mathbb{P}(S_{1,i} \leq t) = \mathbb{P}((N_i(t), K_i(t)) = \dagger) = 1 - \sum_{j=1}^{i} \sum_{k=1}^{m_1} p_{j,k}^{(i)}(t). \tag{12}$$

In addition, we introduce the vector $\boldsymbol{P}_i(t)$ (of dimension $m_1 i$), defined by

$$\left(p_{i,1}^{(i)}(t), \ldots, p_{i,m_1}^{(i)}(t), p_{i-1,1}^{(i)}(t), \ldots, p_{i-1,m_1}^{(i)}(t), \ldots, p_{1,1}^{(i)}(t), \ldots, p_{1,m_1}^{(i)}(t)\right).$$

The sojourn-time distribution of the i-th client can be computed from $\boldsymbol{P}_i(t)$, as, by virtue of Eq. (12),

$$F_{1,i}(t) := \mathbb{P}(S_{1,i} \leq t) = 1 - \boldsymbol{P}_i(t)\mathbf{1}_{m_1 i};$$

here $\mathbf{1}_{m_1 i}$ represents an all-one column vector of dimension $m_1 i$. The question we now focus on, is how $\boldsymbol{P}_i(t)$ can be computed, for

$t \geq 0$, and $i \in \{1,\ldots,n\}$. In the sequel, $\mathbf{0}_{m\times n}$ denotes an $(m \times n)$ all-zero matrix.

- Considering the first client, to arrive at $t_1 = 0$, it is a standard result that $\boldsymbol{P}_1(t) = \boldsymbol{\alpha}^{(1)}\exp(\boldsymbol{S}^{(1)}t)$; conclude that, as a consequence,

  $$(\boldsymbol{\alpha}_1^{(1)}, \boldsymbol{S}_1^{(1)}) = (\boldsymbol{\alpha}^{(1)}, \boldsymbol{S}^{(1)}),$$

  thus defining the phase-type representation of $S_{1,1}$.
- Concerning the second client, arriving x_1 after the first client, realize that there are two scenarios: she can find still some work in the system upon her arrival, and she can find the system empty. It can be argued that it thus follows that the initial distribution of the phase-type distribution, associated with the sojourn time of client 2, reads

  $$\boldsymbol{\alpha}_2^{(1)} = (\boldsymbol{P}_1(x_1), \boldsymbol{\alpha}^{(1)}F_{1,1}(x_1)),$$

  a (row) vector of dimension $2m_1$. It then follows (with the same arguments as the ones used in [12], [21]) that

  $$\boldsymbol{P}_2(t) = (\boldsymbol{P}_1(x_1), \boldsymbol{\alpha}^{(1)}F_{1,1}(x_1))\exp(\boldsymbol{S}_2^{(1)}t) \tag{13}$$

  (being an object of dimension $2m_1$ as well); here, with $\boldsymbol{s}^{(1)} := -\boldsymbol{S}^{(1)}\mathbf{1}_{m_1}$, and

  $$\boldsymbol{S}_2^{(1)} := \begin{pmatrix} \boldsymbol{S}^{(1)} & \boldsymbol{S}^{(1)}\alpha^{(1)} \\ \mathbf{0}_{m_1\times m_1} & \boldsymbol{S}^{(1)} \end{pmatrix}.$$

  We have thus identified $(\boldsymbol{\alpha}_2^{(1)}, \boldsymbol{S}_2^{(1)})$, i.e., the phase-type representation of $S_{1,2}$.
- The sojourn-time distributions of the other clients can be found recursively in a similar manner. To this end, we introduce the matrix $\boldsymbol{T}_i$ of dimension $(i-1)m_1 \times m_1$ through

  $$\boldsymbol{T}_i := \left(\mathbf{0}_{m_1\times m_1}, \ldots, \mathbf{0}_{m_1\times m_1}, \boldsymbol{s}^{(1)}\boldsymbol{\alpha}^{(1)}\right)^{\mathrm{T}};$$

  in addition, we introduce

  $$\boldsymbol{S}_i^{(1)} := \begin{pmatrix} \boldsymbol{S}_{i-1}^{(1)} & \boldsymbol{T}_i \\ \mathbf{0}_{m_1\times(i-1)m_1} & \boldsymbol{S}^{(1)} \end{pmatrix}.$$

  Then the row vector $\boldsymbol{P}_i(t)$ (of dimension $m_1 i$) can be found from $\boldsymbol{P}_{i-1}(t)$ (of dimension $m_1(i-1)$) by the recursion

  $$\boldsymbol{P}_i(t) = \left(\boldsymbol{P}_{i-1}(x_{i-1}), \boldsymbol{\alpha}^{(1)}F_{1,i-1}(x_{i-1})\right)\exp(\boldsymbol{S}_i^{(1)}t), \quad t \geq 0. \tag{14}$$

  This provides us with $(\boldsymbol{\alpha}_i^{(1)}, \boldsymbol{S}_i^{(1)})$.

Realize that for the specific phase-type distributions we are working with, the matrix $\boldsymbol{S}^{(1)}$ is upper triangular (in the hyper-exponential case in fact even diagonal), and hence so are the matrices $\boldsymbol{S}_i^{(1)}$, for $i \in \{1,\ldots,n\}$. As a consequence, the eigenvalues can be read off from the diagonal. This property facilitates easy computation of the matrix exponent $\exp(\boldsymbol{S}_i^{(1)}t)$; in case of the $E_{K-1,K}(\mu;p)$ distribution all eigenvalues are μ; and, in case of the $H_2(\boldsymbol{\mu};p)$ all eigenvalues are entries of the vector $\boldsymbol{\mu}$.

3.3.2. Recursive procedure for the two-node tandem

Where we above determined the sojourn-time distribution at the first queue, this subsection describes the extension of an algorithm that facilitates the computation of the distribution of the *total sojourn time*. More specifically, for each client we determine the phase-type distribution of the time she spends in the system, denoted by $S_i =_{\mathrm{d}} \mathrm{PH}(\boldsymbol{\alpha}_i, \boldsymbol{S}_i)$. Such a sojourn time S_i covers the waiting times and service times at both nodes, and can be used to evaluate our objective function, by using the approach presented in Section 3.2.

To this end, we define the 'tandem counterpart' of (11): for client $i=1,\ldots,n$, we record the process,

$$\{L_{1,i}(t), L_{2,i}(t), t \geq 0\},$$

with, for $r=1,2$, $L_{r,i}(t) := (N_{r,i}(t), K_{r,i}(t))$. Here $N_{r,i}(t)$ is the number of clients present at the r-th server (i.e., clients who are waiting plus potentially a client who is in service), and $K_{r,i}(t)$ represents the phase of the client in service on the r-th server (for $r=1,2$), t time units after the arrival (at the first node) of the i-th client. Again we have to augment the state space; we do so by adding states '$\dagger_1$' ('$\dagger_2$', respectively), representing the situation that no clients are present at node 1 (node 2).

