Ann Oper Res (2012) 198:57–82
DOI 10.1007/s10479-011-0846-y

Time-limited polling systems with batch arrivals and phase-type service times

Ahmad Al Hanbali · Roland de Haan · Richard J. Boucherie · Jan-Kees van Ommeren

Published online: 27 January 2011

Abstract In this paper, we develop a general framework to analyze polling systems with either the autonomous-server or the time-limited service discipline. According to the autonomous-server discipline, the server continues servicing a queue for a certain period of time. According to the time-limited service discipline, the server continues servicing a queue for a certain period of time or until the queue becomes empty, whichever occurs first. We consider Poisson batch arrivals and phase-type service times. It is known that these disciplines do not satisfy the well-known branching property in polling systems. Therefore, hardly any exact results exist in the literature. Our strategy is to apply an iterative scheme that is based on relating in closed-form the joint queue-lengths at the beginning and the end of a server visit to a queue. These kernel relations are derived using the theory of absorbing Markov chains.

Keywords Absorbing Markov chains · Matrix analytic solution · Polling system · Autonomous server discipline · Time limited discipline · Poisson batch arrivals · Phase-type service times · Iterative scheme · Performance analysis

1 Introduction

Polling systems have been extensively studied in the last years due to their vast area of applications in production and telecommunication systems (Levy and Sidi 1990; Takagi 2000). They offer an adequate modeling framework to analyze systems in which a set of

A. Al Hanbali · R. de Haan · R.J. Boucherie · J.-K. van Ommeren (✉)
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
e-mail: J.C.W.vanOmmeren@utwente.nl

A. Al Hanbali
e-mail: a.alhanbali@utwente.nl

R. de Haan
e-mail: R.deHaan@utwente.nl

R.J. Boucherie
e-mail: R.J.Boucherie@utwente.nl

entities need certain service from a single resource. These entities are located at different positions in the system awaiting their turn to receive service.

In queueing theory, a polling system is equivalent to a set of queues with exogenous job arrivals all requiring service from a single server. The server serves each queue according to a specific service discipline and after serving a queue he will move to a next queue. A tractable analysis of a polling system is possible if the system satisfies the so-called branching property (Resing 1993). This property states that each job present at a queue at the arrival instant of the server will be replaced in an independent and identically distributed manner by a random number of jobs during the course of the server's visit. For disciplines not satisfying this property hardly any exact results are known.

The two most well-known disciplines that satisfy the branching property are the exhaustive and gated discipline. Exhaustive means that the server continues servicing a queue until it becomes empty. At this instant the server moves to the next queue in his schedule. Gated means that the server only serves the jobs present in the queue upon its arrival.

The drawback of the exhaustive and gated disciplines is that the server is controlled by the presence of jobs in the queues. To reduce this control on the server, other types of service disciplines were introduced such as the time-limited or the k-limited discipline. According to the time-limited discipline, the server continues servicing a queue for a certain time period or until the queue becomes empty, whichever occurs first. Under the k-limited discipline, the server continues servicing a queue until k jobs are served or the queue becomes empty, whichever occurs first. Another discipline, evaluated more recently in the literature and closely related to the time-limited discipline, is the so-called autonomous-server discipline (Al Hanbali et al. 2008a; de Haan et al. 2009), where the server stays at a queue for a certain period of time, even if the queue becomes empty. This discipline may also be seen as the non-exhaustive time-limited discipline. We should emphasize that these latter disciplines do not satisfy the branching property and thus hardly any closed-form results are known for the queue-length distribution under these disciplines.

To circumvent this difficulty, researchers resort to numerical methods using for instance iterative solution techniques or the power series algorithm. The power series algorithm (Blanc 1992a, 1992b, 1998) aims at solving the global balance equations. To this end, the state probabilities are written as a power series and via a complex computation scheme the coefficients of these series, and thus the queue-length probabilities, are obtained. The iterative techniques (Leung 1991, 1994) exploit the relations between the joint queue-length distributions at specific instants, viz., the start of a server visit and the end of a server visit. The relation between the queue-length at the start and end of a visit to a queue is established via recursively expressing the queue-length at a job departure instant in terms of the queue-length at the previous departure instant of a job. The complementary relation, between the queue-length at the end of a visit to a queue and a start of a visit to a next queue, can easily be established via the switch-over time. Starting with an initial distribution, the stationary queue-length distribution is then obtained by means of iteration. For the autonomous server discipline, the authors in de Haan et al. (2009) followed a similar iterative technique to those in Leung (1991, 1994). For the k-limited discipline, the authors in van Vuuren and Winands (2007) proposed an iterative approximation that is based on a matrix geometric method. Although these methods offer a way to numerically solve intrinsically hard systems, their solution provides little fundamental insight. Recently, the author in Van Houdt (2010) proposed a numerical solution for the discrete-time Bernoulli polling systems that is based on the iterative power method. The Bernoulli service discipline includes as a particular case the exhaustive and k-limited discipline but not the time-limited discipline. In Sect. 7 we shall show that the performance of the algorithm in Van Houdt (2010) when it is applied to the exhaustive polling system is comparable to our numerical scheme.

Under the assumption of exponential service times, we derived in Al Hanbali et al. (2008b) a direct and more insightful relation between the joint number of jobs at the beginning and end of a server visit to a queue for the autonomous-server, the time-limited, and the k-limited discipline. This is done using a matrix analytic approach. In the same paper, we also re-derived a result of Yechiali and Eliazar (1998) for the exhaustive time-limited discipline for the special case of exponential service times. The latter article studied the exhaustive time-limited discipline with the preemptive service. Observing that upon successful service completion at a queue the busy period in fact regenerates, the authors could obtain a closed-form relation between the joint queue-lengths at the end and the beginning of a server visit. In de Haan (2009, Chap. 5) all these results were extended by including routing of jobs between the different queues. This is done by constructing Markov chains at specific embedded epochs and subsequently relating the states at these epochs.

In this paper, we develop a framework to analyze the autonomous server and the time-limited polling systems with Poisson batch arrivals and phase-type service times. Our framework incorporates an iterative solution method which enhances the method introduced in Leung (1991, 1994) and more recently in de Haan et al. (2009). More specifically, contrary to that approach, we will establish a direct relation between the joint number of jobs at the beginning and end of a server visit to a queue without conditioning on any intermediate events that occur during a visit. To this end, we use the theory of absorbing Markov chains (AMC) (Grinstead and Snell 1997; Neuts 1981). We construct an AMC whose transient states represent the states of the polling system. The event of the server leaving a queue is modeled as an absorbing event. We will set the initial state of the AMC to the joint number of jobs at the beginning of a service period of a queue. Therefore, to find the joint number of jobs at the end of a service period, it is sufficient to keep track of the state from which the transition to the absorption state occurs. The probability of the latter event is eventually determined by first ordering the states in a careful way and consequently exploiting the structures that arise in the generator matrix of the AMC. Following this approach, we relate in closed-form the joint queue-length probability generating functions (p.g.f.) at the end of a visit period to a queue to the joint queue-length p.g.f. at the beginning of this visit period. The major part of this paper is devoted to deriving these kernel relations for the above-mentioned two disciplines: autonomous-server and time-limited.

Once the kernel relations are obtained, the joint queue-length distribution at the server departure instants is readily obtained via a numerical iterative scheme. In few words, the numerical scheme works as follows. We start with an empty system. Second, we use the kernel relations to numerically compute the joint queue-length generating function at the server departure instant from a queue, say queue 1. Third, we numerically compute from the last generating function the joint queue-length generating function at the beginning of the server visit to queue 2 based on the Laplace-Stieltjes transform of the switch-over time. Then, we repeat the second and the third step for queue 2, then 3, etc. Whenever the queue index exceeds the number of queues in the system we re-initialize it to 1, and we say that the scheme has completed one computation cycle. We repeat the computation cycle multiple times until the system converges within a predefined numerical precision. When the convergence occurs, our scheme yields the joint queue-length distribution at the server departure instants from the queues. We numerically investigate the computational costs of the scheme for different parameters settings. See Sects. 5 and 6 for more details.

Although we have developed our framework for the case of autonomous-server and time-limited systems, our framework is generally applicable to analyze other branching and non-branching type polling systems. The key step is the correct ordering of the states that allows us to invoke the theory of absorbing Markov chains in order to relate in closed-form the joint number of jobs in the system at the beginning and end of a server visit to a queue.

The paper is organized as follows. In Sect. 2 we give a detailed description of the model and the assumptions. Section 3 analyzes the autonomous-server discipline. In Sect. 4 we study the time-limited discipline. In Sect. 5 we describe the iterative scheme that is important to compute the joint queue-length distribution. Section 6 focuses on the scheme computation cost as function of the system parameters. In Sect. 7 we compare the computation cost of our scheme with other existing algorithm. Section 8 discusses some possible extensions of the scheme. Finally, in Sect. 9, we conclude the paper and give some research directions.

2 Model

We consider a single-server polling model consisting of M first-in-first-out (FIFO) queues with unlimited queue size. We refer to the ith queue as Q_i, $i=1,\dots,M$. Jobs arrive to Q_i in batches according to a Poisson process of rate λ_i. The sequence of batch sizes consists of independent and identically distributed random variables, which are independent of inter-arrival times. Let us denote by D_i the batch size at Q_i with probability mass function $D_i(\cdot)$ and probability generating function $\hat{D}_i(z)$, $|z|\le 1$. We assume that $D_i \ge 1$ for $i=1,\dots,M$. The service time of a job at Q_i is denoted by B_i. B_i is a phase-type random variable with distribution function $B_i(\cdot)$ with mean b_i and h_i phases. That is, B_i is a mixture of h_i exponential random variables. We assume that the service times are independent and identically distributed random variables and they are independent of the batch size and inter-arrival time.

A phase-type distribution can be represented by an initial distribution vector π, a transient generator $\mathbf{T}$, and an absorption rate vector T^o, i.e., $\mathbf{T}^{-1}T^o = -e^T$, where e^T is a column vector with all entries equal to one. For more details we refer, e.g., to Neuts (1981, p. 44). Then, it is well-known that the Laplace-Stieltjes transform (LST) B_i, the service times at Q_i, can be written as

$$\tilde{B}_i(s) = \pi_i(s\mathbf{I}-\mathbf{T}_i)^{-1}T_i^o, \quad \mathrm{Re}(s)\ge 0. \tag{1}$$

For later use, we need to introduce the LST of the residual (phase-type) service times.

Lemma 1 *The LST of the residual service times at Q_i is given by*

$$\tilde{B}_i^*(s) = \frac{1}{b_i}\pi_i(s\mathbf{I}-\mathbf{T}_i)^{-1}e^T, \qquad \mathrm{Re}(s)\ge 0. \tag{2}$$

Proof The LST of the residual service times reads

$$\tilde{B}_i^*(s) = \frac{1}{b_i s}(1-\tilde{B}_i(s)) = -\frac{1}{b_i}\pi_i\mathbf{T}_i^{-1}(s\mathbf{T}_i^{-1}-\mathbf{I})^{-1}\mathbf{T}_i^{-1}T_i^o = \frac{1}{b_i}\pi_i(s\mathbf{I}-\mathbf{T}_i)^{-1}e^T. \qquad \square$$

We let $N_i(t)$ denote the number of jobs in Q_i, $i=1,\dots,M$, at time $t\ge 0$ and it is assumed that $N_i(0)=0$, $i=1,\dots,M$. The server visits the queues in a cyclic fashion. After a visit to Q_i, the server incurs a switch-over time C^i from Q_i to Q_{i+1}. We assume that C^i is independent of the service requirement and follows a general distribution $C^i(\cdot)$ with mean c^i, where at least one $c^i > 0$. The service discipline at each queue is either autonomous-server or time-limited. Under the autonomous-server discipline, the server remains at location Q_i an exponentially distributed time with rate α_i before it migrates to the next queue in the cycle. Under the time-limited discipline, the server departs from Q_i when it becomes empty or when a timer of exponentially distributed duration with rate α_i has expired, whichever occurs first.

In case the server is active at the end of a server visit, which may happen under the autonomous-server and time-limited disciplines, then the service will be preempted. At the beginning of the next visit of the server, the service time will be re-sampled according to $B_i(\cdot)$. This discipline is commonly referred to as *preemptive-repeat-random*.

It is assumed that the queues of the polling system are stable. In the following lemmas we shall state the stability condition for both the autonomous-server and the time-limited systems. The proofs of these lemmas are straightforward extensions to those of Theorems 3.1 and 3.2 in de Haan (2009). We should note that the stability proof in de Haan (2009) relied largely on the stability proof of Fricker and Jaibi (1994) for a class of polling systems with non-preemptive and work-conserving service disciplines.

Lemma 2 (Autonomous-server discipline)

$$\textit{System is stable} \quad \Longleftrightarrow \quad \rho_i < \kappa_i, \quad i = 1, \dots, M,$$

where

$$\rho_i = \lambda_i \mathbb{E}[D_i] \cdot \frac{1 - \tilde{B}_i(\alpha_i)}{\alpha_i \tilde{B}_i(\alpha_i)}, \qquad \kappa_i = \frac{1/\alpha_i}{c_t + \sum_{j=1}^{M} 1/\alpha_j}, \quad c_t = \sum_{j=1}^{M} c_j.$$

We note that $(1 - \tilde{B}_i(\alpha_i))/(\alpha_i \tilde{B}_i(\alpha_i))$ is the expected value of the *effective service time* of a job in Q_i which includes the work lost due to service preemptions. κ_i is the availability fraction of the server at Q_i.

