J. Appl. Prob. **49,** 84–99 (2012)
Printed in England

A RECURRENT SOLUTION OF PH/M/c/N-LIKE AND PH/M/c-LIKE QUEUES

ALEXANDRE BRANDWAJN,* *University of California Santa Cruz*
THOMAS BEGIN,** *Université Lyon 1*

Abstract

We propose an efficient semi-numerical approach to compute the steady-state probability distribution for the number of requests at arbitrary and at arrival time instants in PH/M/c-like systems with homogeneous servers in which the interarrival time distribution is represented by an acyclic set of memoryless phases. Our method is based on conditional probabilities and results in a simple computationally stable recurrence. It avoids the explicit manipulation of potentially large matrices and involves no iteration. Owing to the use of conditional probabilities, it delays the onset of numerical issues related to floating-point underflow as the number of servers and/or phases increases. For generalized Coxian distributions, the computational complexity of the proposed approach grows linearly with the number of phases in the distribution.

Keywords: G/M/c-like queue; phase-type distribution; conditional probability; queue length distribution; recurrent solution; numerical stability; computational efficiency; asymptotic geometric distribution

2010 Mathematics Subject Classification: Primary 60K25; 90B22; 68M20

1. Introduction

In many computer and networking applications, as well as a number of other areas, the request arrival process exhibits significant deviations from a simple Poisson process. This is the case, for example, for Internet traffic where times between request arrivals are thought to be highly variable and possibly 'heavy tailed' [14], [23], as well as for I/O subsystems in large computer installations [22]. To achieve ever increasing performance levels at acceptable energetic expense, a frequent solution entails the use of multiple parallel facilities to process the requests, e.g. in mainframe I/O [12]. If we assume memoryless service times, the system congestion and the delays experienced by requests become those of a G/M/c-type queue. Since in man-made systems the buffer sizes can only be finite, under heavier workloads, effects due to a limited queueing room cannot be neglected, so the G/M/c/N-type queue is then a more adequate model. Additionally, in many situations, the service and/or arrival processes may exhibit nonnegligible dependence on the current state of the system. For example, state-dependent service rates allow a more accurate representation of service in systems such as multicore processors [43] where, because of interference between processors, the processing capacity does not grow linearly with the number of active processors. State-dependent arrival rates may be useful, for instance, to represent congestion avoidance in IP networks.

Received 11 June 2010; revision received 18 October 2011.

* Postal address: Baskin School of Engineering, University of California Santa Cruz, 1156 High Street, Mail Stop SOE3, Santa Cruz, CA 95064, USA. Email address: alexb@soe.ucsc.edu

** Postal address: LIP UMR CNRS - ENS Lyon - UCB Lyon 1 - INRIA, Université Lyon 1, 46 allée d'Italie, 69364 Lyon Cedex 7, France. Email address: thomas.begin@ens-lyon.fr

https://doi.org/10.1239/jap/1331216835 Published online by Cambridge University Press

Although there are analytical results for the unrestricted state-independent G/M/c queue [2], [9], [26], their application requires computations which, depending on the specific distribution of interarrival times, may quickly become difficult [18], [27]. To the best of the authors' knowledge, the G/M/c queue with restricted queueing room has received less attention, and there are few easily usable general analytical results [13], [20]. The same seems to apply to G/M/c-type queues with state dependencies [18].

Since any distribution can be approximated arbitrarily closely by a finite number of exponential phases [35], [36], a possible approach is to use a phase-type distribution for the times between request arrivals [8], [16], [21], [24], [37], [41], and, hence, attempt to solve the resulting PH/M/c-type queue.