We will study the probabilities, for $j_1, j_2 \in \mathcal{J}_i$, where

$$\mathcal{J}_i := \{j_1 \in \{1,\ldots,i-1\}, j_2 \in \{1,\ldots,i-1\} : j_1 + j_2 \in \{1,\ldots,i\}\},$$

and $k_r \in \{1,\ldots,m_r\}$,

$$p^{(i)}_{j_1k_1j_2k_2}(t) := \mathbb{P}\left(L_{1,i}(t) = (j_1,k_1), L_{2,i}(t) = (j_2,k_2)\right),$$

as well as, for $j_r \in \{1,\ldots,i\}$, $k_r \in \{1,\ldots,m_r\}$,

$$p^{(i)}_{\dagger_1 j_2k_2}(t) := \mathbb{P}\left(L_{1,i}(t) = \dagger_1, L_{2,i}(t) = (j_2,k_2)\right),$$
$$p^{(i)}_{j_1k_1\dagger_2}(t) := \mathbb{P}\left(L_{1,i}(t) = (j_1,k_1), L_{2,i}(t) = \dagger_2\right).$$

If the number of clients in both queues is positive (say j_1 and j_2), the client in service at the r-th node can be in m_r states. This explains why we, in this situation, work with the vector (of dimension m_1m_2)

$$\boldsymbol{p}^{(i)}_{[j_1,j_2]}(t) = \left((p^{(i)}_{j_11j_21}(t),\ldots,p^{(i)}_{j_11j_2m_2}(t)),\ldots,(p^{(i)}_{j_1m_1j_21}(t),\ldots,p^{(i)}_{j_1m_1j_2m_2}(t))\right).$$

In addition, we have the vector of dimension m_1 covering the cases that the second queue is empty, and the first is not:

$$\boldsymbol{p}^{(i)}_{[j_1,\dagger_2]}(t) = \left(p^{(i)}_{j_11,\dagger_2}(t),\ldots,p^{(i)}_{j_1m_1,\dagger_2}(t)\right),$$

and a vector of dimension m_2 for the cases that the first queue is empty, and the second is not:

$$\boldsymbol{p}^{(i)}_{[\dagger_1,j_2]}(t) = \left(p^{(i)}_{\dagger_1j_21}(t),\ldots,p^{(i)}_{\dagger_1j_2m_2}(t)\right).$$

Let $\overline{\boldsymbol{p}}^{(i)}_{[j]}(t)$ correspond to all situations in which j clients are present, t time units after the arrival of the i-th client; by concatenating the vectors defined above, we obtain the following vector of dimension $m_1+m_2+(j-1)m_1m_2$:

$$\overline{\boldsymbol{p}}^{(i)}_{[j]}(t) := \left(\boldsymbol{p}^{(i)}_{[j,\dagger_2]}(t), \boldsymbol{p}^{(i)}_{[j-1,1]}(t),\ldots,\boldsymbol{p}^{(i)}_{[1,j-1]}(t), \boldsymbol{p}^{(i)}_{[\dagger_1,j]}(t)\right).$$

Finally, we define $\boldsymbol{P}^{(i)}(t)$ corresponding to all possible system states t time units after arrival of the i-th client:

$$\boldsymbol{P}^{(i)}(t) = \left(\overline{\boldsymbol{p}}^{(i)}_{[i]}(t),\ldots,\overline{\boldsymbol{p}}^{(i)}_{[1]}(t)\right);$$

the dimension of this vector is

$$m[i] := \sum_{j=1}^{i}((m_1+m_2)+(j-1)m_1m_2) = i(m_1+m_2)+\frac{1}{2}i(i-1)m_1m_2.$$

In order to compute the sojourn-time distribution, the option of both queues being empty does not need to be incorporated in the vector $\boldsymbol{P}^{(i)}(t)$, since we have

$$\begin{aligned} F_i(t) := \mathbb{P}(S_i \leq t) &= 1 - \sum_{j_1,j_2\in\mathcal{J}_i}\sum_{k_1=1}^{m_1}\sum_{k_2=1}^{m_2} p^{(i)}_{j_1k_1j_2k_2}(t) \\ &\quad - \sum_{j_1=1}^{i}\sum_{k_1=1}^{m_1} p^{(i)}_{j_1k_1,\dagger_2}(t) - \sum_{j_2=1}^{i}\sum_{k_2=1}^{m_2} p^{(i)}_{\dagger_1j_2k_2}(t) \\ &= 1 - \boldsymbol{P}^{(i)}(t)\mathbf{1}_{m[i]}. \end{aligned}$$

The goal is now to construct a (recursive) algorithm to identify $\mathbf{P}^{(i)}(t)$.

• For the first client, to arrive at $t_1 = 0$, we have

$$\boldsymbol{P}^{(1)}(t) = (\boldsymbol{\alpha}^{(1)}, \mathbf{0}_{m_2}) \exp\left(\begin{pmatrix} \boldsymbol{S}^{(1)} & \boldsymbol{s}^{(1)}\boldsymbol{\alpha}^{(2)} \\ \mathbf{0}_{m_2 \times m_1} & \boldsymbol{S}^{(2)} \end{pmatrix} t \right).$$

which is an (m_1+m_2)-dimensional object. As a consequence, we have for the phase-type description of the random variable S_1 that, with $\mathbf{0}_{m_2}$ an all-zero row vector of dimension m_2,

$$\boldsymbol{\alpha}_1 = (\boldsymbol{\alpha}^{(1)}, \mathbf{0}_{m_2}), \quad \boldsymbol{S}_1 = \begin{pmatrix} \boldsymbol{S}^{(1)} & \boldsymbol{s}^{(1)}\boldsymbol{\alpha}^{(2)} \\ \mathbf{0}_{m_2 \times m_1} & \boldsymbol{S}^{(2)} \end{pmatrix}.$$

• Concerning the second client, arriving x_1 time units after the first client, standard arguments yield that, using standard Kronecker notation,

$$\boldsymbol{\alpha}_2 = \left(\boldsymbol{p}^{(1)}_{[1,\dagger_2]}(x_1), \boldsymbol{\alpha}^{(1)} \otimes \boldsymbol{p}^{(1)}_{[\dagger_1,1]}(x_1), \mathbf{0}_{m_2}, \boldsymbol{\alpha}^{(1)} F_1(x_1), \mathbf{0}_{m_2} \right),$$

where the dimensions of these five vectors are m_1, $m_1 m_2$, m_2, m_1 and m_2, so that the whole vector has dimension $m[2] = 2(m_1+m_2)+m_1 m_2$, as desired. Now we wish to identify the matrix $\boldsymbol{S}_2$ (of dimension $m[2] \times m[2]$) corresponding to the phase-type representation of the distribution of S_2:

$$\boldsymbol{P}^{(\mathbf{2})}(t) = \boldsymbol{\alpha}_2 \exp(\boldsymbol{S}_2 t), \quad t \geq 0.$$

To this end, we first define the following two matrices, for ease sometimes leaving out the dimensions of the $\mathbf{0}$-matrices,

$$\boldsymbol{U}_2 := \begin{pmatrix} \boldsymbol{S}^{(1)} & -\boldsymbol{S}^{(1)}\mathbf{1}_{m_1}\boldsymbol{A}^{(0)} & \mathbf{0}_{m_1 \times m_2} \\ \mathbf{0}_{m_1 m_2 \times m_1} & \boldsymbol{S}^{(1)} \oplus \boldsymbol{S}^{(2)} & -\boldsymbol{S}^{(1)}\mathbf{1}_{m_1} \otimes \boldsymbol{I}_{m_2} \\ \mathbf{0}_{m_2 \times m_1} & \mathbf{0}_{m_2 \times m_1 m_2} & \boldsymbol{S}^{(2)} \end{pmatrix}$$

$$= \begin{pmatrix} \boldsymbol{S}^{(1)} & \boldsymbol{s}^{(1)}\boldsymbol{A}^{(\mathbf{0})} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{S}^{(1)} \oplus \boldsymbol{S}^{(2)} & \boldsymbol{s}^{(1)} \otimes \boldsymbol{I}_{m_2} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{S}^{(2)} \end{pmatrix}$$

and

$$\boldsymbol{V}_2 := \begin{pmatrix} \mathbf{0}_{m_1 \times m_1} & \mathbf{0}_{m_1 \times m_2} \\ -\boldsymbol{I}_{m_1} \otimes \boldsymbol{S}^{(2)}\mathbf{1}_{m_2} & \mathbf{0}_{m_1 m_2 \times m_2} \\ \mathbf{0}_{m_2 \times m_1} & -\boldsymbol{S}^{(2)}\mathbf{1}_{m_2}\boldsymbol{\alpha}^{(2)} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{I}_{m_1} \otimes \boldsymbol{s}^{(2)} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{s}^{(2)}\boldsymbol{\alpha}^{(2)} \end{pmatrix},$$

where $\boldsymbol{A}^{(0)} := \boldsymbol{\alpha}^{(1)} \otimes \boldsymbol{\alpha}^{(2)}$, $\mathbf{1}_{m_r}$ the identity matrix of dimension m_r and $\boldsymbol{s}^{(r)} := -\boldsymbol{S}^{(r)}\mathbf{1}_{m_r}$. It is concluded that $\boldsymbol{U}_2$ is a square matrix with $m_1+m_2+m_1 m_2$ rows and columns, whereas $\boldsymbol{U}_2$ is of dimension $(m_1+m_2+m_1 m_2) \times (m_1+m_2)$. We can now construct the $(m[2] \times m[2])$-dimensional matrix $\boldsymbol{S}_2$ by

$$\boldsymbol{S}_2 = \begin{pmatrix} \boldsymbol{U}_2 & \boldsymbol{V}_2 \\ \mathbf{0} & \boldsymbol{S}_1 \end{pmatrix}.$$