Lemma 3 (Time-limited discipline)

$$\textit{System is stable} \quad \Longleftrightarrow \quad \rho + \max_{i=1,\dots,M} \left(\frac{\lambda_i \mathbb{E}[D_i]}{\mathbb{E}[G_i^*]} \right) \cdot c_t < 1,$$

where

$$\rho = \sum_{j=1}^{M} \frac{\lambda_i \mathbb{E}[D_i](1 - \tilde{B}_i(\alpha_i))}{\alpha_i \tilde{B}_i(\alpha_i)}, \qquad \mathbb{E}[G_i^*] = \frac{\tilde{B}_i(\alpha_i)}{1 - \tilde{B}_i(\alpha_i)}.$$

We note that ρ represents the total offered load to the system and $\mathbb{E}[G_i^*]$ the mean number of served jobs at Q_i during a cycle when Q_i is saturated.

A word on notation. Given a random variable X, $X(t)$ will denote its distribution function. We use $\mathbf{I}$ to denote an identity matrix of an appropriate size and use $\otimes$ as the Kronecker product operator defined as follows. Let $\mathbf{A}$ and $\mathbf{B}$ be two matrices and $a(i, j)$ and $b(i, j)$ denote the (i, j)-entries of $\mathbf{A}$ and $\mathbf{B}$ respectively then $\mathbf{A} \otimes \mathbf{B}$ is a block matrix where the (i, j)-block is equal to $a(i, j)\mathbf{B}$. We use e to denote a row vector of appropriate size with entries equal to one and e_i to denote a row vector of appropriate size with the ith entry equal to one and the other elements equal to zero. Finally, v^T will denote the transpose of vector v.

3 Autonomous-server discipline

In this section, we will relate the joint queue-length probabilities at the beginning and end of a server visit to a queue for the autonomous-server discipline. Under the autonomous-server discipline, the server remains at location Q_i for an exponentially distributed time with rate α_i before it migrates to the next queue in the cycle. It is stressed that even when Q_i becomes empty, the server will remain at this queue.

Without loss of generality let us consider a server visit to Q_1. The number of jobs at the various queues at the beginning of a server visit to Q_1 is denoted by $\mathbf{N}_1^b := (N_{11}^b, \ldots, N_{M1}^b)$; let $\mathbf{N}_1^e := (N_{11}^e, \ldots, N_{M1}^e)$ denote the queue-lengths at the end of such a visit. We assume that the p.g.f. of the steady-state queue-length at the beginning of a server visit to Q_1, denoted by $\beta_1^A(\mathbf{z}) = \mathbb{E}[\mathbf{z}^{\mathbf{N}_1^b}]$, is known, where $\mathbf{z} := (z_1, \ldots, z_M)$ and $|z_i| \le 1$ for $i = 1, \ldots, M$. The aim is to derive the p.g.f. of the steady-state queue-length at the end of the server visit to Q_1, denoted by $\gamma_1^A(\mathbf{z}) = \mathbb{E}[\mathbf{z}^{\mathbf{N}_1^e}]$.

Let $\mathbf{N}(t) := (PH_1(t), N_1(t), \ldots, N_M(t))$ denote the $(M+1)$-dimensional, continuous-time Markov chain with discrete state-space $\xi_A = \{0, 1, \ldots, h_1\} \times \{0, 1, \ldots\}^M \cup \{a\}$, where $N_m(t)$, $m = 1, \ldots, M$, represents the number of jobs in Q_m and $PH_1(t)$ the phase of the job in service at Q_1 at time t. State $\{a\}$ is absorbing. We refer to this absorbing Markov chain by $\mathbf{AMC}_A$. The absorption of $\mathbf{AMC}_A$ occurs when the server leaves Q_1 which happens with rate α_1. Moreover, the initial state of $\mathbf{AMC}_A$ at $t = 0$ is set to the system state at the server's arrival to Q_1, i.e., $\mathbf{N}_1^b = (i_1, \ldots, i_M)$. Therefore, the probability that the absorption of $\mathbf{AMC}_A$ occurs from state $(j_1, \ldots, j_M)$ equals $\mathbb{P}(\mathbf{N}_1^e = (j_1, \ldots, j_M) | \mathbf{N}_1^b = (i_1, \ldots, i_M))$.

We derive now $\mathbb{P}(\mathbf{N}_1^e = (j_1, \ldots, j_M) | \mathbf{N}_1^b = (i_1, \ldots, i_M))$. During a server visit to Q_1, the number of jobs at Q_m, $m = 2, \ldots, M$, may only increase. Therefore, $\mathbb{P}(\mathbf{N}_1^e = (j_1, \ldots, j_M) | \mathbf{N}_1^b = (i_1, \ldots, i_M)) = 0$ for $j_l < i_l$, $l = 2, \ldots, M$. For sake of clarity, we shall first show in detail the structure of $\mathbf{AMC}_A$ in the case of 3 queues, i.e. for $M = 3$, and the procedure of the proof of the desired result before considering the general case.

Case $M = 3$ Let us consider the transient states of $\mathbf{AMC}_A$, i.e., $(ph_1, n_1, n_2, n_3) \in \xi_A \setminus \{a\}$. We recall that we consider a server visit to Q_1. The number of jobs at Q_2 and Q_3 may only increase during a server visit to Q_1, while the number of jobs at Q_1 may increase or decrease. To take advantage of this property, we will order the transient states of the $\mathbf{AMC}_A$ as follows: $(0,0,0,0), (1,0,0,0), \ldots, (0,1,0,0), (1,1,0,0), \ldots, (0,0,1,0), (1,0,1,0), \ldots, (0,0,0,1), (1,0,0,1), \ldots$, i.e., lexicographically ordered first according to n_3, then n_2, n_1, and finally according to ph_1. This ordering induces that the generator matrix of the transitions between the transient states of $\mathbf{AMC}_A$ for $M = 3$, denoted by $\mathbf{Q}_3$, is an infinite upper-triangular block matrix with diagonal blocks equal to $\mathbf{A}_3$ and ith upper-diagonal blocks equal to $\lambda_3 D_3(i)\mathbf{I}$, i.e.,

$$\mathbf{Q}_3 = \begin{pmatrix} \mathbf{A}_3 & \lambda_3 D_3(1)\mathbf{I} & \lambda_3 D_3(2)\mathbf{I} & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{A}_3 & \lambda_3 D_3(1)\mathbf{I} & \lambda_3 D_3(2)\mathbf{I} & \cdots & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}. \tag{3}$$

We note that $\mathbf{A}_3$ denotes the generator matrix of the transitions which do not induce any modification in the number of jobs at Q_3. Moreover, $\lambda_3 D_3(i)\mathbf{I}$ denotes the transition rate matrix between the transient states (ph_1, n_1, n_2, n_3) and $(ph_1, n_1, n_2, n_3 + i)$, i.e., the transitions that represent an arrival of a batch of size i to Q_3. The block matrix $\mathbf{A}_3$ is also an infinite upper-triangular block matrix with diagonal blocks equal to $\mathbf{A}_2$, and ith upper-diagonal blocks equal $\lambda_2 D_2(i)\mathbf{I}$, i.e.,

$$\mathbf{A}_3 = \begin{pmatrix} \mathbf{A}_2 & \lambda_2 D_2(1)\mathbf{I} & \lambda_2 D_2(2)\mathbf{I} & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{A}_2 & \lambda_2 D_2(1)\mathbf{I} & \lambda_2 D_2(2)\mathbf{I} & \cdots & \cdots \\ \vdots\ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}, \tag{4}$$

where $\lambda_2 D_2(i)\mathbf{I}$ denotes the transition rate matrix between the states (ph_1, n_1, n_2, n_3) and $(ph_1, n_1, n_2 + i, n_3)$. $\mathbf{A}_2$ is the generator matrix of the transition between the states (ph_1, n_1, n_2, n_3) and (l, k, n_2, n_3) with $k \geq \max(n_1 - 1, 0)$ and $l \leq h_1$, the total number of phases in the service times. Observe that $\mathbf{A}_2$ equals the sum of the matrix $-(\lambda_2 + \lambda_3 + \alpha_1)\mathbf{I}$ and the generator matrix of an $M^X/PH/1$ queue with Poisson batch arrivals and phase-type service times. Let $\mathbf{A}_1$ denote the generator of an $M^X/PH/1$. It is readily seen that (see, e.g., Neuts 1981, Chap. 3, Sect. 2)

$$\mathbf{A}_1 = \begin{pmatrix} -\lambda_1 & \lambda_1 D_1(1)\pi_1 & \lambda_1 D_1(2)\pi_1 & \cdots & \cdots & \cdots \\ T_1^o & \mathbf{T}_1 - \lambda_1 \mathbf{I} & \lambda_1 D_1(1)\mathbf{I} & \lambda_1 D_1(2)\mathbf{I} & \cdots & \cdots \\ \mathbf{0} & T_1^o \pi_1 & \mathbf{T}_1 - \lambda_1 \mathbf{I} & \lambda_1 D_1(1)\mathbf{I} & \lambda_1 D_1(2)\mathbf{I} & \cdots \\ \vdots\ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}. \tag{5}$$

We recall that T_1^o is a column vector and π_1 is a row vector thus $T_1^o \pi_1$ is a matrix of rank one with (i, j)-entry representing the transition rate from state (i, n_1, n_2, n_3) to $(j, n_1 - 1, n_2, n_3)$.

Now, we compute $\mathbb{P}(\mathbf{N}_1^e = (j_1, j_2, j_3)|\mathbf{N}_1^b = (i_1, i_2, i_3))$ as function of the inverse of $\mathbf{Q}_3$, $\mathbf{A}_3$ and $\mathbf{A}_2$ and later on we shall uncondition on N_{13}^e, then on N_{12}^e, and finally on N_{11}^e. We emphasize that since $\mathbf{Q}_3$, $\mathbf{A}_3$ and $\mathbf{A}_2$ are all sub-generators with the sum of their row elements strictly negative, these matrices are invertible. It shall become clear that in this paper we do not need to determine these inverse matrices in closed-form. For convenience, we abbreviate the condition $\mathbf{N}_1^b = (i_1, i_2, i_3)$ to $\mathbf{N}_1^b$, e.g., $\mathbb{P}(\mathbf{N}_1^e = (j_1, j_2, j_3)|\mathbf{N}_1^b)$ denotes $\mathbb{P}(\mathbf{N}_1^e = (j_1, j_2, j_3)|\mathbf{N}_1^b = (i_1, i_2, i_3))$.

From the theory of absorbing Markov chains, given that $\mathbf{AMC}_A$ starts in state $\mathbf{N}_1^b = (i_1, i_2, i_3)$, the probability that the transition to the absorption state $\{a\}$ occurs from state (j_1, j_2, j_3) reads (see, e.g., Gaver et al. 1984)

$$\mathbb{P}\big(\mathbf{N}_1^e = (j_1, j_2, j_3)|\mathbf{N}_1^b\big) = -\alpha_1 c_3 (\mathbf{Q}_3)^{-1} d_3, \tag{6}$$

where c_3 is the probability distribution vector of $\mathbf{AMC}_A$'s initial state which is given by

$$c_3 := e_{i_3} \otimes e_{i_2} \otimes e_{i_1} \otimes \pi_1,$$

and $\alpha_1 d_3$ is the transition rate vector to $\{a\}$ given that (j_1, j_2, j_3) is the last state visited before absorption with

$$d_3 := e_{j_3} \otimes e_{j_2} \otimes e_{j_1} \otimes e.$$

Note that the presence of π_1 in c_3 is due to the preemptive-repeat discipline, and e in d_3 is due to the un-conditioning on the phase of the service times in Q_1 when the server leaves the queue. By analogy with Guillemin and Simonian (1995), the absorption probability was applied on infinite state space absorbing Markov chains.

For later use, let us define the following row vectors:

$$c_2 := e_{i_2} \otimes e_{i_1} \otimes \pi_1, \qquad d_2 := e_{j_2} \otimes e_{j_1} \otimes e,$$
$$c_1 := e_{i_1} \otimes \pi_1, \qquad d_1 := e_{j_1} \otimes e.$$

We are now ready to formulate our first result.