The matrix-geometric techniques pioneered by Neuts [34], [35] are a possible avenue to evaluate such processes. There is a large body of previous work in the area of matrix-analytic approaches. In particular, Latouche and Ramaswami [29] proposed the logarithmic reduction algorithm as a numerically stable approach to the computation of steady-state probabilities in level-independent systems, i.e. systems without state dependency. This algorithm is based on stochastic complementation [39]. The work of Gaver *et al.* [17] is devoted to finite queues in randomly changing environments, and includes a numerical method that involves a recursive determination of certain matrices. The logarithmic reduction algorithm of Latouche and Ramaswami has been extended by Bright and Taylor [11] to level-dependent infinite queues. Bright and Taylor pointed out possible numerical problems in their approach due to the recursive calculation of matrices involved in the solution. Gaver *et al.* [17] mentioned similar problems. Latouche [28] showed that Newton's method applied to nonlinear equations in Markov chains is quadratically convergent, although not very attractive because of its computational complexity. Akar and Sohraby [1] proposed an invariant subspace approach whose convergence rates are at least quadratic, and whose accuracy may be better due to the avoidance of truncation. The generalization of matrix-geometric stationary distributions to level-dependent quasi-birth-and-death processes was considered by Ramaswami and Taylor [38]. Bean *et al.* [3] studied the quasistationary distributions for level-dependent processes and proposed a method derived from the Latouche–Ramaswami algorithm for their computation. The work of Ye [44] examined the theoretical properties of the Latouche–Ramaswami logarithmic reduction algorithm, including numerical stability issues, and offered a more stable algorithm for inverting a diagonally dominant matrix. An improved matrix-geometric algorithm was proposed by Naoumov *et al.* [33]. The spectral expansion method can also be applied to obtain a solution for these types of systems [15], [31], [32].

We refer the reader to the books by Latouche and Ramaswami [30], Bini *et al.* [5], and, at a more introductory level, Bolch *et al.* [9] for an overview of properties of matrix-analytic approaches and numerical methods for quasi-birth-and-death problems and G/M/1-type Markov chains. Bini *et al.* [6], [7] discussed practical considerations and software implementation for several of the methods mentioned above. Mitrani and Chakka [31] compared the performance of the spectral expansion and matrix-geometric methods for an M/M/c-type queue. Haverkort and Ost [19] also compared the spectral expansion method with the Latouche–Ramaswami algorithm for a model of a fault-tolerant system. Tran and Do [42] presented a comparison of the practical performance of matrix-geometric methods and of the spectral expansion for a specific quasi birth-and-death process.

We propose a considerably simpler semi-numerical approach to compute the steady-state probability distribution for the number of requests in the system both at arbitrary times and at instants of request arrival for PH/M/c-type queues with and without state dependencies.

https://doi.org/10.1239/jap/1331216835 Published online by Cambridge University Press

Our method, inspired by a recent semi-numerical solution for M/C_n/1-type queues [10], exploits the known form of the steady-state distribution, involves no iteration, and results in a simple numerically stable recurrence. It avoids the explicit manipulation of potentially large matrices, and, because of the use of conditional probabilities, reduces the possibility of floating-point numerical problems, especially as the number of servers and/or phases increases.

This paper is organized as follows. In the next section we describe in more detail the PH/M/c-type and PH/M/c/N-type models considered, and the proposed recurrent solution. In Section 3 we present a formal proof of the computational stability of our recurrent solution. In Section 4 we discuss the computational complexity of our approach, and present an example of numerical results obtained using it. Section 5 concludes the paper.

2. Model and its recurrent solution

2.1. State description

The queue considered comprises c homogeneous servers. The service times are assumed to be memoryless with individual service rate $\mu(n)$, where n is the current number of requests in the system. In the case of a restricted queueing room, N is the maximum value for this number.

Similarly to several authors [30], [35], we represent the times between arrivals as a phase-type distribution, which we assume to be acyclic. We denote by a the number of memoryless phases, by $\tau_j(n)$ the probability that the arrival process starts in phase j, $j = 1, \dots, a$, by $\lambda_j(n)$ the rate of phase j, by $r_{jl}(n)$ $(l > j)$ the phase transition probabilities, and by $\hat{r}_j(n)$ the probability that the arrival process terminates after phase j. We consider this queue in steady state for which the joint probability of the current stage of the arrival process and the current number of requests in the system, $p(j, n)$, is a common description. We discuss briefly in Section 2.3 the conditions under which our queue with unlimited queueing room has a steady-state solution. In the case of a finite queueing room, the steady state of our queue always exists. A summary of the principal notation used in this paper is as follows.

c Number of servers.

n Total current number of requests in the system; $n = 0, \dots, N$ for a finite queueing room.