•For the other clients, the same iterative procedure can be followed. We first define the following two 'start matrices', relating to which server starts serving a new client:

$$\boldsymbol{A}^{(1)} := \boldsymbol{\alpha}^{(1)} \otimes \boldsymbol{I}_{m_2} \quad \text{and} \quad \boldsymbol{A}^{(2)} := \boldsymbol{I}_{m_1} \otimes \boldsymbol{\alpha}^{(2)}.$$

In addition, we introduce the following vector of dimension $m_1+m_2+jm_1 m_2$, for $j \in \{1, \ldots, i\}$:

$$\check{\boldsymbol{p}}^{(i)}_{[j]}(t) := \left(\boldsymbol{p}^{(i)}_{[j,\dagger_2]}(t), \boldsymbol{p}^{(i)}_{[j-1,1]}(t), \ldots, \boldsymbol{p}^{(i)}_{[1,j-1]}(t), \boldsymbol{\alpha}^{(1)} \otimes \boldsymbol{p}^{(i)}_{[\dagger_1,j]}(t), \mathbf{0}_{m_2} \right).$$

Regarding the start distribution corresponding to the phase-type description of the sojourn time S_i, it follows that

$$\boldsymbol{\alpha}_i = \left(\check{\boldsymbol{p}}^{(i-1)}_{[i-1]}(x_{i-1}), \ldots, \check{\boldsymbol{p}}^{(i-1)}_{[1]}(x_{i-1}), \boldsymbol{\alpha}^{(1)} F_{i-1}(x_{i-1}), \mathbf{0}_{m_2} \right),$$

which can be verified to be of dimension $m[i]$.

Regarding the matrix $\boldsymbol{S}_i$ in $\boldsymbol{P}^{(i)}(t) = \boldsymbol{\alpha}_i \exp(\boldsymbol{S}_i t)$, this has the form

$$\boldsymbol{S}_i = \begin{pmatrix} \boldsymbol{U}_i & \boldsymbol{V}_i \\ \mathbf{0} & \boldsymbol{S}_{i-1} \end{pmatrix}.$$

Here

$$\boldsymbol{U}_i := \begin{pmatrix} \boldsymbol{S}^{(1)} & \boldsymbol{s}^{(1)}\boldsymbol{A}^{(0)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{S}^{(1)} \oplus \boldsymbol{S}^{(2)} & \boldsymbol{s}^{(1)} \otimes \boldsymbol{A}^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & & \ddots & \ddots & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{S}^{(1)} \oplus \boldsymbol{S}^{(2)} & \boldsymbol{s}^{(1)} \otimes \boldsymbol{A}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{S}^{(1)} \oplus \boldsymbol{S}^{(2)} & \boldsymbol{s}^{(1)} \otimes \boldsymbol{I}_{m_2} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{S}^{(2)} \end{pmatrix},$$

of dimension $((i-1)m_1 m_2 + m_1 + m_2) \times ((i-1)m_1 m_2 + m_1 + m_2)$, and

$$\boldsymbol{V}_i = \begin{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{I}_{m_1} \otimes \boldsymbol{s}^{(2)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{s}^{(2)} \otimes \boldsymbol{A}^{(2)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & & \ddots & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{s}^{(2)} \otimes \boldsymbol{A}^{(2)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{S}^{(2)}\boldsymbol{\alpha}^{(2)} \end{pmatrix}, \mathbf{0} \end{pmatrix}$$

of dimension $((i-1)m_1 m_2 + m_1 + m_2) \times m[i-1]$. We conclude that $\boldsymbol{S}_i$ indeed has dimension $m[i] \times m[i]$.

It can be verified that the analysis simplifies greatly in the case in which both service times have exponential distributions, $B_{r,i} =_{\mathrm{d}} \mathbb{P}\mathrm{h}(1, \mu_r)$ for $r = 1,2$. In that situation, one needs to record only the number of clients present at both nodes (as a consequence of the fact that a service time corresponds to just a single exponential phase).

4. Extensions

In the previous section we have considered our 'base model'; in this section we point out how a number of variants can be dealt with. In the first subsection we consider the situation of heterogeneous service times, whereas the second subsection concentrates on models in which the second node may 'block' the first node.

4.1. Heterogeneous service-time distributions

The model analyzed in the previous section considers the situation in which at each station r (for $r = 1,2$) the n service times, say $B_{r,1}$ up to $B_{r,n}$, are i.i.d. samples, distributed as a random variable B_r; importantly, the distributions of B_1 and B_2 do not necessarily coincide. We already indicated in Section 3.3 that our procedure extends to the situation in which the service-time distributions (at each of the nodes) are *client-specific*: i.e., each of the $B_{r,i}$ has an own distribution. As this extension is notationally involved, we restrict ourselves to explaining the main ideas behind it. We let job $B_{r,i}$ corresponds to a phase-type representation $(\boldsymbol{\alpha}^{(r,i)}, \boldsymbol{S}^{(r,i)})$, for $r = 1,2$ and $i = 1, \ldots, n$, with 'dimension' $m_{r,i} \in \mathbb{N}$.

We start our exposition by pointing out how the recursive procedure for the first node needs to be adapted. As it turns out, essentially all steps carry over after minor modifications. The vector $\boldsymbol{P}_i(t)$, defined as in Section 3.3, has now dimension $\overline{m}_i := m_{1,1} + \cdots + m_{1,i}$. Regarding the first client, we obviously have

$$\boldsymbol{P}_1(t) = \boldsymbol{\alpha}^{(1,1)} \exp(\boldsymbol{S}^{(1,1)} t).$$

For the second client, (13) still applies, but with $\boldsymbol{\alpha}^{(1)}$ replaced by

$\boldsymbol{\alpha}^{(1,2)}$ and $\boldsymbol{S}_2^{(1)}$ now being the $\overline{m}_2 \times \overline{m}_2$ matrix given by

$$\boldsymbol{S}_2^{(1)} := \begin{pmatrix} \boldsymbol{S}^{(1,1)} & \boldsymbol{s}^{(1,1)}\boldsymbol{\alpha}^{(1,1)} \\ \boldsymbol{0}_{m_{1,2}\times m_{1,1}} & \boldsymbol{S}^{(1,2)} \end{pmatrix},$$

where $\boldsymbol{s}^{(1,i)} := -\boldsymbol{S}^{(1,i)}\boldsymbol{1}_{m_{1,i}}$. This idea carries over to any client i, in the sense that the recursive relation (14) remains valid, but with $\boldsymbol{\alpha}^{(1)}$ replaced by $\boldsymbol{\alpha}^{(1,i)}$ and $\boldsymbol{S}_i^{(1)}$ now a matrix of size $\overline{m}_i \times \overline{m}_i$ defined as

$$\boldsymbol{S}_i^{(1)} := \begin{pmatrix} \boldsymbol{S}_{i-1}^{(1)} & \boldsymbol{T}_i \\ \boldsymbol{0}_{m_{1,i}\times \overline{m}_{i-1}} & \boldsymbol{S}^{(1,i)} \end{pmatrix},$$

where $\boldsymbol{T}_i$ is a matrix of size $\overline{m}_{i-1} \times m_{1,i}$:

$$\boldsymbol{T}_i := \left(\boldsymbol{0}_{m_{1,1}\times m_{1,i}}, \ldots, \boldsymbol{0}_{m_{1,i-2}\times m_{1,i}}, \boldsymbol{s}^{(1,i-1)}\boldsymbol{\alpha}^{(1,i)}\right)^{\mathrm{T}}.$$

In the same way we can extend the procedure for the two-node tandem (as developed in Section 3.3) to the situation of heterogeneous service times, but this becomes notationally rather involved. We therefore do not provide all details here, but restrict ourselves to a couple of general remarks.