Lemma 4 *The conditional generating function of the queue-length of Q_3 at the end of the server visit to Q_1 is given by*

$$\mathbb{E}\big[z_3^{N_{31}^e}\mathbf{1}_{\{N_{11}^e=j_1,N_{21}^e=j_2)\}}|\mathbf{N}_1^b\big]=-\alpha_1 z_3^{i_3}c_2\big(\lambda_3\hat{D}_3(z_3)\mathbf{I}+\mathbf{A_3}\big)^{-1}d_2^T. \tag{7}$$

Proof Multiplying (6) by $z_3^{j_3}$ and summing these equations over j_3 we find that

$$\begin{aligned}\mathbb{E}\big[z_3^{N_{31}^e}\mathbf{1}_{\{N_{11}^e=j_1,N_{21}^e=j_2)\}}|\mathbf{N}_1^b\big]&=-\alpha_1 c_3(\mathbf{Q}_3)^{-1}\sum_{j_3\geq i_3}z_3^{j_3}(e_{j_3}\otimes d_2)^T\\&=-\alpha_1 c_3(\mathbf{Q}_3)^{-1}\bigg(\sum_{j_3\geq i_3}z_3^{j_3}e_{j_3}\otimes d_2\bigg)^T\\&=-\alpha_1\bigg(\sum_{j_3\geq i_3}z_3^{j_3}u_3(j_3)\bigg)d_2^T,\end{aligned} \tag{8}$$

where $\mathbf{u}_3=(u_3(0),u_3(1),\ldots):=c_3(\mathbf{Q}_3)^{-1}$. First, let us derive $\sum_{j_3\geq i_3}z_3^{j_3}u_3(j_3)$. Note that $\mathbf{u}_3\mathbf{Q}_3=c_3$. Inserting $\mathbf{Q}_3$ given in (3) into the latter equation gives that

$$u_3(0)\mathbf{A_3}=\mathbf{0}, \tag{9}$$

$$\lambda_3\sum_{l=0}^{n-1}D_3(n-l)u_3(l)\mathbf{I}+u_3(n)\mathbf{A_3}=\mathbf{1}_{\{n=i_3\}}c_2,\quad n\geq 1. \tag{10}$$

Note, since $\mathbf{A_3}$ is nonsingular, (9) yields that $u_3(0)=\mathbf{0}$, i.e., $u_3(0)$ is a vector of zeros. Inserting $u_3(0)=\mathbf{0}$ into (10) with $n=1$ yields that $u_3(1)=\mathbf{0}$. Therefore, we deduce by an induction argument that $u_3(n)=\mathbf{0}$ for $n=0,\ldots,i_3-1$. The latter system of equations now rewrites

$$u_3(i_3)\mathbf{A_3}=c_2, \tag{11}$$

$$\lambda_3\sum_{l=i_3}^{n-1}D_3(n-l)u_3(l)+u_3(n)\mathbf{A_3}=\mathbf{0},\quad n>i_3. \tag{12}$$

Multiplying (11) by $z_3^{i_3}$ and (12) by z_3^n and summing these equations over n we find that

$$\sum_{j_3\geq i_3}z_3^{j_3}u_3(j_3)=z_3^{i_3}c_2\big(\lambda_3\hat{D}_3(z_3)\mathbf{I}+\mathbf{A_3}\big)^{-1}. \tag{13}$$

Inserting (13) into (8) readily gives Lemma 4. □

Lemma 5 *The conditional generating function of the joint queue-length of Q_2 and Q_3 at the end of the server visit to Q_1 is given by*

$$\mathbb{E}\big[z_2^{N_{21}^e}z_3^{N_{31}^e}\mathbf{1}_{\{N_{11}^e=j_1\}}|\mathbf{N}_1^b\big]=-\alpha_1 z_2^{i_2}z_3^{i_3}c_1\big(\lambda_2\hat{D}_2(z_2)\mathbf{I}+\lambda_3\hat{D}_3(z_3)\mathbf{I}+\mathbf{A_2}\big)^{-1}d_1^T. \tag{14}$$

Proof Multiplying (7) by $z_2^{j_2}$ and summing over j_2 gives that

$$\begin{aligned}\mathbb{E}\big[z_2^{N_{21}^e} z_3^{N_{31}^e} \mathbf{1}_{\{N_{11}^e = j_1\}} | \mathbf{N}_1^b\big] &= -\alpha_1 z_3^{i_3} c_2 \big(\lambda_3 \hat{D}_3(z_3)\mathbf{I} + \mathbf{A_3}\big)^{-1} \Bigg(\sum_{j_2 \geq i_2} z_2^{j_2} e_{j_2} \otimes d_1\Bigg)^T \\ &= -\alpha_1 z_3^{i_3} \Bigg(\sum_{j_2 \geq i_2} z_2^{j_2} u_2(j_2)\Bigg) d_1^T, \end{aligned} \tag{15}$$

where $\mathbf{u}_2 = (u_2(0), u_2(1), \ldots) := c_2(\lambda_3 \hat{D}_3(z_3)\mathbf{I} + \mathbf{A_3})^{-1}$. We emphasize that the matrices $\mathbf{Q_3}$ and $(\lambda_3 \hat{D}_3(z_3)\mathbf{I} + \mathbf{A_3})$ given in (3) and (4) have a similar structure. Therefore, by analogy with the derivation of (8) in Lemma 4 we deduce that

$$\sum_{j_2 \geq i_2} z_2^{j_2} u_2(j_2) = z_2^{i_2} c_1 \big(\lambda_2 \hat{D}_2(z_2)\mathbf{I} + \lambda_3 \hat{D}_3(z_3)\mathbf{I} + \mathbf{A_2}\big)^{-1}. \tag{16}$$

Inserting (16) into (15) readily gives the desired result. $\square$

We are now ready to state our main result for the autonomous-server discipline in the case $M = 3$.

Theorem 1 *The generating function of the joint queue-length of Q_1, Q_2 and Q_3 at the end of the server visit to Q_1 is given by*

$$\mathbb{E}\big[\mathbf{z}^{\mathbf{N}_1^e}\big] = p(\mathbf{z})\mathbb{E}\big[r_1(z_2, z_3)^{N_{11}^b} z_2^{N_{21}^b} z_3^{N_{31}^b}\big] + q(\mathbf{z})\mathbb{E}\big[z_1^{N_{11}^b} z_2^{N_{21}^b} z_3^{N_{31}^b}\big], \tag{17}$$

where $\mathbf{z} := (z_1, z_2, z_3)$,

$$p(\mathbf{z}) = \frac{\alpha_1}{s_1(r_1(z_2, z_3), z_2, z_3)} \times \frac{(z_1 - 1)\tilde{B}_1(s_1(z_1, z_2, z_3))}{z_1 - \tilde{B}_1(s_1(z_1, z_2, z_3))}, \tag{18}$$

$$q(\mathbf{z}) = \frac{\alpha_1}{s_1(z_1, z_2, z_3)} \times \frac{z_1(1 - \tilde{B}_1(s_1(z_1, z_2, z_3)))}{z_1 - \tilde{B}_1(s_1(z_1, z_2, z_3))}, \tag{19}$$

$s_1(z_1, z_2, z_3) = \alpha_1 + \sum_{i=1}^3 \lambda_i(1 - \hat{D}_i(z_i))$, *and where $r_1(z_2, z_3)$ is the root with smallest absolute value of*: (*solving for z_1*)

$$z_1 = \tilde{B}_1\big(s_1(z_1, z_2, z_3)\big).$$

Proof Multiplying (14) by $z_1^{j_1}$ and summing over all values of j_1 gives that

$$\begin{aligned}\mathbb{E}\big[\mathbf{z}^{\mathbf{N}_1^e} | \mathbf{N}_1^b\big] &= \mathbb{E}\big[z_1^{N_{11}^e} z_2^{N_{21}^e} z_3^{N_{31}^e} | \mathbf{N}_1^b\big] \\ &= -\alpha_1 z_2^{i_2} z_3^{i_3} c_1 \big(\lambda_2 \hat{D}_2(z_2)\mathbf{I} + \lambda_3 \hat{D}_3(z_3)\mathbf{I} + \mathbf{A_2}\big)^{-1} \Bigg(\sum_{j_1 \geq 0} z_1^{j_1} e_{j_1} \otimes e\Bigg)^T \\ &= -\alpha_1 z_2^{i_2} z_3^{i_3} \Bigg(\sum_{j_1 \geq 0} z_1^{j_1} u_1(j_1)\Bigg) e^T, \end{aligned} \tag{20}$$

where $\mathbf{u}_1 = (u_1(0), u_1(1), \ldots) := c_1(\lambda_2 \hat{D}_2(z_2)\mathbf{I} + \lambda_3 \hat{D}_3(z_3)\mathbf{I} + \mathbf{A_2})^{-1}$. Let us now derive $\sum_{j_1 \ge 0} z_1^{j_1} u_1(j_1)$. Note that $\mathbf{A}_2 = \mathbf{A}_1 - (\lambda_2 + \lambda_3 + \alpha_1)\mathbf{I}$ and $\mathbf{u}_1(\lambda_2 \hat{D}_2(z_2)\mathbf{I} + \lambda_3 \hat{D}_3(z_3)\mathbf{I} + \mathbf{A_2}) = c_1$. Inserting $\mathbf{A}_1$ given in (5) into the latter equation gives that

$$-\theta u_1(0) + u_1(1) T_1^0 = 0, \tag{21}$$

$$\lambda_1 D_1(n) u_1(0)\pi_1 + \lambda_1 \sum_{l=1}^{n-1} D_1(n-l) u_1(l)\mathbf{I}$$
$$+ u_1(n)(\mathbf{T}_1 - \theta \mathbf{I}) + u_1(n+1) T_1^0 \pi_1 = \mathbf{1}_{\{n=i_1\}}\pi_1, \quad n \ge 1, \tag{22}$$

where $\theta := \alpha_1 + \lambda_1 + \lambda_2(1 - \hat{D}_2(z_2)) + \lambda_3(1 - \hat{D}_3(z_3))$. By multiplying (21) by π_1 and adding it to the sum over n of (22) multiplied by z_1^n, we find that

$$\sum_{n \ge 1} u_1(z_1) z_1^n \left[\mathbf{T}_1 - \big(\theta - \lambda_1 \hat{D}_1(z_1)\big)\mathbf{I} + \frac{1}{z_1} T_1^0 \pi_1\right] = \left[z_1^{i_1} + u_1(0)\big(\theta - \lambda_1 \hat{D}_1(z_1)\big)\right]\pi_1. \tag{23}$$

Let $\mathbf{R} := [\mathbf{T}_1 - (\theta - \lambda_1 \hat{D}_1(z_1))\mathbf{I} + \frac{1}{z_1} T_1^0 \pi_1]$. Then,

$$\sum_{n \ge 1} u_1(z_1) z_1^n = \left[z_1^{i_1} + u_1(0)\big(\theta - \lambda_1 \hat{D}_1(z_1)\big)\right]\pi_1 \mathbf{R}^{-1}. \tag{24}$$

Inserting (24) into (20) we find that

$$\mathbb{E}\left[z_1^{N_1^e} z_2^{N_2^e} z_3^{N_3^e} | \mathbf{N}_1^b\right] = -\alpha_1 z_2^{i_2} z_3^{i_3} \left(u_1(0) + \left[z_1^{i_1} + u_1(0)\big(\theta - \lambda_1 \hat{D}_1(z_1)\big)\right]\pi_1 \mathbf{R}^{-1} e^T\right). \tag{25}$$

Now, we shall compute $\pi_1 \mathbf{R}^{-1} e$. For the ease of the notation, let us denote $\mathbf{R}_1 := \mathbf{T}_1 - (\theta - \lambda_1 \hat{D}_1(z_1))\mathbf{I}$. Therefore, $\mathbf{R} = \mathbf{R}_1 + \frac{1}{z_1} T_1^0 \pi_1$. By the Sherman-Morrison formula, see (Bernstein 2005, Fact 2.14.2, p. 67), we have that

$$\begin{aligned}
\pi_1 \mathbf{R}^{-1} e^T &= \pi_1 \left[\mathbf{R}_1^{-1} - \frac{1}{z_1 + \pi_1 \mathbf{R}_1^{-1} T_1^0} \mathbf{R}_1^{-1} T_1^0 \pi_1 \mathbf{R}_1^{-1}\right] e^T \\
&= \pi_1 \mathbf{R}_1^{-1} e^T \left[1 + \frac{\tilde{B}_1(\theta - \lambda_1 \hat{D}_1(z_1))}{z_1 - \tilde{B}_1(\theta - \lambda_1 \hat{D}_1(z_1))}\right] \\
&= -\frac{1 - \tilde{B}_1(\theta - \lambda_1 \hat{D}_1(z_1))}{\theta - \lambda_1 \hat{D}_1(z_1)} \times \frac{z_1}{z_1 - \tilde{B}_1(\theta - \lambda_1 \hat{D}_1(z_1))},
\end{aligned} \tag{26}$$

where the second equality follows from (1) and the last equality from Lemma 1. Inserting (26) into (25) yields that

$$\begin{aligned}
\mathbb{E}\left[z_1^{N_1^e} z_2^{N_2^e} z_3^{N_3^e} | \mathbf{N}_1^b\right] &= \frac{\alpha_1 z_1 z_2^{i_2} z_3^{i_3} [1 - \tilde{B}_1(s_1(z_1, z_2, z_3))][z_1^{i_1} + u_1(0) s_1(z_1, z_2, z_3)]}{s_1(z_1, z_2, z_3)[z_1 - \tilde{B}_1(s_1(z_1, z_2, z_3))]} \\
&\quad - \alpha_1 z_2^{i_2} z_3^{i_3} u_1(0),
\end{aligned} \tag{27}$$

where $s_1(z_1, z_2, z_3) = \theta - \lambda_1 \hat{D}_1(z_1)$. We shall show that for $|z_1| \le 1$ the denominator of (27) is not equal to zero except at one point. First, note that the real part of $\theta - \lambda_1 \hat{D}_1(z_1)$ is strictly positive for $\alpha_1 > 0$, $|z_i| \le 1$, $i = 1, 2, 3$. Moreover, by Rouché's theorem it is readily seen that $z_1 - \tilde{B}_1(\theta - \lambda_1 \hat{D}_1(z_1)) = 0$ has a unique root, $r_1(z_2, z_3)$, inside the unit disk. Note that $r_1(z_2, z_3)$ is function of z_2 and z_3 due to θ that is function of z_2 and z_3. Since the l.h.s. in (27) is a p.g.f., it is analytical for $|z_1| \le 1$ we deduce that $r_1(z_2, z_3)$ is a removable singularity in (27), which gives

$$u_1(0) = -\frac{r_1(z_2, z_3)^{i_1}}{\theta - \lambda_1 \hat{D}_1(r_1(z_2, z_3))}. \tag{28}$$