$\tau_j(n)$ Probability that the arrival process starts in phase j, $j = 1, \dots, a$, when there are n requests in the system.

$\lambda_j(n)$ Completion rate for phase j of the arrival process.

$r_{jl}(n)$ Probability that the arrival process continues in phase l upon completion of phase j, where $j, l = 1, \dots, a$, $l > j$.

$\hat{r}_j(n)$ Probability that the arrival process ends (new request generated) upon completion of phase j, $j = 1, \dots, a$.

$u(n)$ Rate of request completions given by $\min(c, n)\mu(n)$, where $\mu(n)$ is the service rate of a single server.

u^* Limiting value of $u(n)$ with unrestricted queueing room.

$p(j \mid n)$ Conditional probability that the arrival stage is j given that the number in the system is n.

$\alpha(n)$ Conditional rate of arrivals given that the number in the system is n (cf. (2) below).

$\tilde{\alpha}$ Limiting value of $\alpha(n)$ with unrestricted queueing room.

$p(n)$ Steady-state probability that the number of customers in the system is n.

$P_{\mathrm{A}}(n)$ Probability that an arriving customer finds n customers already present in the system.
n^* Number of requests for which the state dependence ends for arrivals and service ($\tilde{n} = n^* + 1$).
$\tilde{\rho}$ Limiting geometric factor for an unrestricted queue.

2.2. Restricted queueing room (PH/M/c/N-type queue)

We first consider the case where the queueing room is finite so that no more than N ($N > c$) requests can be in the system (queued and in service). With a finite queueing room, there are several possible assumptions regarding the arrival process when the buffer is full. One simple possibility is that requests arriving to find the buffer full are simply lost (lost arrivals) and the arrival process continues unperturbed. Another possibility, of interest in networking applications, is that the arrival process stops altogether when the buffer becomes full, and remains blocked until a request leaves the system (blocked arrivals). Our approach can accommodate both assumptions.

It is a straightforward matter to obtain the balance equations for the steady-state probability $p(j, n)$ that the current stage of the arrival process is j and that the current number of requests is n. It is also not difficult to show that the marginal probability that there are n requests in the system, $p(n)$, can be expressed as

$$p(n) = \frac{1}{G} \prod_{m=1}^{n} \frac{\alpha(m-1)}{u(m)}, \qquad n = 0, 1, \ldots, \tag{1}$$

where

$$\alpha(n) = \sum_{j=1}^{a} \lambda_j(n) \hat{r}_j(n) p(j \mid n), \tag{2}$$

$u(n) = \min(c, n)\mu(n)$, and G is a normalizing constant chosen so that $\sum_n p(n) = 1$. Note that $\alpha(n)$ can be viewed as the steady-state rate of request arrivals conditioned on the current number of requests already present in the system. Except in the trivial case of Poisson or quasi-Poisson arrivals, $\alpha(n)$ is not known in advance, so our goal is to compute it so as to obtain the steady-state probabilities $p(n)$ from (1).

From the definition of conditional probability, we have $p(j, n) = p(j \mid n)p(n)$. Using this relationship together with (1) in the balance equations, we obtain, for $n = 0, 1, \ldots, N-1$ in the case of lost arrivals,

$$\begin{aligned} p(j \mid n)[\lambda_j(n) + u(n)] = p(j \mid n+1)\alpha(n) + \sum_{l=1}^{j-1} \lambda_l(n) r_{lj}(n) p(l \mid n) \\ + \tau_j(n)u(n), \qquad j = 1, \ldots, a. \end{aligned} \tag{3}$$

With lost arrivals, for $n = N$, we have

$$\begin{aligned} p(j \mid N)[\lambda_j(N) + u(N)] = \sum_{l=1}^{j-1} \lambda_l(N) r_{lj}(N) p(l \mid N) + \tau_j(N)u(N) \\ + \tau_j(N)\alpha(N), \qquad j = 1, \ldots, a. \end{aligned} \tag{4}$$