In the first place, observe that a full description of our system now consists, at the moment that i clients have entered the system, of the number j_1 present at the first node and the number j_2 at the second node (where obviously $j_1 + j_2 = j \in \{0, 1, \ldots, i\}$), together with the phases of the clients in service. It is seen that client $\ell := i - j + 1$ is in service at node 2 (if $j_2 > 0$), since $i - j$ clients already left the system. This also means that clients $i - j + 1$ up to $i - j_1$ are present at the second node. It thus follows that the client in service there has a service-time distribution $B_{2,\ell}$ (represented by a phase-type distribution of dimension $m_{2,\ell}$). Likewise, clients $i - j_1 + 1$ up to i are present at node 1, with client $k := i - j_1 + 1$ in service (as long as $j_1 > 0$), with service-time distribution $B_{1,k}$ (represented by a phase-type distribution of dimension $m_{1,k}$).

The above extension allows us to study the effect of all sorts of correlations. If client i tends to take relatively long at both nodes (relative to the other clients), one could put this information into the random variables $B_{1,i}$ and $B_{2,i}$ (for instance by giving them larger means than the other clients).

4.2. Models with blocking

The general setup we have considered in Section 3.3 is a model in which there is an *infinite* buffer (i.e., waiting room) after stage 1, and thus clients waiting for service at the second node do not prevent the first node from processing work. Models in which there is such a blocking effect [7], however, are relevant in specific cases. They turn out to be relatively easy to model, and simpler than the base model. In this subsection we show how to adapt the base model to incorporate two common types of blocking; these adaptations to the model still follow the recursive methods outlined in Section 3.3. For ease we consider the case that for a given r the $B_{r,i}$ are distributed as a random variable B_r for all $i \in \{1, \ldots, n\}$, but the situation of heterogeneity among the $B_{r,i}$ can be dealt with as described in Section 4.1.

- In a first type of blocking, so-called 'blocking-before-service', the first server can only start a new job when the second server, is empty. Such a system can obviously be modeled by a single-node system, in which the phase-type representation of the per client service-time distribution B_i is derived by taking the convolution of the individual service times at both nodes (for client i represented by $B_{1,i} = {}_{\mathrm{d}}\mathbb{P}\mathrm{h}(\boldsymbol{\alpha}^{(1)}, \boldsymbol{S}^{(1)})$ and $B_{2,i} = {}_{\mathrm{d}}\mathbb{P}\mathrm{h}(\boldsymbol{\alpha}^{(2)}, \boldsymbol{S}^{(2)})$), that is,

$$B_i = {}_{\mathrm{d}}\mathbb{P}\mathrm{h}\left((\boldsymbol{\alpha}^{(1)}, \boldsymbol{0}_{m_2}), \begin{pmatrix} \boldsymbol{S}^{(1)} & \boldsymbol{s}^{(1)}\boldsymbol{\alpha}^{(2)} \\ \boldsymbol{0} & \boldsymbol{S}^{(2)} \end{pmatrix}\right).$$

  In addition, the server-specific costs cannot be differentiated between servers as the servers are considered as a single system. Therefore, it is natural to compute the objective function as in the single-node case; see Eq. (2).
- Another type of blocking is called 'blocking-after-service', a client can only move to the second node when this node is idle. It means that the client stays at the first node when she has been served at this first node, but cannot move on to the second node (as a consequence of the fact that there is a client being served there). With 'blocking' we refer to the situation that the next client-in-line cannot commence service at node 1, although the service time of the client in service has finished. As such, when 1 or 2 clients have entered the system, no blocking can occur; only for $i \geq 3$ one can be confronted with blocking. (For $i = 2$ the second server can still be serving the first client while the second client already finished her service at the first node, but this case does not require an adaptation of the algorithm presented in Section 3.3, since the second client is *de facto* waiting at the second node.)
  When $i \geq 3$ clients have entered the system, we adapt the procedure in the following way. The number of clients at the second server, as before denoted by j_2, is an element of $\{0, 1, 2\}$ (where '2' corresponds to the situation in which there is a client who blocks the first server, awaiting to be served at the second server). Furthermore, when $j \in \{3, \ldots, i\}$ clients are present in the system and $j_2 = 2$ (and hence the number of the clients at the first node, denoted by j_1, equals $j - 2$), then the system can only evolve by serving the client on the second server, that is, the service-time of the $(i - j + 1)$-th client at the second server should elapse. More precisely, let $\overline{\boldsymbol{p}}_{[ij]}^{(i,b)}(t)$ correspond to all situations in which j clients are present in our model with blocking, t time units after the arrival of the i-th client; we obtain the following vector of dimension $m_1 + m_2 + (\min\{j, 2\} - 1)m_1 m_2$:

$$\overline{\boldsymbol{p}}_{[ij]}^{(i,b)}(t) := \begin{cases} \left(\boldsymbol{p}_{[j,\dagger_2]}^{(i)}(t), \boldsymbol{p}_{[j-1,1]}^{(i)}(t), \boldsymbol{p}_{[j-2,2]}^{(i)}(t)\right) & \text{if } j \geq 3, \\ \left(\boldsymbol{p}_{[j,\dagger_2]}^{(i)}(t), \boldsymbol{p}_{[j-1,1]}^{(i)}(t), \boldsymbol{p}_{[\dagger_1 j]}^{(i)}(t)\right) & \text{if } j \leq 2. \end{cases}$$

Finally, let $\boldsymbol{P}^{(i,b)}(t)$ correspond to the probability vector related to all possible system states t time units after arrival of the i-th client:

$$\boldsymbol{P}^{(i,b)}(t) = \left(\overline{\boldsymbol{p}}_{[i]}^{(i,b)}(t), \ldots, \overline{\boldsymbol{p}}_{[1]}^{(i,b)}(t)\right);$$

the dimension of this vector is less than or equal to $m[i]$. The transitions given by the matrix $\boldsymbol{S}_i^b$ can be found by

$$\boldsymbol{S}_i^b = \begin{pmatrix} \boldsymbol{U}_i^b & \boldsymbol{V}_i^b \\ \boldsymbol{0} & \boldsymbol{S}_{i-1}^b \end{pmatrix},$$

where the matrices $\boldsymbol{U}_i^b$ and $\boldsymbol{V}_i^b$ can be constructed as in Section 3.3, but have just $(\min\{i, 2\} - 1)$ diagonal elements (instead of $i - 1$), due to the fact that when $j_2 = i - 2$ the first node is blocked, and only the second node is busy. The matrix $\boldsymbol{S}_i^b$ can then be used in $\boldsymbol{P}^{(i,b)}(t) = \boldsymbol{\alpha}_i^b \exp\left(\boldsymbol{S}_i^b t\right)$ (where the initial probabilities $\boldsymbol{\alpha}_i^b$ are adapted accordingly).
  The above setup enables us to compute the sojourn-time distributions. We can only use Eqs. (9) and (10) to compute the expected idle and waiting times, since the sojourn time at the first server is affected by the performance of the second server; due to the blocking effect the first server has to be analyzed separately. In the case of equal weights (i.e., $w = 0.5$ and $\beta = \delta$) this issue is trivially resolved. In other situations, one could opt for explicitly keeping track of the epoch that the first server finishes its service.

Importantly, where the above setup corresponds to the situation of no waiting room between the nodes, one can easily generalize the procedure to the case of $b \in \mathbb{N}$ positions in the waiting room.

5. Optimal schedules in a transient environment

In this section we present a numerical assessment related to the *transient* case, i.e., we determine, for various model instances, the optimal arrival times for n clients. The methodology outlined in the previous section enables us to compute the aggregate risk of a given schedule, and this risk is then to be minimized over the arrival epochs of the n clients (where the first client arrives at $t_1 = 0$), so as to obtain the optimal schedule. This minimization can be done relying on standard numerical packages.