Inserting $u_1(0)$ into (27) and removing the condition on $\mathbf{N}_1^b$ readily gives $\mathbb{E}[\mathbf{z}^{\mathbf{N}_1^e}]$ in Theorem 1. □

General case By analogy with the case of $M = 3$, we order the transient states of $\mathbf{AMC}_A$ first according to n_M, then $n_{M-1}, \dots, n_1$, and finally according to ph_1. During a server visit to Q_1, the number of jobs at Q_j, $j = 2, \dots, M$, may only increase. Therefore, similarly to the case of $M = 3$, the generator matrix of $\mathbf{AMC}_A$ of the transition rates between the transient states of $\mathbf{AMC}_A$ for the general case, denoted by $\mathbf{Q}_M$, is an upper-triangular block matrix with diagonal blocks equal to $\mathbf{A}_M$, and ith upper-diagonal blocks equal to $\lambda_M D_M(i)\mathbf{I}$. Moreover, $\mathbf{A}_M$ in turn is an upper-triangular block matrix with diagonal blocks equal to $\mathbf{A}_{M-1}$, and ith upper-diagonal blocks equal to $\lambda_{M-1} D_{M-1}(i)\mathbf{I}$. We emphasize that $\mathbf{A}_j$, $j = M, \dots, 3$, all satisfy the previous property. Finally, the matrix $\mathbf{A}_2 = \mathbf{A}_1 - (\lambda_2 + \cdots + \lambda_M + \alpha_1)\mathbf{I}$, where $\mathbf{A}_1$ is the generator matrix of an $M^X/PH/1$ queue, with Poisson batch arrivals of inter-arrival rate λ_1 and batch size distribution function $D_1(\cdot)$.

By analogy with the $M = 3$ case, we find that the probability of $\mathbf{N}_i^e = (j_1, \dots, j_M)$, given that $\mathbf{N}_1^b = (i_1, \dots, i_M)$, reads

$$\mathbb{P}\big(\mathbf{N}_1^e = (j_1, \dots, j_M)|\mathbf{N}_1^b\big) = -\alpha_1 c_M (\mathbf{Q}_M)^{-1} d_M, \tag{29}$$

where

$$c_M := e_{i_M} \otimes \cdots \otimes e_{i_1} \otimes \pi_1, \qquad d_M := e_{j_M} \otimes \cdots \otimes e_{j_1} \otimes e.$$

Lemma 6 *The conditional generating function of the joint queue-length of $Q_2, \dots, Q_M$ at the end of the server visit to Q_1 is given by*

$$\mathbb{E}\left[\prod_{i=2}^M z_i^{N_{i1}^e} \mathbf{1}_{\{N_{11}^e = j_1\}} \middle| \mathbf{N}_1^b\right] = -\alpha_1 \left(\prod_{n=2}^M z_n^{i_n}\right) c_1 \left(\sum_{i=2}^M \lambda_i \hat{D}_i(z_i)\mathbf{I} + \mathbf{A_2}\right)^{-1} d_1^T.$$

Proof Similar to the proof of Lemma 5. □

We are now ready to present our main result for the general case.

Theorem 2 (Autonomous-server discipline) *The generating function of the joint queue-length of $Q_1, \dots, Q_M$ at the end of the server visit to Q_1 is given by*

$$\gamma_1^A(\mathbf{z}) = p_1^A(\mathbf{z})\beta_1^A(\mathbf{z}_1^*) + q_1^A(\mathbf{z})\beta_1^A(\mathbf{z}), \tag{30}$$

where $\mathbf{z} = (z_1, \dots, z_M)$, $\mathbf{z}_1^* = (r_1(z_2, \dots, z_M), z_2, \dots, z_M)$,

$$p_1^A(\mathbf{z}) = \frac{\alpha_1}{s_1(\mathbf{z}_1^*)} \times \frac{(z_1 - 1)\tilde{B}_1(s_1(\mathbf{z}))}{z_1 - \tilde{B}_1(s_1(\mathbf{z}))}, \qquad q_1^A(\mathbf{z}) = \frac{\alpha_1}{s_1(\mathbf{z})} \times \frac{z_1(1 - \tilde{B}_1(s_1(\mathbf{z})))}{z_1 - \tilde{B}_1(s_1(\mathbf{z}))},$$

$s_1(\mathbf{z}) = \alpha_1 + \sum_{i=1}^{M} \lambda_i(1 - \hat{D}_i(z_i))$, *and where* $r_1(z_2, \ldots, z_M)$ *is the root with smallest absolute value of*: (*solving for* z_1)

$$z_1 = \tilde{B}_1\bigl(s_1(\mathbf{z})\bigr).$$

Proof By analogy with the proof of Theorem 1. □

Equation (30) relates $\gamma_1^A(\mathbf{z})$, the p.g.f. of the joint queue-length at the end of a server visit to Q_1, to $\beta_1^A(\mathbf{z}_1)$, the p.g.f. of the joint queue-length at the beginning of a server visit to Q_1. From Theorem 2, we deduce that for a server visit to Q_i, $i = 1, \ldots, M$,

$$\gamma_i^A(\mathbf{z}) = p_i^A(\mathbf{z})\beta_i^A(\mathbf{z}_i^*) + q_i^A(\mathbf{z})\beta_i^A(\mathbf{z}), \tag{31}$$

where $\mathbf{z}_i^* = (z_1, \ldots, z_{i-1}, r_i(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_M), z_{i+1}, \ldots, z_M)$,

$$p_i^A(\mathbf{z}) = \frac{\alpha_i}{s_i(\mathbf{z}_i^*)} \times \frac{(z_i - 1)\tilde{B}_i(s_i(\mathbf{z}))}{z_i - \tilde{B}_i(s_i(\mathbf{z}))}, \qquad q_i^A(\mathbf{z}) = \frac{\alpha_i}{s_i(\mathbf{z})} \times \frac{z_i(1 - \tilde{B}_i(s_i(\mathbf{z})))}{z_i - \tilde{B}_i(s_i(\mathbf{z}))},$$

where $s_i(\mathbf{z}) = \alpha_i + \sum_{j=1}^{M} \lambda_j(1 - \hat{D}_j(z_j))$, and where $r_i(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_M)$ is the root with smallest absolute value of:

$$z_i = \tilde{B}_i\bigl(s_i(\mathbf{z})\bigr). \tag{32}$$

Finally, introducing the switch-over times from Q_{i-1} to Q_i, thus by using that $\mathbb{E}[\mathbf{z}^{\mathbf{N}_i^b}] = \mathbb{E}[\mathbf{z}^{\mathbf{N}_{i-1}^e}]\hat{C}^{i-1}(\mathbf{z})$, where $\hat{C}^{i-1}(\mathbf{z}) = \tilde{C}^{i-1}(\sum_{j=1}^{M} \lambda_j(1 - \hat{D}_j(z_j)))$ is the p.g.f. of the number of Poisson batch arrivals during C^{i-1}, we obtain

$$\gamma_i^A(\mathbf{z}) = p_i^A(\mathbf{z})\gamma_{i-1}^A(\mathbf{z}_i^*)\hat{C}^{i-1}(\mathbf{z}_i^*) + q_i^A(\mathbf{z})\gamma_{i-1}^A(\mathbf{z})\hat{C}^{i-1}(\mathbf{z}). \tag{33}$$

Remark 1 In the particular case where $\hat{D}_i(z_i) = z_i$, i.e., the arriving batches are all of size one, (31) agrees with de Haan (2009, Theorem 5.3).

Remark 2 The root $r_i(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)$ in (32) shall be computed numerically. Note that since the service time distribution is phase-type $r_i(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)$ becomes the root with the smallest absolute value of a polynomial function of degree equal to the total number of service phases. Note that an approximation for the root of the analytical functions can be constructed using the Lagrange expansion theorem, see, e.g., Cohen (1982, Appendix, Sect. 6).

Remark 3 The marginal queue length distributions with the autonomous-server discipline can be readily obtained by analyzing each individual queue as a single-server queue with vacation, see, e.g., Nakatsuka (2009). In this case, the vacation duration is equal to the sum of the server visit time to the other queues plus the switch-over times between the queues. It is clearly seen that this vacation duration is independent of the queue-length which considerably facilitates the marginal analysis of the individual queues. Note that the previous statement does not imply that the lengths of the queues are independent.

4 Time-limited discipline

In this section, we will relate the joint queue-length probabilities at the beginning and end of a server visit to a queue for the time-limited discipline. Under this discipline, the server departs from Q_i when it becomes empty or when a timer of exponentially distributed duration with rate α_i has expired, whichever occurs first. Moreover, if the server arrives to an empty queue, he leaves the queue immediately and jumps to the next queue in the schedule. For this reason, we should distinguish here between the two events where the server joins an empty and non-empty queue.

We will follow the same approach as in Sect. 3. Thus, we first assume that there are $\mathbf{N}_1^b := (i_1, \ldots, i_M)$ jobs in $(Q_1, \ldots, Q_M)$, with $i_1 \geq 1$, at the beginning of a server visit to Q_1 and second there are $\mathbf{N}_1^e := (\mathbf{N}_{11}^e, \ldots, \mathbf{N}_{1M}^e) = (j_1, \ldots, j_M)$ jobs in $(Q_1, \ldots, Q_M)$ at the end of a server visit to Q_1. Note that if Q_1 is empty at the beginning of a server visit, i.e., $i_1 = 0$, then $\mathbb{P}(\mathbf{N}_1^e = \mathbf{N}_1^b) = 1$. We shall exclude the latter obvious case from the analysis in the following. However, we shall include it when the result is unconditioned on $\mathbf{N}_1^b$.

Let $\mathbf{N}(t) := (PH_1(t), N_1(t), \ldots, N_M(t))$ denote the $(M+1)$-dimensional, continuous-time Markov chain with discrete state-space $\xi_T = \{1, \ldots, h_1\} \times \{0, 1, \ldots\}^M \cup \{a\}$, where $N_j(t)$ represents the number of jobs in Q_j at time t and at which Q_1 is being served. State $\{a\}$ is absorbing. We refer to this absorbing Markov chain by $\mathbf{AMC}_T$. The absorption of $\mathbf{AMC}_T$ occurs when the server leaves Q_1 which happens with rate α_1 from all transient states. The transient states of the form $(ph_1, 1, n_2, \ldots, n_M)$ have an additional transition rate to $\{a\}$ that is equal to the (ph_1)-entry of T_1^0 which represents the departure of the last job at Q_1 from the service phase ph_1.

We shall now derive the joint moment of the p.g.f. of $\mathbf{N}_1^e$ and the event that the absorption is due to timer expiration and later the joint conditional p.g.f. of $\mathbf{N}_1^e$ and the event that the absorption is due to Q_1 empty. We set $\mathbf{N}(0) = (PH_1(0), \mathbf{N}_1^b)$, where $PH_1(0)$ is distributed according to π_1, i.e., preemptive repeat discipline. We order the transient states lexicographically first according to n_M, then to $n_{M-1}, \ldots, n_1$, and finally according to ph_1. Similarly to the autonomous-server discipline, during a server visit to Q_1, the number of jobs at Q_j, $j = 2, \ldots, M$, may only increase. It then follows that the transient generator of $\mathbf{AMC}_T$ has the same structure as the transient generator of $\mathbf{AMC}_A$, i.e. it is an upper-triangular Toeplitz matrix of upper-triangular Toeplitz diagonal blocks. Therefore, by the same arguments as for the autonomous-server, we find that the joint moment of the p.g.f. of $\mathbf{N}_1^e$ and the event that the absorption is due to timer expiration, denoted by {timer}, given $\mathbf{N}_1(0)$, reads

$$\mathbb{E}[\mathbf{z}^{\mathbf{N}_1^e} \mathbf{1}_{\{\text{timer}\}} | \mathbf{N}_1^b] = -\alpha_1 \left(\prod_{n=2}^{M} z_n^{i_n} \right) c_1 \left(\sum_{i=2}^{M} \lambda_i \hat{D}_i(z_i) \mathbf{I} + \mathbf{B_2} \right)^{-1} g_1(z_1)^T, \tag{34}$$

where $\mathbf{B}_2 := \mathbf{B}_1 - (\lambda_2 + \cdots + \lambda_M + \alpha_1)\mathbf{I}$, $\mathbf{B}_1$ is the generator matrix of an $M^X/PH/1$ queue restricted to the states with the number of jobs strictly positive, i.e., $\mathbf{B}_1$ is obtained by deleting the first row of blocks and column of the matrix $\mathbf{A}_1$ defined in (5), and where

$$g_1(z_1) := \sum_{j_1 \geq 1} z_1^{j_1} e_{j_1} \otimes e = (z_1 e, z_1^2 e, \ldots), \quad c_1 = e_{i_1} \otimes \pi_1.$$

Let $\mathbf{Q_T}(\mathbf{z}) = \sum_{j=2}^{M} \lambda_j (1 - \hat{D}_j(z_j))\mathbf{I} + \mathbf{B_1}$.