The equations for the case of blocked arrivals are given in Appendix A.5.

https://doi.org/10.1239/jap/1331216835 Published online by Cambridge University Press

Clearly, we have

$$\sum_{j=1}^{a} p(j \mid n) = 1 \quad \text{for all } n = 0, 1, \dots. \tag{5}$$

Based on the forms of (3) and (4), we let

$$p(j \mid n) = \varphi_j(n)\alpha(n) + \xi_j(n)u(n). \tag{6}$$

We start from $n = N$. Using (6) in (4), we readily obtain the coefficients $\varphi_j(N)$ and $\xi_j(N)$ in the order $j = 1, \dots, a$ (cf. (15) and (16) in Appendix A.3). From (6) and the normalizing condition (5), we determine $\alpha(N)$. We are thus able to compute $p(j \mid N)$ using $\alpha(N)$, and the previously computed coefficients $\varphi_j(N)$ and $\xi_j(N)$ in (6). We then consider consecutive decreasing values of $n = N - 1, \dots, 0$. For each n, the values of $p(j \mid n+1)$ are known, and we easily determine the coefficients $\varphi_j(n)$ and $\xi_j(n)$ using (6) in (3) (cf. (15) and (16) in Appendix A.3). We obtain $\alpha(n)$ using (6) in the normalizing condition (5), and, hence, $p(j \mid n)$ from (6). Details of the recurrent computation are shown in Appendix A.3. Note that the simple recurrence described in the appendix requires that the phase-type distribution of the time between arrivals be acyclic.

Having obtained the values of $\alpha(n)$ for $n = 0, 1, \dots$, we are ready to compute the steady-state probabilities for the number of requests in the system $p(n)$ from (1).

The probability that an arriving customer finds n customers already present in the system, $P_{\mathrm{A}}(n)$, can be expressed as

$$P_{\mathrm{A}}(n) = \frac{\alpha(n)p(n)}{\sum_{i\ge 0}\alpha(i)p(i)}, \qquad n = 0, 1, \dots. \tag{7}$$

From (7), it follows that the loss probability in the case of lost arrivals is given by $P_{\mathrm{A}}(N)$. In the case of blocked arrivals, we have $\alpha(N) = 0$ so that, as expected, $P_{\mathrm{A}}(N) = 0$. The fraction of time during which the arrival process is blocked is then given by $p(N)$. Note that $\alpha(n)$, the conditional state-dependent rate of arrivals, exhibits in general a strong nontrivial dependence on n, except in the case of a Poisson arrival process, in which case $\alpha(n)$ is a constant.

2.3. Unrestricted queueing room (PH/M/*c*-type queue)

We now consider the case of an unrestricted buffer size. We assume that the state dependencies in the arrival process and in the service rate vanish starting with some value of the number of requests in the system, say, $n = n^*$, so that we have $\tau_j(n) = \tau_j^*$, $\lambda_j(n) = \lambda_j^*$, $r_{jl}(n) = r_{jl}^*$, $\hat{r}_j(n) = \hat{r}_j^*$, and $\mu(n) = \mu^*$ (and, hence, $u(n) = u^* = c\mu^*$) for $n \ge n^*$. Following a reasoning similar to that used for a standard G/M/*c* queue [26, p. 75], it can be shown that the distribution of the number of requests found at arrival instants is geometric for $n > n^*$. It is easy to show that the steady-state probabilities $p(n)$ are also geometrically distributed for $n > n^*$. The geometric factor being $\alpha(n)/u^*$, it follows that, for the case $n > n^*$, the $\alpha(n)$ reach their limiting value $\tilde{\alpha}$. Thus, from (2), the conditional probabilities $p(j \mid n)$ must also reach their limiting values $\tilde{p}(j)$. The limiting value $\tilde{\alpha}$ is given by

$$\tilde{\alpha} = \sum_{j=1}^{a} \tilde{p}(j)\lambda_j^* r_j^*.$$

https://doi.org/10.1239/jap/1331216835 Published online by Cambridge University Press