Indeed, the phase-type representation, as obtained by the recursive method presented in the previous section, allows us to evaluate the sojourn-time distributions of the individual clients, and hence also the associated risk. Being able to compute optimal schedules, the impact of various parameters can be assessed. More specifically, in this section we perform such sensitivity analysis with respect to (i) both servers' SCVs; (ii) both servers' means; (iii) the weight parameter w. In all experiments we assume that clients are homogeneous, in that their service times at node 1 (node 2, respectively) are identically distributed.

In general, a schedule consisting of n clients can be written as a vector of n arrival epochs $(t_1, \ldots, t_n)$, or, equivalently, $n-1$ interarrival times $\boldsymbol{x} = (x_1, \ldots, x_{n-1})$. In the sequel we represent the optimal schedule by the vector of interarrival times $\boldsymbol{x}^\star = (x_1^\star, \ldots, x_{n-1}^\star)$.

5.1. Effect of coefficient of variation

First we examine the effect of the variability of the service times. In healthcare applications the typical range for the SCV is 0.35–0.85, see Cayirli and Veral [4]. In our experiments, however, we do not restrict the SCV to this range; realize that the method can be used in other areas as well, such as the planning of jobs in a manufacturing environment, in which potentially other SCVs apply.

In Fig. 1 we consider optimal schedule for various values of the SCVs, while keeping the mean service times fixed. In Fig. 1 we plot the optimal interarrival times when varying the SCV of the first server, whereas in Fig. 1 the SCV of the second server is varied. It is seen that the schedule has a so-called 'dome shape', as described by Kaandorp and Koole [10]: the optimal interarrival times are relatively short at the beginning (as there is still little uncertainty in the system) and the end (as there are few later clients suffering from long service times) of the schedule.

From the graphs we observe that the variability at the first server has a more pronounced impact. An explanation lies in the very nature of the tandem queue: variations in the service times at the first server are propagated to the second server. As a consequence, fluctuations in the service times at the first server affect the schedule more than additional variations at the second server.

5.2. Effect of mean

In Fig. 2, we systematically assess the effect of the mean service times on the schedule. Fig. 2(a) shows how the optimal schedule is affected by the mean service time at the first node, whereas Fig. 2(b) visualizes the effect of the mean service time at the second node. It is observed that these mean service times have less impact than the SCVs. More precisely, nearly until the last client, the computed optimal schedules behave virtually identically; only at the very end of the session we see a (mild) discrepancy. In addition, it is seen that in Fig. 2(a) the optimal interarrival times for the last client have clearly distinct values, whereas in Fig. 2(b) these are considerably closer together.

5.3. Effect of weight

We now assess the effects of the weight parameter w, by varying w from 0 to 1 (in steps of 0.2). We set the mean service times and coefficients of variation equal to 1 (at both servers). In the case $w=1$ we are optimizing over the first server only, i.e., we are in the setting of the well-known $D/M/1$ queue (with non-homogeneous arrival times), see e.g. Kuiper et al. [12]. The other extreme situation, $w=0$, is equivalent to only optimizing over the second server. The resulting schedules are presented in Fig. 3. From this graph we observe that the interarrival times essentially decrease in w. The reason for the phenomenon is that, in order to control the sojourn times in node 2 relatively long interarrival times are needed (compared to the sojourn times in node 1); this is an immediate consequence of the fact that node 2 is facing a non-deterministic arrival process (as opposed to node 1). As a result, giving node 1 more weight (i.e., increasing w) leads to 'more predictability in the objective function', and hence shorter optimal interarrival times. Similar graphs are obtained when choosing other values for the mean service times and coefficients of variation.

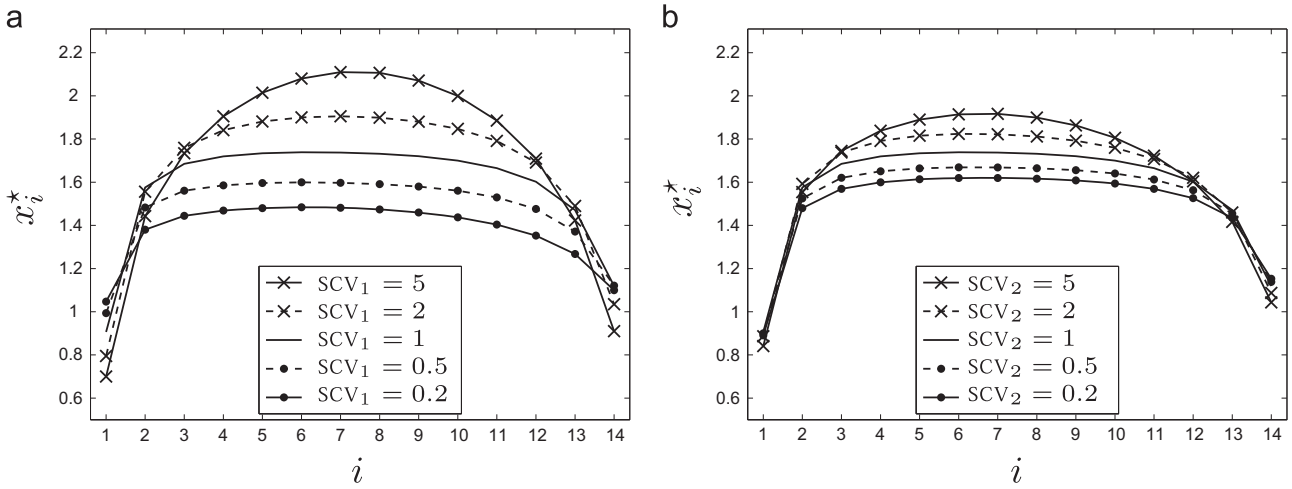

Fig. 1. The other parameters are kept such that $\mathbb{E}B_1 = \mathbb{E}B_2 = 1$ and $w=0.5$. (a) SCV$_1$ varies, while SCV$_2=1$. (b) SCV$_2$ varies, while SCV$_1=1$.

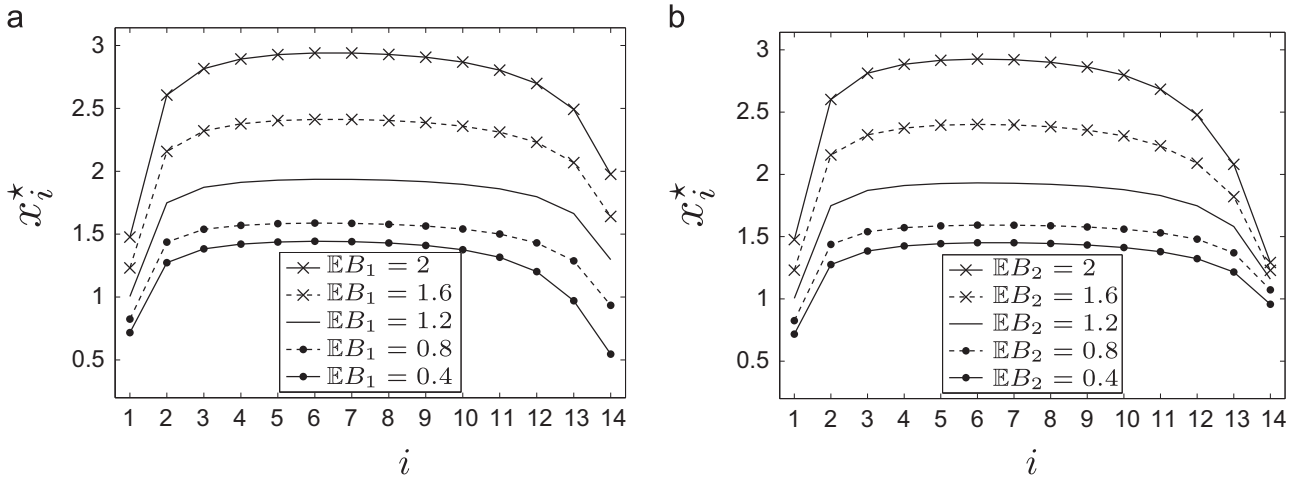

Fig. 2. The other parameters are kept such that $\text{SCV}_1 = \text{SCV}_2 = 1$ and $w = 0.5$. (a) $\mathbb{E}B_1$ varies, while $\mathbb{E}B_2 = 1$. (b) $\mathbb{E}B_2$ varies, while $\mathbb{E}B_1 = 1$.