Lemma 7 *The joint moment of the p.g.f. of* $\mathbf{N}_1^e$ *and the event that the absorption is due to timer expiration, given* $\mathbf{N}_1^b = (i_1, \ldots, i_M)$, *is given by*

$$\mathbb{E}[\mathbf{z}^{\mathbf{N}_1^e}\mathbf{1}_{\{\text{timer}\}}|\mathbf{N}_1^b] = \alpha_1 z_1 \left(\prod_{n=2}^M z_n^{i_n}\right) \frac{[z_1^{i_1} - r_1(z_2, \ldots, z_M)^{i_1}][1 - \tilde{B}_1(s_1(\mathbf{z}))]}{s_1(\mathbf{z})[z_1 - \tilde{B}_1(s_1(\mathbf{z}))]}, \tag{35}$$

where $r_1(z_2, \ldots, z_M) = \tilde{B}_1(s_1(r_1(z_2, \ldots, z_M), z_2, \ldots, z_M))$ *and* $s_1(\mathbf{z}) = \alpha_1 + \sum_{j=1}^M [\lambda_j\ (1 - \hat{D}_j(z_j))]$.

Proof Equation (34) yields that

$$\mathbb{E}[\mathbf{z}^{\mathbf{N}_1^e}\mathbf{1}_{\{\text{timer}\}}|\mathbf{N}_1^b] = -\alpha_1 \left(\prod_{n=2}^M z_n^{i_n}\right)\left(\sum_{j_1 \geq 1} z_1^{j_1} u_1(j_1)\right) e^T, \tag{36}$$

where $\mathbf{u}_1 = (u_1(1), u_1(2), \ldots) := c_1(\mathbf{Q}_\mathbf{T}(\mathbf{z}))^{-1}$. Note that $\mathbf{u}_1\mathbf{Q}_\mathbf{T}(\mathbf{z}) = c_1$. Inserting $\mathbf{Q}_\mathbf{T}(\mathbf{z})$ into the latter equation gives that

$$\mathbf{1}_{\{n\geq 2\}}\lambda_1 \sum_{l=1}^{n-1} D_1(n-l)u_1(l)\mathbf{I} + u_1(n)(\mathbf{T}_1 - \theta\mathbf{I}) + u_1(n+1)T_1^0\pi_1 = \mathbf{1}_{\{n=i_1\}}\pi_1, \tag{37}$$

where $n > 0$ and $\theta = \alpha_1 + \lambda_1 + \sum_{j=2}^M \lambda_j(1 - \hat{D}_j(z_j))$. Multiplying (37) by z_1^n and summing over n yields that

$$\sum_{n\geq 1} u_1(z_1) z_1^n = [z_1^{i_1} + u_1(1)T_1^0]\pi_1\mathbf{R}^{-1}. \tag{38}$$

Inserting (38) into (36) we find that

$$\begin{aligned}\mathbb{E}[\mathbf{z}^{\mathbf{N}_1^e}\mathbf{1}_{\{\text{timer}\}}|\mathbf{N}_1^b] &= -\alpha_1 \left(\prod_{n=2}^M z_n^{i_n}\right)[z_1^{i_1} + u_1(1)T_1^0]\pi_1\mathbf{R}^{-1}e^T \\ &= \alpha_1 z_1 \left(\prod_{n=2}^M z_n^{i_n}\right)\frac{[z_1^{i_1} + u_1(1)T_1^0][1 - \tilde{B}_1(s_1(\mathbf{z}))]}{s_1(\mathbf{z})[z_1 - \tilde{B}_1(s_1(\mathbf{z}))]},\end{aligned} \tag{39}$$

where the second equality follows from (26) and $s_1(\mathbf{z}) = \theta - \lambda_1\hat{D}_1(z_1)$. Because the joint moment generating function $\mathbb{E}[\mathbf{z}^{\mathbf{N}_1^e}\mathbf{1}_{\{\text{timer}\}}|\mathbf{N}_1^b]$ in (39) has a singular point at $z_1 = r_1(z_2, \ldots, z_M)$, $|r_1(z_2, \ldots, z_M)| < 1$, it should be removable. Thus,

$$u_1(1)T_1^0 = -r_1(z_2, \ldots, z_M)^{i_1}, \tag{40}$$

where $r_1(z_2, \ldots, z_M) = \tilde{B}_1(s_1(r_1(z_2, \ldots, z_M), z_2, \ldots, z_M))$. Inserting $u_1(1)T_1^0$ into (39) readily gives $\mathbb{E}[\mathbf{z}^{\mathbf{N}_1^e}\mathbf{1}_{\{\text{timer}\}}|\mathbf{N}_1^b]$. □

Lemma 8 *The joint moment of the p.g.f. of* $\mathbf{N}_1^e$ *and the event that the absorption is due to empty* Q_1, *given* $\mathbf{N}_1^b = (i_1, \ldots, i_M)$, *is given by*

$$\mathbb{E}[\mathbf{z}^{\mathbf{N}_1^e}\mathbf{1}_{\{\text{timer}\}}|\mathbf{N}_1^b] = r_1(z_2, \ldots, z_M)^{i_1}\prod_{n=2}^M z_n^{i_n}, \tag{41}$$

where $r_1(z_2, \ldots, z_M) = \tilde{B}_1(s_1(r_1(z_2, \ldots, z_M), z_2, \ldots, z_M))$ *and* $s_1(\mathbf{z}) = \alpha_1 + \sum_{j=1}^{M}[\lambda_j\ (1 - \hat{D}_j(z_j))]$.

Proof The joint moment of the p.g.f. of $\mathbf{N}_1^e$ and the event that the absorption is due to Q_1 being empty, is given by

$$\begin{aligned}\mathbb{E}[\mathbf{z}^{\mathbf{N}_1^e}\mathbf{1}_{\{Q_1 \text{ empty}\}}|\mathbf{N}_1^b] &= -\prod_{n=2}^{M}(z_n^{i_n})c_1\mathbf{Q}_\mathbf{T}(\mathbf{z})^{-1}e_1^T \otimes T_1^0\\ &= -\prod_{n=2}^{M}(z_n^{i_n})u_1(1)T_1^0\\ &= r_1(z_2, \ldots, z_M)^{i_1}\prod_{n=2}^{M} z_n^{i_n},\end{aligned}$$

where $\mathbf{u}_1 = c_1(\mathbf{Q}_\mathbf{T}(\mathbf{z}))^{-1}$ and the last equality follows from (40). □

Combining Lemmas 7 and 8 we obtain our main theorem for the time-limited discipline.

Theorem 3 (Time-limited discipline) *The generating function of the joint queue-length of* $Q_1, \ldots, Q_M$ *at the end of the server visit to* Q_1 *is given by*

$$\gamma_1^T(\mathbf{z}) = p_1^T(\mathbf{z})\beta_1^T(\mathbf{z}_1^*) + q_1^T(\mathbf{z})\beta_1^T(\mathbf{z}),$$

where $\mathbf{z} = (z_1, \ldots, z_M)$, $\mathbf{z}_1^* = (r_1(z_2, \ldots, z_M), z_2, \ldots, z_M)$,

$$p_1^T(\mathbf{z}) = 1 - \frac{\alpha_1}{s_1(\mathbf{z})} \times \frac{z_1(1 - \tilde{B}_1(s_1(\mathbf{z})))}{z_1 - \tilde{B}_1(s_1(\mathbf{z}))}, \qquad q_1^T(\mathbf{z}) = \frac{\alpha_1}{s_1(\mathbf{z})} \times \frac{z_1(1 - \tilde{B}_1(s_1(\mathbf{z})))}{z_1 - \tilde{B}_1(s_1(\mathbf{z}))},$$

where $s_1(\mathbf{z}) = \alpha_1 + \sum_{j=1}^{M}\lambda_j(1 - \hat{D}_j(z_j))$ *and* $r_1(z_2, \ldots, z_M)$ *is the root with smallest absolute value of*: (*solving according to* z_1)

$$z_1 = \tilde{B}_1\big(s_1(\mathbf{z})\big).$$

We deduce that for a server visit to Q_i, $i = 1, \ldots, M$,

$$\gamma_i^T(\mathbf{z}) = p_i^T(\mathbf{z})\beta_i^T(\mathbf{z}_i^*) + q_i^T(\mathbf{z})\beta_i^T(\mathbf{z}), \tag{42}$$

where $\mathbf{z}_i^* = (z_1, \ldots, z_{i-1}, r_i(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_M), z_{i+1}, \ldots, z_M)$,

$$p_i^T(\mathbf{z}) = 1 - \frac{\alpha_i}{s_i(\mathbf{z})} \times \frac{z_i(1 - \tilde{B}_i(s_i(\mathbf{z})))}{z_i - \tilde{B}_i(s_i(\mathbf{z}))}, \qquad q_i^T(\mathbf{z}) = \frac{\alpha_i}{s_i(\mathbf{z})} \times \frac{z_i(1 - \tilde{B}_i(s_i(\mathbf{z})))}{z_i - \tilde{B}_i(s_i(\mathbf{z}))},$$

where $s_i(\mathbf{z}) = \alpha_i + \sum_{j=1}^{M}\lambda_j(1 - \hat{D}_j(z_j))$, and where $r_i(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_M)$ is the root with smallest absolute value of:

$$z_i = \tilde{B}_i\big(s_i(\mathbf{z})\big). \tag{43}$$

Finally, introducing the switch-over times from Q_{i-1} to Q_i, thus by using that $\mathbb{E}[\mathbf{z}^{\mathbf{N}_i^b}] = \mathbb{E}[\mathbf{z}^{\mathbf{N}_{i-1}^e}]\hat{C}^{i-1}(\mathbf{z})$, where $\hat{C}^{i-1}(\mathbf{z})$ is the p.g.f. of the number of Poisson batch arrivals during C^{i-1}, we obtain

$$\gamma_i^T(\mathbf{z}) = p_i^T(\mathbf{z})\gamma_{i-1}^T(\mathbf{z}_i^*)\hat{C}^{i-1}(\mathbf{z}_i^*) + q_i^T(\mathbf{z})\gamma_{i-1}^T(\mathbf{z})\hat{C}^{i-1}(\mathbf{z}). \tag{44}$$

Remark 4 In the particular case where $\hat{D}_i(z_i) = z_i$, i.e. the arriving batches are all of size one, (42) agrees with de Haan (2009, Theorem 5.10).

Remark 5 (Exhaustive discipline) Taking the limit of (42) for $\alpha_i \to 0$ the time-limited discipline is equivalent to the exhaustive discipline. We find that

$$\mathbb{E}\big[\mathbf{z}^{\mathbf{N}_i^e}\big] = \mathbb{E}\big[(\mathbf{z}_i^*)^{\mathbf{N}_i^b}\big], \tag{45}$$

where $\mathbf{z}_i^* := (z_1, \dots, z_{i-1}, y_i, z_{i+1}, \dots, z_M)$ and y_i is the root of

$$z_i = \tilde{B}_i\left(\sum_{j=1}^{M} \lambda_j(1 - \hat{D}_j(z_j))\right). \tag{46}$$

Equation (45) is equivalent to the well-known relation for the exhaustive discipline in (see, e.g., (Eisenberg 1972, (24))).

5 Iterative scheme and implementation issues

In this section, we shall explain how to obtain the joint queue-length distribution embedded at the server departure instants from the queues using an iterative scheme. This scheme is similar for the autonomous-server and the time-limited discipline. For this reason, in the following we shall drop the super-script of $\gamma_i^A(\mathbf{z})$ and $\gamma_i^T(\mathbf{z})$. Let $\gamma_i(\mathbf{z})$ denote a generic joint queue-length generating function embedded at the server departure instants from Q_i, $i = 1, \dots, M$. In the following, we first explain how to obtain $\gamma_i(\mathbf{z})$ as function $\gamma_{i-1}(\mathbf{z})$, $\mathbf{z} = (z_1, \dots, z_M)$. Second, we describe in detail our iterative scheme.