Hence, for $\tilde{n} = n^* + 1$, the values of $p(j \mid n)$ become equal to $\tilde{p}(j)$ and $\alpha(n) = \tilde{\alpha}$. Denoting the geometric factor by $\tilde{\rho} = \tilde{\alpha}/u^*$ we have

$$p(n) = \frac{1}{G}\begin{cases} \displaystyle\prod_{k=1}^{n} \frac{\alpha(k-1)}{u(k)}, & n \le \tilde{n}, \\ \displaystyle\left[\prod_{k=1}^{\tilde{n}} \frac{\alpha(k-1)}{u(k)}\right]\tilde{\rho}^{\,n-\tilde{n}}, & n > \tilde{n}. \end{cases}$$

The normalizing constant G can be written as

$$G = 1 + \sum_{n=1}^{\tilde{n}-1}\prod_{k=1}^{n} \frac{\alpha(k-1)}{u(k)} + \left[\prod_{k=1}^{\tilde{n}} \frac{\alpha(k-1)}{u(k)}\right]\frac{1}{1-\tilde{\rho}}.$$

Clearly, we must have $\tilde{\rho} < 1$, i.e. $\tilde{\alpha} < u^*$ for the steady solution to exist. An in-depth discussion of stability and related issues in a mutiserver queue can be found in the works of Kiefer and Wolfowitz [25] and Scheller-Wolf and Sigman [40].

The limiting values $\tilde{p}(j)$ and $\tilde{\alpha}$ can be determined from the limit of equation (3) for $n \to \infty$:

$$\tilde{p}(j)(\lambda_j^* + u^*) = \tilde{p}(j)\tilde{\alpha} + \sum_{l=1}^{j-1} \lambda_l^* r_{lj}^* \tilde{p}(l) + \tau_j^* u^*, \qquad j = 1, \dots, a. \tag{8}$$

Obviously, we must have $\sum_{j=1}^{a} \tilde{p}(j) = 1$. Based on this fact, a simple bisection can be used to solve (8) (see Appendix A.1). Thus, starting from the value $\tilde{n} = n^* + 1$ and using the limiting values $\tilde{p}(j)$ as $p(j \mid \tilde{n} + 1)$, we compute the values of $p(j \mid n)$, $j = 1, \dots, a$, and $\alpha(n)$ for consecutive decreasing values of $n = \tilde{n}, \tilde{n} - 1, \dots, 0$, from recurrence (3) as described in the preceding section.

In the case of a standard PH/M/c queue, i.e. when there are no state dependencies in the arrival process or the service rate, the steady-state probabilities $p(n)$ are known to have a geometric form starting from $n = c$ [2, p. 698], [26]. Therefore, it is clear that our asymptotic factor $\tilde{\rho}$ coincides with the solution of the well-known equation involving the Laplace–Stieltjes transform of the interarrival time distribution (see, e.g. [2, p. 698]) and [9, Equation (6.85), p. 265]. Thus, the simple bisection (cf. Appendix A.1) for the limiting probabilities $\tilde{p}(j)$ (see (8)) provides an alternate way of computing this solution.

3. Stability of the recurrent solution

In this section we show that our recurrent solution is computationally stable. We follow closely the line of reasoning presented for M/G/1-like queues [10], adapting it to the distribution type and queue considered in this paper. We write the recurrence as

$$\begin{aligned} p(j \mid n)[\lambda_j(n) + u(n)] = p(j \mid n+1)\alpha(n) &+ \sum_{l=1}^{j-1} \lambda_l(n) r_{lj}(n) p(l \mid n) \\ &+ \tau_j(n)u(n), \qquad j = 1, \dots, a, \end{aligned}$$

where we use the notation $p(j \mid N + 1) = \tau_j$ for all j in the case of a finite buffer with lost arrivals, and $p(j \mid N) = 0$ for all j in the case of a finite buffer with blocking.