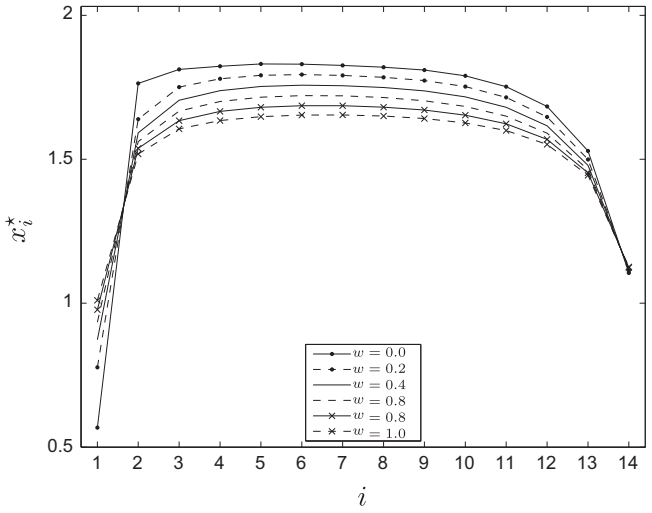

Fig. 3. The optimal schedule is computed for different weights $w \in [0, 1]$. The other parameters are kept such that $\mathbb{E}B_1 = \mathbb{E}B_2 = 1$ and $\text{SCV}_1 = \text{SCV}_2 = 1$.

5.4. Comparison with single-server system

Finally we study the difference between the two-node tandem with a corresponding single-node system. We consider the situation of a tandem network (with both mean service times equal to 1), and a single-server queue (with mean service time equal to 1). We set all SCVs equal to a half (which is a common setting in healthcare, being in the interval identified in Cayirli and Veral [4]). As observed above, the second node is fed by a non-deterministic arrival process, thus explaining that the optimal interarrival times in the two-node case are higher than those for the single node.

To be able to compare the per-client loss of the tandem network with the loss in the single-node setting, we consider the average of the two expected waiting times, and for the idle times we did the same. We see for both systems that the mean waiting times are increasing functions (turning from concave to convex somewhere in the middle). For the mean idle times we observe the familiar dome-shape pattern, cf. Fig. 4(a). Obviously the mean per-client loss in the two-node tandem is substantially higher than in the single-node system, as a result of the extra variation the second node is facing.

To further explore the effect of the tandem structure on the schedule, in relation *to* the corresponding single-node system, we also consider the corresponding optimal *steady-state* schedule; in Section 6 we point out how this schedule can be efficiently evaluated. It is stressed that transient solutions converge relatively rapidly to their steady-state counterparts, as is pictorially illustrated in Fig. 4(a); there we additionally plotted the optimal interarrival times in steady state, leading to the horizontal lines at 1.5363 and 1.4761 for the two-node tandem and the single-node system respectively. Another motivation for using steady-state schedules is that they are easier to compute and conceptually simpler than transient schedules, as they consist of just a single value.

6. Optimal schedules in steady state

In this section we consider the situation that the number of clients grows large, assuming that at both nodes the service-time distribution is identical across the clients. As a consequence, the optimal interarrival time tends to a constant, the *steady-state optimal interarrival time*, which we explain how to evaluate. As before, we restrict ourselves to the situation of phase-type service times at both nodes. The second part of this section presents a series of experiments.

The evaluation of the steady-state optimal interarrival time relies directly on the transition matrix computed, so as to compute the invariant distribution of the embedded discrete-time Markov chain. This idea reduces the computational effort drastically, in that it is not necessary to compute specific integrals and summations in the way proposed in Section 5.2 of Kuiper et al. [12] (borrowing elements from Wang [21]). To the best of our knowledge, this new approach to compute the stationary distribution of a $D/G/1$ or $D/G/1 \to G/1$ queue has not been pointed out before.

6.1. Procedure

The optimal interarrival time in steady state is particularly important, because, as indicated by the experiments reported in Kuiper et al. [12], schedules for a finite number of clients converge rapidly to their steady-state counterparts. In addition, as we will show, the steady-state solution can be determined with relatively low computational effort. It is further remarked that it can be used as an upper bound for transient schedules (e.g. the dashed lines in Fig. 4(a)). Our approach originates naturally from the phase-type framework featuring in Section 3.3. The method borrows elements from the one presented for the single node in Kuiper et al. [12], but is significantly more efficient.

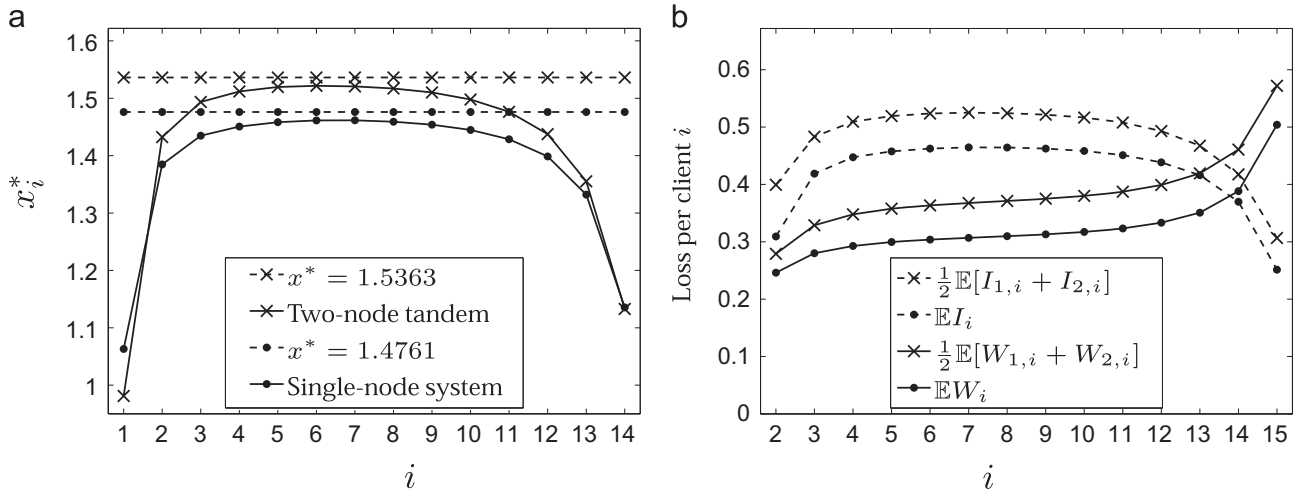

Fig. 4. The parameters are set such that every server operates with mean 1 and squared coefficient of variation of 0.5. (a) Comparing the optimal solution $\boldsymbol{x}^\star$. (b) Comparing the individual losses corresponding to $\boldsymbol{x}^\star$.

With b^u denoting the largest of the two mean service times, the optimal interarrival time, to be denoted by $x^\star$, should evidently be larger than b^u; we denote $\rho := b^u/x^\star$. It is further noted that, in contrast *to* the transient setting, the number of clients can attain any positive integer. However, when $\rho < 1$ holds, the stationary probability of having more than, say, M clients in the system decay essentially geometrically in M. This fact justifies truncating the state space such that, at both queues, we do not allow more than M clients to be simultaneously present.

In our stationary setting, the optimization problem can be rephrased as

$$\min_x w\big(\beta\mathbb{E}[I_1(x)]+(1-\beta)\mathbb{E}[W_1(x)]\big)+(1-w)\big(\delta\mathbb{E}[I_2(x)]+(1-\delta)\mathbb{E}[W_2(x)]\big), \tag{15}$$

where $W_k(x)$ ($I_k(x)$, respectively) is the steady-state waiting time (idle time, respectively) at node r ($r=1,2$) given the interarrival times are x. The mean idle and waiting times can be derived from the steady-state sojourn-time distribution, as pointed out in Section 3.2.