Note that $\gamma_i(\mathbf{z})$ is a function of $\gamma_{i-1}(\mathbf{z})$ and $\gamma_{i-1}(\mathbf{z}_i^*)$ where $\mathbf{z}_i^* = (z_1, \dots, z_{i-1}, r_i, z_{i+1}, \dots, z_M)$ with $|z_i| = 1$, $i = 1, \dots, M$ and $|r_i| \le 1$. Moreover, we note that r_i is the root defined in (32) and (43) that is a function of z_l for all $l = 1, \dots, M$ and $l \ne i$. Since $\gamma_{i-1}(\mathbf{z})$ is a p.g.f. it should be analytic in z_i for all $z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_M$. Hence, we can write

$$\gamma_{i-1}(\mathbf{z}) = \sum_{m=0}^{\infty} g_{im}(z_1, \dots, z_{i-1}, z_{i+1} \dots, z_M) z_i^m, \quad |z_i| \le 1, \tag{47}$$

where $g_{im}(.)$ is again an analytic function that is given by

$$g_{im}(z_1, \dots, z_{i-1}, z_{i+1} \dots, z_M) = \frac{1}{2\pi \mathbf{i}} \oint_C \frac{\gamma_{i-1}(\mathbf{z})}{z_i^{m+1}} dz_i, \quad m = 0, 1, \dots, \tag{48}$$

where C is the unit circle and $\mathbf{i}^2 = -1$. From complex function theory, it is well known that (see, e.g., Titchmarsh 1976)

$$\gamma_{i-1}(\mathbf{z}_i^*) = \frac{1}{2\pi \mathbf{i}} \oint_C \frac{\gamma_{i-1}(\mathbf{z})}{z_i - r_i} dz_i, \quad |r_i| \le 1.$$

These formulas show that we only need to know the p.g.f. $\gamma_{i-1}(\mathbf{z})$ for all $\mathbf{z}$ with $|z_i| = 1$, to be able to compute $\gamma_i(\mathbf{z})$.

When there is a switch-over time incurred from queue $i-1$ to i the p.g.f. of the joint queue-length at the end of the nth server visit to Q_i, denoted by $\gamma_i^n(\mathbf{z})$, can be computed as function of $\gamma_{i-1}^n(\mathbf{z})$, see (33) and (44). The kernel step is to iterate over all queues in order to express numerically $\gamma_i^{n+1}(\mathbf{z})$ as function of $\gamma_i^n(\mathbf{z})$. When this is done we say that the algorithm has completed one computational cycle, i.e., it has started at Q_i with an initial value of $\gamma_i^n(\mathbf{z})$ and passed to Q_{i+1} to compute $\gamma_{i+1}^n(\mathbf{z})$, then to Q_{i+2} to compute $\gamma_{i+2}^n(\mathbf{z})$, and so on until it returns to Q_i. After 'infinitely' many cycles, we get $\gamma_i^\infty(\mathbf{z})$, the steady state joint queue-length p.g.f. To find the joint queue-length probability distribution embedded at the server departure from Q_i we use

$$\mathbb{P}\big(\mathbf{N}_i^e = (n_1, \dots, n_M)\big) = \frac{1}{(2\pi\mathbf{i})^M} \oint_C \cdots \oint_C \frac{\gamma_i^\infty(z_1, \dots, z_M)}{z_1^{n_1+1} \cdots z_M^{n_M+1}} dz_1 \cdots dz_M. \tag{49}$$

Since we do not have an explicit analytical form for $\gamma_i^\infty(\mathbf{z})$ we resorted to the following numerical integration

$$\mathbb{P}\big(\mathbf{N}_i^e = (n_1, \dots, n_M)\big) \approx \frac{1}{\prod_{j=1}^M N_j^{\max}} \sum_{k_1=0}^{N_1^{\max}-1} \cdots \sum_{k_M=0}^{N_M^{\max}-1} \frac{\gamma_i^\infty(w_1^{k_1}, \dots, w_M^{k_M})}{(w_1^{k_1})^{n_1} \cdots (w_M^{k_M})^{n_M}}, \tag{50}$$

for $n_i = 0, \dots, N_i^{\max} - 1$, where $w_i = \exp(-2\pi\mathbf{i}/N_i^{\max})$ and $N_i^{\max}$ is the number of discrete points on C used to approximate the ith contour integral in (49), $i = 1, \dots, M$. According to the latter equation it is clearly seen that $\gamma_i^\infty(\cdot)$ only needs to be evaluated at the discrete points $(w_1^{k_1}, \dots, w_M^{k_M})$. For this reason, we shall restrict the computations during the cycles to these discrete points. Note that the integration in (48) can be approximated using the same set of discrete points. In the following, we shall explain how to find $N_i^{\max}$ and when to stop the iterations over the cycles.

We now give more details on our iterative scheme. The scheme runs over a number of consecutive loops that each consists of multiple computational cycles. The loops are introduced to find the best value of $N_i^{\max}$, $i = 1, \dots, M$, that gives an accurate approximation of the embedded joint queue-length probability distribution. At the beginning of a loop, we shall enlarge the number of discrete points on C used to approximate the contour integral in (49). Let us denote by $N_{i,l}^{\max}$, $i = 1, \dots, M$, the number of these discrete points in the lth loop. In the first loop, we set $(N_{1,1}^{\max}, \dots, N_{M,1}^{\max})$ to some initial value. Let $\mathcal{W}_l$ denote the set of discrete points in the lth loop defined as follows,

$$\mathcal{W}_l := \left\{ \big(w_1^{k_1}, \dots, w_M^{k_M}\big) : w_i = \exp\left(\frac{-2\pi\mathbf{i}}{N_{i,l}^{\max}}\right),\ k_i = 0, \dots, N_{i,l}^{\max} - 1, i = 1, \dots, M \right\}.$$

In the lth loop, we run the kernel step, explained previously, for multiple computational cycles until the system converges. In the nth cycle of the lth loop, we shall compute a new approximation of the joint queue-length p.g.f. denoted as $\gamma_i^{l,n}(\mathbf{w})$, $\mathbf{w} \in \mathcal{W}_l$ and $i = 1, \dots, M$. The system converges when $|\gamma_i^{l,n+1}(\mathbf{w}) - \gamma_i^{l,n}(\mathbf{w})|$ is small enough $\forall \mathbf{w}, i$. As seen previously, in the kernel step we need to obtain $\gamma_{i-1}^{l,n}(\mathbf{w}_i^*)$, $\mathbf{w}_i^*$ is the vector $\mathbf{w} \in \mathcal{W}_l$ with the ith entry replaced by $r_i^{k_i}$, in order to compute $\gamma_i^{l,n}(\mathbf{w})$. To do so, we find that it is numerically more stable to first use the *inverse discrete fast Fourier transform* (IFFT) of $\gamma_{i-1}^{l,n}(\mathbf{w})$, $\mathbf{w} \in \mathcal{W}_l$, along the ith dimension. This directly yields $g_{im}(w_1^{k_1}, \dots, w_{i-1}^{k_{i-1}}, w_{i+1}^{k_{i+1}}, \dots, w_M^{k_M})$,

$m = 0, \dots, N_{i,l}^{\max}$, in (48). We then approximate $\gamma_{i-1}^{l,n}(\mathbf{w}_i^*)$ as follows

$$\gamma_{i-1}^{l,n}(\mathbf{w}_i^*) = \sum_{m=0}^{N_{i,l}^{\max}-1} g_{im}(w_1^{k_1}, \dots, w_{i-1}^{k_{i-1}}, w_{i+1}^{k_{i+1}}, \dots, w_M^{k_M}) r_i^m.$$

For more details about the p.g.f. and the FFT we refer to, e.g., Tijms (2003, Appendix D).

We are now ready to explain our iterative scheme:

First loop We start with an empty system and set $N_{i,1}^{\max}$, $i = 1, \dots, M$, to some initial values. Based on these values, we execute the kernel step explained previously, i.e., we compute $\gamma_i^{1,1}(\mathbf{w})$, $\gamma_i^{1,2}(\mathbf{w})$, and so on, $\forall \mathbf{w} \in \mathcal{W}_1$ and $\forall i$. The iteration over the cycles is stopped whenever the system converges, i.e.,

$$|\gamma_i^{1,n+1}(\mathbf{w}) - \gamma_i^{1,n}(\mathbf{w})| \le \epsilon, \quad i = 1, \dots, M, \ \forall \mathbf{w} \in \mathcal{W}_1, \tag{51}$$

where $\epsilon > 0$ is the convergence control parameter. There are two ways to find a new approximation of the embedded joint queue-length distribution from $\gamma_i^{1,n+1}(\mathbf{w})$ that satisfies the last inequality. The first one is by directly applying (50) with $\gamma_i^{\infty}(\mathbf{w})$ replaced by $\gamma_i^{1,n+1}(\mathbf{w})$. The second way is to observe that (50) is nothing else than the inverse Fourier transform equation of $\gamma_i^{\infty}(\mathbf{w})$. Therefore, applying the IFFT algorithm on $\gamma_i^{1,n+1}(\mathbf{w})$, $\forall i$, yields in a fast way the approximation of the embedded joint queue-length distribution, referred to as $\mathbb{P}^1(\mathbf{N}_i^e)$.

Main loop This loop will be executed several times before the algorithm converges. Let l denote the number of times the main loop was executed. In the beginning, we need to check the accuracy of the approximation of the joint queue-length distribution $\mathbb{P}^{l-1}(\mathbf{N}_i^e)$ that was computed at the end of the $(l-1)$st loop. To do so, we first enlarge $N_{i,l}^{\max}$, $\forall i$. To better reflect the system characteristic, we selected the increments to be equal to Δ times the mean queue length of an $M/M/1$ queue with load given by the system parameters, $\Delta \ge 1$. Second, we initialize $\gamma_i^{l,1}(\mathbf{w})$ to the FFT of $\mathbb{P}^{l-1}(\mathbf{N}_i^e)$ using the new values of $N_{i,l}^{\max}$. Third, we repeat the computations in a similar way to the first loop, i.e., we compute $\gamma_i^{l,2}(\mathbf{w})$, $\gamma_i^{l,3}(\mathbf{w})$, and so on. This is done $\forall \mathbf{w} \in \mathcal{W}_l$ and $\forall i$. The iteration over the cycles is stopped when a similar condition to (51) is satisfied. By analogy with the first loop, inverting $\gamma_i^{l,n}(\mathbf{w})$ using the IFFT algorithm gives the steady state joint queue-length distribution at the server departure instants from Q_i, referred to as $\mathbb{P}^l(\mathbf{N}_i^e)$, $i = 1, \dots, M$. Finally, we check the number of cycles required in the current loop to the system to converge. If it is *equal to* 1, we deduce that $\gamma_i^{l,n}(\mathbf{w})$ is the steady state embedded joint queue-length transform; otherwise, we repeat the main loop.

We conclude that at the end of execution of our scheme we have the joint queue-length distribution at the server departure instant from Q_i, $\forall i$. In the following, we shall analyze the computational costs of our proposed scheme.

Remark 6 According to (51) we determine the DFT points up to an error of order ϵ. In the following, we shall prove that an error of order ϵ in the DFT points corresponds to an error in the probabilities of order ϵ. Let us first introduce some notations. Let $\gamma_i^{exact}(\mathbf{z})$ denote the exact DFT at point $\mathbf{z} = (z_1, \dots, z_M)$. Let $\gamma_i^{app}(\mathbf{z})$ denote an approximation of the DFT at $\mathbf{z}$ such that $|\gamma_i^{exact}(\mathbf{z}) - \gamma_i^{app}(\mathbf{z})| < \epsilon$, $\forall \mathbf{z}$ and i. Using the inverse transform we have that the exact probability density of $\mathbf{N}_i^e$ at point $(n_1, \ldots, n_M)$ is equal to

$$\mathbb{P}^{exact}\big(\mathbf{N}_i^e = (n_1, \ldots, n_M)\big) = \frac{1}{\prod_{j=1}^M N_j^{\max}} \sum_{k_1=0}^{N_1^{\max}-1} \cdots \sum_{k_M=0}^{N_M^{\max}-1} \frac{\gamma_i^{exact}(w_1^{k_1}, \ldots, w_M^{k_M})}{(w_1^{k_1})^{n_1} \cdots (w_M^{k_M})^{n_M}}.$$

In addition, the approximate probability density of $\mathbf{N}_i^e$ at point $(n_1, \ldots, n_M)$ is given by

$$\mathbb{P}^{app}\big(\mathbf{N}_i^e = (n_1, \ldots, n_M)\big) = \frac{1}{\prod_{j=1}^M N_j^{\max}} \sum_{k_1=0}^{N_1^{\max}-1} \cdots \sum_{k_M=0}^{N_M^{\max}-1} \frac{\gamma_i^{app}(w_1^{k_1}, \ldots, w_M^{k_M})}{(w_1^{k_1})^{n_1} \cdots (w_M^{k_M})^{n_M}}.$$

The difference between the exact and the approximate probability density of $\mathbf{N}_i^e$ at $(n_1, \ldots, n_M)$ gives,

$$\begin{aligned}
&\big|\mathbb{P}^{exact}\big(\mathbf{N}_i^e = (n_1, \ldots, n_M)\big) - \mathbb{P}^{app}\big(\mathbf{N}_i^e = (n_1, \ldots, n_M)\big)\big| \\
&\quad < \frac{1}{\prod_{j=1}^M N_j^{\max}} \left| \sum_{k_1=0}^{N_1^{\max}-1} \cdots \sum_{k_M=0}^{N_M^{\max}-1} \frac{\gamma_i^{exact}(z_1, \ldots, z_M) - \gamma_i^{app}(z_1, \ldots, z_M)}{z_1^{n_1+1} \cdots z_M^{n_M+1}} \right| \\
&\quad < \frac{1}{\prod_{j=1}^M N_j^{\max}} \sum_{k_1=0}^{N_1^{\max}-1} \cdots \sum_{k_M=0}^{N_M^{\max}-1} \left| \frac{\gamma_i^{exact}(z_1, \ldots, z_M) - \gamma_i^{app}(z_1, \ldots, z_M)}{z_1^{n_1+1} \cdots z_M^{n_M+1}} \right| \le \epsilon.
\end{aligned}$$

Remark 7 Using the finite summations in (50) as an approximation of the multidimensional contour integrations in (49) it is clear that an error is induced. This error is known in the literature as the aliasing error. We refer the reader to Abate and Whitt (1992) and Daigle (1989) for approaches to correct for these errors. We note that we did not apply these approaches in our algorithm. This is because we would like to keep our algorithm as simple as possible. Moreover, the comparison between the simulation and our scheme of the mean, the second moment, and the joint moment of the queue-length is giving a satisfactory result.