As described in Section 2, the solution can be expressed as

$$p(j \mid n) = \varphi_j(n)\alpha(n) + \xi_j(n)u(n) \tag{9}$$

https://doi.org/10.1239/jap/1331216835 Published online by Cambridge University Press

with the coefficients $\varphi_j(n)$ and $\xi_j(n)$ given by

$$\varphi_j(n)[\lambda_j(n)+u(n)] = p(j \mid n+1) + \sum_{l=1}^{j-1} \lambda_l(n) r_{lj}(n) \varphi_l(n), \qquad j = 1, \ldots, a, \tag{10}$$

$$\xi_j(n)[\lambda_j(n)+u(n)] = \sum_{l=1}^{j-1} \lambda_l(n) r_{lj}(n) \xi_l(n) + \tau_j(n), \qquad j = 1, \ldots, a. \tag{11}$$

From (5) and (6) we obtain, for $\alpha(n)$,

$$\alpha(n) = \left[1 - u(n)\sum_{j=1}^{k} \xi_j(n)\right] \bigg/ \sum_{j=1}^{k} \varphi_j(n).$$

We assume that the phase-type distribution of the time between arrivals has indeed a stages, so $\lambda_j(n) > 0$ and $\tau_j(n) + \sum_{l=1}^{j-1} r_{lj}(n) > 0$ for $j = 1, \ldots, a$ (in particular, $\tau_1(n) > 0$).

We first show that our recurrence for $p(j \mid n)$ produces positive values.

Lemma 1. *If $p(j \mid n+1) > 0$ for $i > 1$ then we have*

(a) $\varphi_j(n) > 0$ *and* $\xi_j(n) > 0$ *for* $j \geq 1$,

(b) $\alpha(n) > 0$, *and*

(c) $p(j \mid n) > 0$ *for* $j \geq 1$.

Proof. The proof of part (a) follows directly from (10) and (11).

(b) Summing (11) over all values of j and using the fact that $\sum_{j=1}^{a} \tau_j(n) = 1$, we obtain

$$\sum_{j=1}^{a} \xi_j(n)[\lambda_j(n)+u(n)] = \sum_{j=1}^{a}\sum_{l=1}^{j-1} \lambda_l(n) r_{lj}(n) \xi_l(n) + 1.$$

Rearranging the terms and using the fact that $\sum_{l=j+1}^{a} r_{jl}(n) = 1 - \hat{r}_j(n)$, we have

$$1 - u(n)\sum_{j=1}^{a} \xi_j(n) = \sum_{j=1}^{a} \xi_j(n)\lambda_j(n)\hat{r}_j(n) > 0,$$

and, hence, $\alpha(n) > 0$.

(c) Follows directly from the results of (a) and (b).

Lemma 1 establishes that $p(j \mid n) > 0$ for $j \geq 1$ and $n \geq 0$.

Consider now a set of perturbed conditional probabilities, where the perturbation corresponds to floating-point roundoff errors, i.e.

$$\tilde{p}(j \mid n+1) = p(j \mid n+1) + \Delta p_j^{(n+1)}.$$

Because the conditional probabilities are normalized at each step of the recurrence, the perturbations $\Delta p_j^{(n+1)}$ must satisfy $\sum_{j=1}^{a} \Delta p_j^{(n+1)} = 0$. We assume that the perturbations are small so that we have $\tilde{p}(j \mid n+1) > 0$. From (11), it is clear that the perturbation does not

https://doi.org/10.1239/jap/1331216835 Published online by Cambridge University Press

affect $\xi_j(n)$. Let $\tilde{\varphi}_j(n) = \varphi_j(n) + \Delta\varphi_j^{(n)}$ be the solution of (10) when $p(j \mid n+1)$ is replaced by $\tilde{p}(j \mid n+1)$. The perturbations $\Delta p_j^{(n+1)}$ and $\Delta\varphi_j^{(n)}$ are related by

$$\Delta\varphi_j^{(n)}[\lambda_j(n) + u(n)] = \sum_{l=1}^{j-1} \lambda_l(n) r_{lj}(n) \Delta\varphi_l^{(n)} + \Delta p_j^{(n+1)}. \tag{12}$$

In Lemma 2 below we bound relative perturbations in $\varphi_j(n)$ in terms of relative perturbations in $p(j \mid n+1)$.