Now consider the number of clients in both queues, as well as the phase of the client in service (if any), *just before* arrival epochs (at the first node), i.e., the epochs $nx-$, for $n \in \mathbb{N}$). This process evidently constitutes a discrete-time Markov chain. Since the number of clients is truncated at M we can work with the matrix $\boldsymbol{S}_M$ that we identified in Section 3.3. The transition matrix of the embedded discrete-time process follows from the matrix exponent $\boldsymbol{Q}_M = \exp(\boldsymbol{S}_M x)$, where a minor correction needs to be applied, in order to take care of the arrival that takes place immediately after the 'embedded epochs'. In more detail, let $\boldsymbol{\pi}_{M-1}$ be the stationary probabilities, with the state space truncated at $M-1$. Starting with this vector $\boldsymbol{\pi}_{M-1}$ of dimension $m[M-1]+1$ (including the state of *no* clients in the system), we first perform a 'shift' by one (cf. the transient setting), due to the client arriving at nx, resulting in a vector of dimension $m[M]$. This vector can be multiplied by the transition matrix of the embedded discrete-time Markov chain $\boldsymbol{Q}_M$, and, because $\boldsymbol{\pi}_{M-1}$ was the stationary distribution, this should equal $\boldsymbol{\pi}_{M-1}$ again. Written in a compact way, we are therefore to solve

$$\boldsymbol{\pi}_{M-1} = t(\boldsymbol{\pi}_{M-1})\boldsymbol{Q}_M, \tag{16}$$

where the function $t(\cdot)$ corresponds to the shift operation applied to the vector $\boldsymbol{\pi}_{M-1}$, as described above. Using the normalizing equation $\boldsymbol{\pi}_{M-1} \cdot \mathbf{1}_{m[M-1]+1} = 1$ and Eq. (16), we find the equilibrium distribution. Having found this vector, the objective function can be evaluated, by the phase-type representation for the steady-state sojourn-time distribution, given by $\mathbb{P}\mathrm{h}(t(\boldsymbol{\pi}_{M-1}), \boldsymbol{S}_M)$. Having a procedure to evaluate the objective function for given x, we can then optimize it over $x^\star > b^u$.

Along the same lines one can derive the steady-state sojourn-time distribution for the first server only, which is computationally less involved. Combining both steady-state sojourn-time distributions, we can find all expected idle and waiting times by the relations derived in Section 3.2.

6.2. Computational results

In this subsection we evaluate the effect of (i) both SCVs (ii) the heterogeneity in the mean service times, i.e., $\mathbb{E}B_1$ and $\mathbb{E}B_2$, and (iii) the weight w on the steady-state optimal arrival time. To this end, we have considered 9 scenarios: all combinations of 3 different values of the weight w and three different values of $\mathbb{E}B_2$ (fixing, without loss of generality, $\mathbb{E}B_1$ at 1). For all these scenarios we let the SCVs of the two service times vary. In Fig. 5 the resulting graphs are given. The computational time per data point to compute the steady-state optimal interarrival time is less than 1 min, which is considerably less, roughly ten times, than computing the corresponding transient schedule for $n=25$ clients.

Perhaps the most striking observation from Fig. 5 is that, when moving from the top/right graph to the bottom/left graph, the level curves per figure change from nearly flat (gradient is 'orientated in the SCV_2 direction') to almost vertical (gradient is 'orientated in the SCV_1 direction'). Evidently, if w is close to 1 and service times in the first queue are substantially bigger than those in the second queue, then the impact of SCV_2 is small. Likewise, for w small and service times in the first queue being small relative to those in the second queue, then the impact of SCV_1 is small.

It is further observed that the level curves are nearly linear in SCV_1 and SCV_2. The exceptions to this rule are Fig. 5(a) and (i); in Fig. 5(a) both the weight of node 1 and the service time at node 2 are relatively high, whereas in Fig. 5(i) the weight of node 2 and the service time at node 1 are high. In addition, in some of the scenarios the distance between the contour lines is nearly constant.

An evident global conclusion is that both SCVs have a significant impact on the schedule. In the case that both nodes are equally important ($w=\frac{1}{2}$) and have similar means, that is, Fig. 5(e), we see that the first server's SCV has more impact than the second server. This finding is in line with the observations made in Section 5, where we argued that this is a consequence of the fact that the variability of the first queue propagates to the second queue.

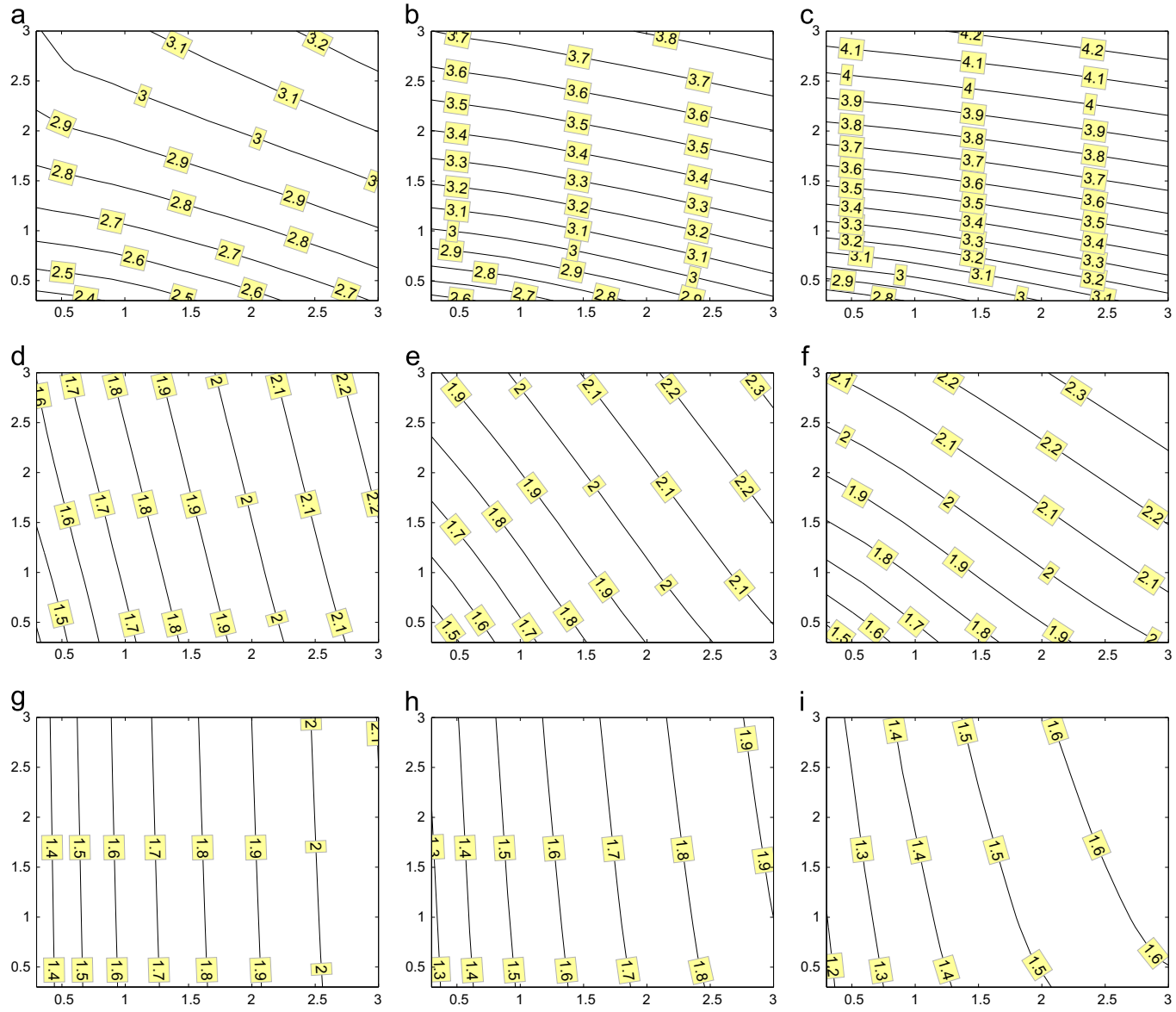

Fig. 5. The optimal interarrival times x^* as a function of SCVs for different scenarios. The SCV_1 is varying along the horizontal axis, while SCV_2 is varying along the vertical axis. The mean of the first server is set by $\mathbb{E}B_1 = 1$, while $\mathbb{E}B_2$ and w vary. (a) $w = 0.8$, $\mathbb{E}B_2 = 2$, (b) $w = 0.5$, $\mathbb{E}B_2 = 2$, (c) $w = 0.2$, $\mathbb{E}B_2 = 2$, (d) $w = 0.8$, $\mathbb{E}B_2 = 1$, (e) $w = 0.5$, $\mathbb{E}B_2 = 1$, (f) $w = 0.2$, $\mathbb{E}B_2 = 1$, (g) $w = 0.8$, $\mathbb{E}B_2 = 0.5$, (h) $w = 0.5$, $\mathbb{E}B_2 = 0.5$, (i) $w = 0.2$, $\mathbb{E}B_2 = 0.5$.