6 Computational costs

We measure the computational cost of our scheme in terms of the total number of cycles and the total run (CPU) time required for the scheme to converge. In addition, we are also interested in *the number of points* on the unit circle C used to approximate the multiple contour integrations in (49) defined as:

$$S := \prod_{i=1}^{M} N_{i,L}^{\max},$$

where $N_{i,L}^{\max}$ is the number of points in the ith (dimension) summation in (50) when the scheme converges in the last loop L. The number of points gives an indication on the amount of computer memory required by the scheme to represent the multidimensional transforms $\gamma_i^{L,n}(\mathbf{w})$.

We implemented our scheme in Matlab version 7.8.0 release 2009*a* where we extensively used its multidimensional FFT package. We performed the experiments on an Intel dual core computer of a processor speed 2.8 GHz and 3 GB memory RAM.

6.1 Scenario

In the following, we shall consider a polling system operating under the autonomous-server discipline, which consists of three queues, i.e., $M = 3$. At the end of this section we shall discuss the impact of M on the computational costs. The arrivals to Q_i, $i = 1, 2, 3$, are Poisson batch processes with inter-arrival rate λ_i and geometrically distributed batch size with success probability $p = 0.95$ and with batch size strictly positive. The service time distribution of the jobs in Q_i follows a two-phase Coxian distribution with mean $1/\mu_i$ and squared coefficient of variation c_s^2. We shall consider an asymmetric case in which $\lambda_1 = \lambda_2 = \lambda_3 = \lambda$, $\mu_1 = 1/\mu$, $\mu_2 = 2/\mu$ and $\mu_3 = 3/\mu$, and the rates of the server visit time to Q_i, $i = 1, 2, 3$, are equal to $\alpha_1 = 0.4\alpha$, $\alpha_2 = 1.0\alpha$, and $\alpha_3 = 0.7\alpha$. The switch-over times between the queues are deterministic and equal to 1. We define the average load per queue as follows:

$$\bar{\rho} := \frac{\sum_{i=1}^{M} \rho_i/\kappa_i}{M},$$

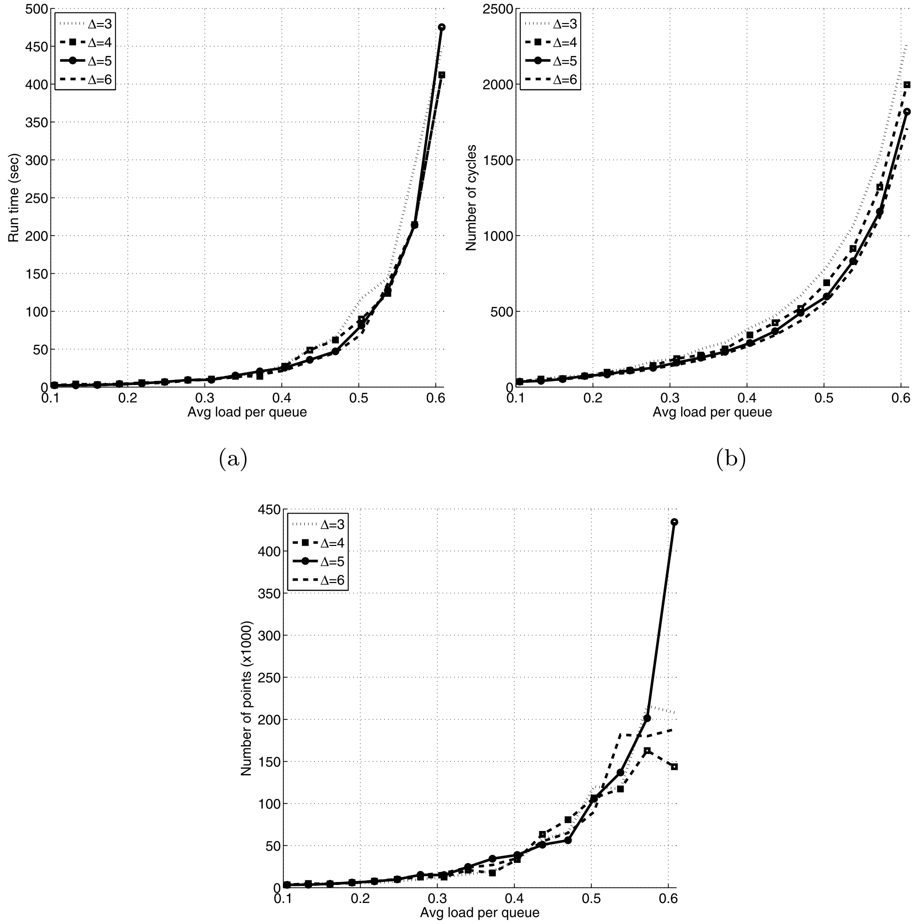
where ρ_i and κ_i are given in Lemma 2. According to the previous parameters setting we find that $\rho_1/\kappa_1 \approx 1.9\rho_3/\kappa_3$ and $\rho_2/\kappa_2 \approx 2.4\rho_3/\kappa_3$. Therefore, Q_3 has the smallest load and Q_2 has the highest load. Finally, we set the convergence control parameter ϵ to 10^{-6} and the initial number of points $(N_{1,1}^{\max}, N_{2,1}^{\max}, N_{3,1}^{\max}) = (10, 10, 10)$. We note that as ϵ decreases the joint probability distribution becomes more precise but this comes at the expense of a higher computational cost.

In the following section, we shall evaluate the computation complexity of the scheme as function of: (1) the service rate μ_i, (2) the arrival rate λ, (3) the server visit rate α_i, (4) the squared coefficient of variation of the service times.

6.2 Evaluation

Let us first focus on the impact of the service rate on the computation complexity of the scheme. We vary $\mu \in [0.2, 2.0]$ and fix $\lambda = 0.08$ and $\alpha = 1$. We set c_s^2 to 0.5 for all the queues. In Fig. 1, we show the run time, the number of cycles and the number of points, $\prod_{i=1}^{M} N_{i,L}^{\max}$, for different values of Δ. Recall that Δ is the increment multiplier of $N_{i,l}^{\max}$ after each loop (see the main loop just before Remark 6). Observe that the computation complexity of the scheme tends to increase monotonically as function of the average load per queue, $\bar{\rho}$. We shall discuss later the behavior of the number of points S. Note that for $\bar{\rho} \le 0.4$ the parameter Δ has a minor impact on the computation complexity in contrast to the case where the average load is between $[0.4, 0.6]$. In this case, the value of $\Delta = 6$ achieves the best performance especially in term of the run time.

Observe that the scheme experiences different convergence behavior for different load, which explains the reason that in Fig. 1(c) the number of points drops for $\Delta = 3, 4, 6$ and $\bar{\rho}$ between 0.57 and 0.61. More precisely, Table 1 shows the convergence results with $\Delta = 4$ and for $\bar{\rho}$ equal to 0.57 and 0.61. In the case with higher load the scheme requires six loops to converge. We now discuss this result. Recall that in the first loop the number of points is equal to $N_{1,1}^{\max} * N_{2,1}^{\max} * N_{3,1}^{\max} = 10 * 10 * 10 = 1000$. Moreover, after the lth loop we enlarge $N_{i,l}^{\max}$, $\forall i, l$, by an amount that is equal to Δ times the mean queue length of an $M/M/1$ with a load equal to ρ_i/κ_i. Therefore, we find that for $\Delta = 4$ and $\bar{\rho} = 0.57$ the increment vector of $N_{i,l}^{\max}$, $i = 1, 2, 3$, is equal to $(7, 15, 2)$ and for $\bar{\rho} = 0.61$ it is equal to $(8, 21, 3)$. Comparing the number of points in both cases we find that in the 8th loop for $\bar{\rho} = 0.57$ it is equal to $N_{1,8}^{\max} * N_{2,8}^{\max} * N_{3,8}^{\max} = 59 * 115 * 24 = 162840$ and in the 6th loop for $\bar{\rho} = 0.61$



(a) (b)

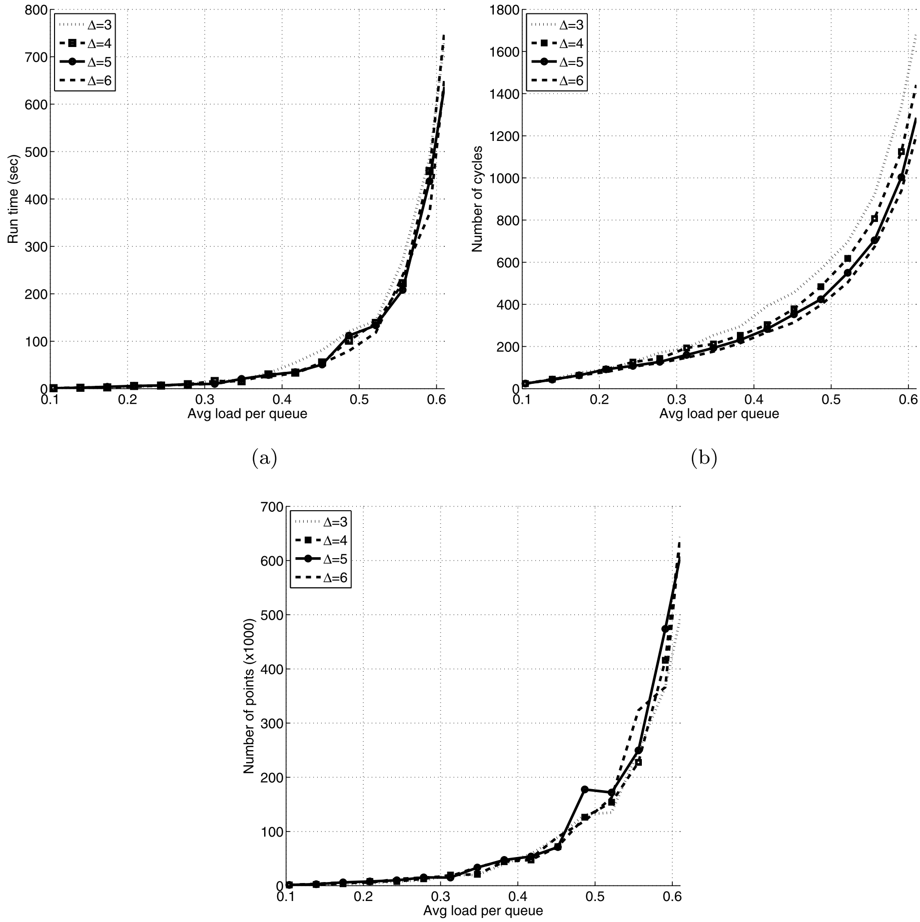
(c)

Fig. 1 Scheme computational cost in terms of the run time (**a**), the total number of cycles (**b**) and the number of points (**c**), as function of $\bar{\rho}$, the average load per queue, and for different values of Δ obtained with $\lambda = 0.08$, $\mu \in [0.2, 2.0]$, $\alpha = 1$ and $c_s^2 = 0.5$. Note that an average load per queue $\bar{\rho} = 0.61$ corresponds to the load in (Q_1, Q_2, Q_3) that is equal to $(0.64, 0.83, 0.35)$

Table 1 Scheme convergence behavior for different values of $\bar{\rho}$ with $\Delta = 4$, $\lambda = 0.08$, $\alpha = 1$ and $c_s^2 = 0.5$. These results are complementary to those in Fig. 1(c)

Avg. load ($\bar{\rho}$)	No. of cycles in the consecutive loops								Total cyc.
0.57	60	248	383	372	232	16	8	1	1320
0.61	63	381	638	608	305	1			1996

it is equal to $N_{1,6}^{\max} * N_{2,6}^{\max} * N_{3,6}^{\max} = 50 * 115 * 25 = 143750$. Since Q_2 is the queue with the highest load we deduce that $N_{2,l}^{\max} \geq 115$ is a sufficient condition for the convergence in both cases. Since the increment for $N_{2,l}^{\max}$ for $\bar{\rho} = 0.57$, which is 15, is much smaller than that for



(a) (b)

(c)

Fig. 2 Scheme computational cost in terms of the run time (**a**), the total number of cycles (**b**) and the number of points (**c**), as function of $\bar{\rho}$ for different values of Δ obtained with $\lambda = [0.03, 0.17]$, $\mu = 1$, $\alpha = 1$ and $c_s^2 = 0.5$. Note that $\bar{\rho} = 0.59$ corresponds to the load in (Q_1, Q_2, Q_3) that is equal to $(0.62, 0.80, 0.35)$

$\bar{\rho} = 0.61$, which is 21, this explains the larger number of loops required in the first case. In addition, this comes with $N_{1,8}^{\max} = 59$ for the case with $\bar{\rho} = 0.57$ compared to $N_{1,6}^{\max} = 50$ for $\bar{\rho} = 0.57$. On one hand this explains the reason that the number of points is smaller for load 0.61 compared to 0.57. On the other hand, the considerably smaller total number of cycles in the case of 0.57, see Table 1, explains the reason that the run time is much smaller than the case of 0.61. Similar results hold for $\Delta = 3$ and $\Delta = 6$.