Moreover, our procedure makes it possible to verify whether it is justified to make use of steady-state schedules rather than their transient counterparts. In specific situations already after three clients the optimal interarrival times are hardly distinguishable from those obtained when evaluating the steady-state schedule; see e.g. the examples in Section 5 on transient schedules. This fact can be used by managers: the steady-state schedules depicted in Fig. 5 can then serve as some sort of 'cookbook' to determine the optimal interarrival times in the specific situation they encounter. A pragmatic view is that one could use the equidistant schedule as resulting from a steady-state analysis, which is in particular cases already close to optimal, and that one further improves it by slightly modifying the schedule at the start and end of the schedule (so as to obtain a dome-shape pattern, similar to the ones found in the section on transient schedules).

7. Conclusion and discussion

In this paper we have considered the problem of finding appointment schedules that balance the clients' mean waiting times and the server's idle times. We have extended the approach that was developed in Kuiper et al. [12] for the single-node queue to its tandem counterpart. A key step in our procedure is that we approximate the service times by appropriately chosen phase-type random variables. Importantly, phase-type distributions allow for (relatively) easy calculations; in particular, the sojourn-time distributions of the individual clients can be determined recursively. Furthermore, we show how to efficiently compute the steady-state sojourn-time distribution. Having the sojourn-time distribution at our disposal, optimization techniques can be used to determine optimal schedules. We note that it was shown in Kuiper et al. [12] that replacing service-time distributions (Weibull and lognormal) by their phase-type counterparts (of low dimension) hardly affects the optimal schedule.

The experiments in Sections 5 and 6 (for schedules in a transient environment and in stationarity, respectively) give insight into the behavior of the optimal schedules under a broad variety of parameter settings (corresponding to the weights between both servers, and the mean and SCV of each server).

There are several directions for further research. (i) In the first place, one could study the optimal schedules in alternative multi-

node settings, such as the fork-join queue. In addition, it would be interesting to systematically assess the impact of the risk function on the optimal schedule; in this paper all results are based on a specific risk function (the linear one, that is). One could also investigate the *sequential* approach (as proposed in Kemper et al. [11] for the single node), which assigns optimal arrival times sequentially to individual users (i.e., the schedule gradually fills). (ii) In the second place, the optimal schedule can be studied in situations in which additional features play a role, such as 'urgent clients' (whose arrivals correspond to a Poisson process, or more realistically by the processes identified by Alexopoulos et al. [1]), different types of clients (each type being characterized by its own service-time distribution), and no-shows; it is observed that the latter can be easily incorporated in the phase-type representation $(\boldsymbol{\alpha}, \boldsymbol{S})$ by adapting the initial probability vector $\boldsymbol{\alpha}$.

It is also remarked that in the heterogeneous scenario, there is the issue of identifying the *order* of the clients that minimizes the objective function. We have observed that the 'variation' at the first server (expressed in terms of SCV_1) propagates to the second server. As this variation results in higher idle and waiting times, it is anticipated that SCV_1 has a crucial impact on the objective function. This suggests the heuristic, for situations in which the SCV_2s are roughly equal, to schedule clients in ascending order of their SCV_1s. Unfortunately, even for the single-node system there are hardly rigorous results for such properties yet, notable exceptions being Kemper et al. [11] (focusing on the sequential approach mentioned above), Rohleder and Klassen [17] (focusing on simulation studies where clients with low variance are scheduled first) and Mak et al. [15] (focusing on a 'distribution-free' setup, minimizing the worst-case expected waiting and overtime over all probability distributions with given moments).

References

[1] Alexopoulos C, Goldsman D, Fontanesi J, Kopald D, Wilson JR. Modeling patient arrivals in community clinics. Omega 2008;36(1):33–43.

[2] Asmussen S. Applied probability and queues, 2nd ed. In: Vol. 51 of applications of mathematics. New York, NY, USA: Springer-Verlag; 2003 (stochastic modelling and applied probability).

[3] Bailey N. A study of queues and appointment systems in hospital out-patient departments, with special reference to waiting-times. Journal of the Royal Statistical Society. Series B (Methodological) 1952;14(2):185–99.

[4] Cayirli T, Veral E. Outpatient scheduling in health care: a review of literature. Production and Operations Management 2003;12(4):519–49.

[5] Côté MJ, Stein WE. A stochastic model for a visit to the doctors office. Mathematical and Computer Modelling 2007;45(34):309–23.

[6] Cox TF, Birchall JP, Wong H. Optimising the queuing system for an ear, nose and throat outpatient clinic. Journal of Applied Statistics 1985;12(2):113–26.

[7] Dallery Y, Gershwin SB. Manufacturing flow line systems: a review of models and analytical results. Queueing Systems 1992;12(December (1–2)):3–94.

[8] De Vuyst, S, Bruneel, H, Fiems, D. Fast evaluation of appointment schedules for outpatients in health care. In: Proceedings of ASMTA; 2011. pp. 113–131.

[9] Hassin R, Mendel S. Scheduling arrivals to queues: a single-server model with no-shows. Management Science 2008;54(3):565–72.

[10] Kaandorp G, Koole G. Optimal outpatient appointment scheduling. Health Care Management Science 2007;10(3):217–29.

[11] Kemper B, Klaassen C, Mandjes M. Optimized appointment scheduling. European Journal of Operational Research 2014;239(1):243–55.

[12] Kuiper A, Kemper B, Mandjes M. A computational approach to optimized appointment scheduling. Queueing Systems 2015;79(1):5–36.

[13] Lau H, Lau A. A fast procedure for computing the total system cost of an appointment schedule for medical and kindred facilities. IIE Transactions 2007;32(9):833–9.

[14] Luo J, Kulkarni V, Ziya S. Appointment scheduling under patient no-shows and service interruptions. Manufacturing and Service Operations Management 2012;14(4):670–84.

[15] Mak H, Rong Y, Zhang J. Appointment scheduling with limited distributional information. Management Science 2015;61(2):316–34.

[16] Rising EJ, Baron R, Averill B. A systems analysis of a university-health-service outpatient clinic. Operations Research 1973;21(5):1030–47.

[17] Rohleder TR, Klassen KJ. Using client-variance information to improve dynamic appointment scheduling performance. Omega 2000;28(3):293–302.

[18] Swisher JR, Jacobson SH, Jun J, Balci O. Modeling and analyzing a physician clinic environment using discrete-event (visual) simulation. Computers & Operations Research 2001;28(2):105–25.

[19] Tijms H. Stochastic modelling and analysis—a computational approach. Wiley series in probability and mathematical statistics: applied probability and statistics. Chichester, UK: John Wiley & Sons; 1986.

[20] Wang P. Static and dynamic scheduling of customer arrivals to a single-server system. Naval Research Logistics 1993;40(3):345–60.

[21] Wang P. Optimally scheduling in customer arrival times for a single-server system. Computers & Operations Research 1997;24(8):703–16.

[22] Weiss E. Models for determining estimated start times and case orderings in hospital operating rooms. IIE Transactions 1990;22(2):143–50.

[23] Welch JD. Appointment systems in hospital outpatient departments. Operations Research 1964;15(3):224–32.