In Fig. 2, we evaluate the algorithm complexity as function of the average load per queue obtained by varying $\lambda \in [0.03, 0.18]$. By analogy with the previous case of different values of μ we find that: (1) the computation complexity of the scheme increases with the average load per queue, (2) the computation complexity of the scheme is insensitive to the value of Δ for average load smaller than 0.4, and (3) $\Delta = 6$ yields the best performance especially when the average load is high. Observe that the scheme in Fig. 1 requires less time to converge

Table 2 Scheme convergence behavior with $\varDelta = 6$ for two different scenarios with the same $\bar{\rho}$ equal to 0.5005

Q_1, Q_2, Q_3 loads	No. of cycles in the consecutive loops							Total cyc.	Run time
0.5286, 0.676, 0.2977	32	96	129	113	57	8	1	436	93.5 sec
0.5278, 0.684, 0.2891	58	170	194	130	8	1		561	71.4 sec

than in Fig. 2. This is because of the possibility that a different settings of the loads yield the same average load per queue. To explain this issue let us consider the following settings. We fixed $\alpha = 1$ $c_s^2 = 0.5$ and first set λ to 0.144 and μ to 1, and second set λ to 0.08 and μ to 1.691. These two settings yield an average load per queue equal to 0.5005. Note that in the first setting the load in Q_1, Q_2, and Q_3 are equal to 0.5287, 0.676 and 0.2977, however in the second case the load in the queues are equal to 0.5278, 0.684 and 0.2891. Observe that there is a slight difference of 0.008 especially for the load in Q_2, which happens to be the queue with the highest load in both settings. Table 2 shows the convergence sensitivity to the small deviation in the loads in the two cases. Observe that in the second case with a higher load in Q_2 the scheme requires more cycles per loop in the starting phase but this comes with a smaller total number of loops, which yields a smaller run time.

We note that we evaluated the impact of α on the scheme computation cost with $\lambda = 0.1$, $\mu = 1.2$, $c_s^2 = 0.5$, $\varDelta = 5$ and $\alpha \in [0.4, 1.5]$. Observe that as α increases the server visit time to Q_i is smaller, which makes the loads in the queues increase. For this reason, we numerically noticed that the computational cost of the scheme increases monotonically with α. In addition, we evaluated the scheme run time as function of the squared coefficient of variation, c_s^2, with a fixed mean service time at Q_i equal to $1/\mu_i$ and different values of λ, and for $\mu = \alpha = 1$. Observe that the run time tends to decrease as function of c_s^2. On the one hand, this is due to the preemptive discipline considered in this section that forces the load to decrease as function of c_s^2. On the other hand, as c_s^2 increases the queues become more variable in size which compensates for the load reduction caused by a higher c_s^2. For example, an almost equal run time is experienced for $c_s^2 = 1.5, 2.5$ with $\lambda = 1.2$.

We conclude that for an average load per queue smaller or equal to 0.5 the run time of our scheme is smaller than 100 sec and the number of points is smaller than 150000.

7 Comparison with other numerical methods

In this paper we developed an iterative scheme to compute the joint queue-length distribution at embedded epochs of the time-limited polling systems. This is done using the closed-form relation between the p.g.f. of the joint queue-length at the beginning and the end of a server visit to a queue. Another way to solve our problem is to represent our model as a finite-state Markov chain and apply a numerical method to compute the steady-state probabilities. In order to do so, it is necessary to assume that the switch-over times are distributed according to a phase-type distribution. In addition, it is necessary to apply a dynamic approach that requires multiple loops to truncate the queues at a proper value to satisfy a predefined convergence criterion. This will result in a large, finite-state, structured Markov chain that should be solved in each loop where the size of the queues is updated. The literature on numerical solution of Markov chain is abundant. The most commonly used methods are the iterative methods. For an overview on this topic see, e.g., Bolch et al. (2006, Chap. 3), Malhis and Sanders (1996), Philippe et al. (1992) and Stewart (2009, Chap. 10).

Recently, Van Houdt in Van Houdt (2010) proposed a numerical solution for the polling systems. Van Houdt approach is based on the iterative method especially the so-called power method. In Van Houdt (2010), the author studied a discrete-time Bernoulli polling systems with zero switch-over times. The Bernoulli service discipline includes as a particular case the exhaustive and k-limited discipline but not the time-limited discipline. In Van Houdt (2010), it is proposed to truncate the queues in order to obtain a large, structured, finite Markov chain. The analysis of the Markov chain relies on the power method together with the shuffle algorithm and the Kronecker structure to speed up the computations. In addition, they have a dynamic approach similar to us that requires multiple loops in order to truncate the queues at a proper size. Our model is different from their model in the sense that we have a continuous-time time-limited polling systems. Despite these facts, a comparison between the run time of our and their algorithm shows that both algorithms have a comparable performance. More precisely, in Van Houdt (2010) it is reported there that the algorithm requires less than 4 sec to converge with a total offered load equal to 0.7. This result is obtained for the discrete-time, zero switch-over times, exhaustive polling system that consists of four queues. We implemented exactly the same dynamic approach with a more precise convergence parameter than the one in Van Houdt (2010), i.e. $\epsilon = 10^{-10}$ instead of $\epsilon = 10^{-7}$, but for a continuous-time, zero switch-over times, exhaustive polling system that consists of four queues. For the same offered load 0.7, our algorithm converges in less than 2 sec. This comparison shows that both algorithms have a comparable result.

The advantage of our algorithm compared to the one in Van Houdt (2010) is that we can consider an arbitrarily distributed switch-over time without the need to approximate it with a phase-type distribution. Moreover, we believe that our algorithm can be extended for the case with arbitrarily distributed service time. The advantage of the algorithm in Van Houdt (2010) compared to our is that it is more generic. This is because our algorithm requires the derivation beforehand of the relation between the p.g.f. of the joint queue-length at the beginning and the end of a server visit to a queue.

8 Discussion

Let us discuss the impact of adding a queue in the system on the computation complexity of the scheme. First, note that in this case the load in the queues increase. This is because the availability of the server in the queues decrease. Second, the time required for a computational cycle increases linearly with the number of added queues. Third, the total number of cycles increases monotonically because of the higher load in the queues. In the end, all these increments add together to make the run time increase monotonically with a similar form of those in Figs. 1 and 2(a) but in a much faster way.

We now discuss the assumption that the server visits time to the queues are exponentially distributed. The general case with arbitrarily distributed visit time cannot be tackled with our approach. However, as an approximation one can fit a phase-type distribution to (some) moments of the general distribution. In this case, our approach can be modified as follows. We embed the joint queue-length at the beginning instants of the visit phases. Extending the kernel relation in Theorems 2 and 3 we can relate the queue-length p.g.f. at the beginning and the end of a server visit to a queue. This is possible by conditioning on the phase of the service time, of the customer in service, at the end of the phases of the server visit to the queue. By analogy with the iterative scheme in Sect. 5 one can compute the joint queue-length distribution embedded at the end of a server visit phase. We expect that the computational costs will increase linearly with the number of the server visit phases.

9 Conclusion

In this paper, we have developed a general framework to analyze polling systems with Poisson batch arrivals and phase-type service times for the autonomous-server and the time-limited service discipline. The framework is based on the key idea of relating directly the joint queue-lengths distribution at the beginning and the end of a server visit. In order to do so, we used the theory of absorbing Markov chains. We have illustrated our framework for the autonomous-server and the time-limited service discipline. The analysis presented in this paper is restricted to the case of a single job service at a time. We emphasize that the analysis can be extended to the more general batch service disciplines, see Cohen (1982, Chap. III.2). For instance, Lemma 6 holds in this case, however, the matrix $\mathbf{A_2}$ becomes a full block matrix.

In this paper we have shown that our framework is applicable to disciplines that do not satisfy the branching property which are, in general, considered to be hard to analyze. Our framework is also applicable to branching type polling systems such as the exhaustive and the gated discipline.

Acknowledgements We first would thank the reviewers for their valuable comments which helped to improve the paper. In the Netherlands, the 3 universities of technology have formed the 3TU.Federation. This article is the result of joint research in the 3TU.Centre of Competence NIRICT (Netherlands Institute for Research on ICT). The authors would also thank De Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) for their financial support.

References

Abate, J., & Whitt, W. (1992). Numerical inversion of probability generating functions. *Operations Research Letters*, *12*(4), 245–251.

Al Hanbali, A., de Haan, R., Boucherie, R. J., & van Ommeren, J.-K. (2008a). A tandem queueing model for delay analysis in disconnected ad hoc networks. In *LCNS: Vol. 5055*. *Proc. of ASMTA* (pp. 189–205), Nicosia, Cyprus, June 2008.

Al Hanbali, A., de Haan, R., Boucherie, R. J., & van Ommeren, J.-K. (2008b). Time-limited and k-limited polling systems: a matrix analytic solution. In *Proc. of SMCTools*, Athens, Greece, Oct. 2008.

Bernstein, D. S. (2005). *Matrix mathematics*. Princeton: Princeton University Press.

Blanc, J. (1992a). An algorithmic solution of polling models with limited service disciplines. *IEEE Transactions on Communications*, *40*(7), 1152–1155.

Blanc, J. (1992b). Performance evaluation of polling systems by means of the power-series algorithm. *Annals of Operation Research*, *35*(3), 155–186.

Blanc, J. (1998). The power-series algorithm for polling systems with time limits. *Probability in the Engineering and Informational Sciences*, *12*, 221–237.

Bolch, G., Greiner, S., de Meer, H., & Trivedi, K. (2006). *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. New York/Oxford: Wiley/Blackwell.

Cohen, J. W. (1982). *The single server queue*. Amsterdam: North-Holland.

Daigle, J. (1989). Queue length distributions from probability generating functions via discrete Fourier transforms. *Operations Research Letters*, *8*(4), 229–236.

de Haan, R. (2009). *Queueing models for mobile ad hoc networks*. PhD thesis, Enschede, June 2009. http://doc.utwente.nl/61385/.

de Haan, R., Boucherie, R. J., & van Ommeren, J.-K. (2009). A polling model with an autonomous server. *Queueing Systems*, *62*(3), 279–308.

Eisenberg, M. (1972). Queues with periodic service and changeover times. *Operations Research*, *20*(2), 440–451.

Fricker, C., & Jaibi, M. (1994). Monotonicity and stability of periodic polling models. *Queueing Systems*, *15*(1–4), 211–238.
Gaver, D. P., Jacobs, P. A., & Latouche, G. (1984). Finite birth-and-death models in randomly changing environments. *Advances in Applied Probability*, *16*, 715–731.
Grinstead, C., & Snell, J. (1997). *Introduction to Probability*. Providence: American Mathematical Society.
Guillemin, F., & Simonian, A. (1995). Transient characteristics of an M/M/1/infinity system. *Advances in Applied Probability*, *27*, 862–888.
Leung, K. (1991). Cyclic-service systems with probabilistically-limited service. *IEEE Journal on Selected Areas in Communications*, *9*(2), 185–193.
Leung, K. (1994). Cyclic-service systems with non-preemptive time-limited service. *IEEE Transactions on Communications*, *42*(8), 2521–2524.
Levy, H., & Sidi, M. (1990). Polling systems: applications, modeling, and optimization. *IEEE Transactions on Communications, 38*(10).
Malhis, L., & Sanders, W. (1996). An efficient two-stage iterative method for the steady-state analysis of Markov regenerative stochastic Petri net models. *Performance Evaluation*, *27*, 583–601.
Nakatsuka, T. (2009). Queue length distribution in M/G/1, M^x/G/1 and their variants with completion time. *Journal of the Operations Research*, *52*(1), 11–34.
Neuts, M. (1981). *Matrix-geometric solutions in stochastic models: an algorithmic approach*. Baltimore: Johns Hopkins University Press.
Philippe, B., Saad, Y., & Stewart, W. (1992). Numerical methods in Markov chain modeling. *Operations Research*, *40*(6), 1156–1179.
Resing, J. (1993). Polling systems and multitype branching processes. *Queueing Systems*, *13*(10), 409–429.
Stewart, W. (2009). *Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling*. Princeton: Princeton University Press.
Takagi, H. (2000). Analysis and application of polling models. In *LNCS: Vol. 1769*. *Performance evaluation: origins and directions* (pp. 423–442). Berlin: Springer.
Tijms, H. (2003). *A first course in stochastic models*. New York: Wiley.
Titchmarsh, E. (1976). *The theory of functions*. Oxford: Oxford Science Publications.
Van Houdt, B. (2010). Numerical solution of polling systems for analyzing networks on chips. In *Proc. of NSMC*, Virginia, USA.
van Vuuren, M., & Winands, E. (2007). Iterative approximation of k-limited polling systems. *Queueing Systems: Theory and Applications*, *55*(3), 161–178.
Yechiali, U., & Eliazar, I. (1998). Polling under the randomly-timed gated regime. *Stochastic Models*, *14*(1), 79–93.