Lemma 2. *The perturbations $\Delta p_j^{(n+1)}$ and $\Delta\varphi_j^{(n)}$ satisfy*

$$\max_j \frac{\Delta\varphi_j^{(n)}}{\varphi_j(n)} \le \max_j \frac{\Delta p_j^{(n+1)}}{p(j \mid n+1)}, \qquad \min_j \frac{\Delta\varphi_j^{(n)}}{\varphi_j(n)} \ge \min_j \frac{\Delta p_j^{(n+1)}}{p(j \mid n+1)}.$$

Proof. See Appendix A.4.

Let $\tilde{p}(j \mid n) = p(j \mid n) + \Delta p_j^{(n)}$ be the conditional probabilities at n corresponding to $\tilde{p}(j \mid n+1)$. In (9) $\tilde{p}(j \mid n)$ is expressed as

$$\tilde{p}(j \mid n) = \tilde{\varphi}_j(n)\tilde{\alpha}(n) + \xi_j(n)u(n),$$

where

$$\tilde{\alpha}(n) = \biggl[1 - u(n) \sum_{j=1}^{a} \xi_j(n)\biggr] \bigg/ \sum_{j=1}^{a} \tilde{\varphi}_j(n).$$

The following lemma relates $\Delta p_j^{(n)}$ to $\Delta\varphi_j^{(n)}$.

Lemma 3. *The perturbations $\Delta p_j^{(n)}$ satisfy*

$$\frac{\Delta p_j^{(n)}}{\alpha(n)\varphi_j(n)} = \frac{1}{1 + \Delta\beta^{(n)}} \biggl[\frac{\Delta\varphi_j^{(n)}}{\varphi_j(n)} - \Delta\beta^{(n)}\biggr],$$

where

$$\Delta\beta^{(n)} = \sum_{j=1}^{a} \Delta\varphi_j^{(n)} \bigg/ \sum_{j=1}^{a} \varphi_j(n).$$

Proof. See Appendix A.4.

The preceding lemmas give us the elements to prove the stability of our recurrent algorithm. We consider the following function to assess the magnitude of the relative error in $p(j \mid n)$:

$$g(n) = \frac{\max_j \Delta p_j^{(n)}/p(j \mid n) - \min_j \Delta p_j^{(n)}/p(j \mid n)}{1 + \min_j \Delta p_j^{(n)}/p(j \mid n)}.$$

Theorem 1. *The function $g(n)$ satisfies*

(a) $g(n) \le g(n+1)$,

(b) $\max_j |\Delta p_j^{(n)}/p(j \mid n)| \le g(n)$.

Proof. (a) Because $\sum_{j=1}^{a} \Delta p_j^{(n)} = 0$, we must have

$$\max_j \frac{\Delta p_j^{(n)}}{p(j \mid n)} \ge 0 \quad \text{and} \quad \min_j \frac{\Delta p_j^{(n)}}{p(j \mid n)} \le 0. \tag{13}$$

https://doi.org/10.1239/jap/1331216835 Published online by Cambridge University Press

Using (9) and the results of Lemma 1, we have

$$\max_j \frac{\Delta p_j^{(n)}}{p(j \mid n)} \le \max_j \frac{\Delta p_j^{(n)}}{\alpha(n)\varphi_j(n)} \quad \text{and} \quad \min_j \frac{\Delta p_j^{(n)}}{p(j \mid n)} \ge \min_j \frac{\Delta p_j^{(n)}}{\alpha(n)\varphi_j(n)}.$$

Substituting into the definition of the function $g(n)$ we obtain

$$g(n) \le \frac{\max_j \Delta p_j^{(n)}/(\alpha(n)\varphi_j(n)) - \min_j \Delta p_j^{(n)}/(\alpha(n)\varphi_j(n))}{1 + \min_j \Delta p_j^{(n)}/(\alpha(n)\varphi_j(n))}.$$

From the results of Lemma 2 and Lemma 3, we have

$$\begin{aligned}\left[\max_j \frac{\Delta p_j^{(n)}}{\alpha(n)\varphi_j(n)} - \min_j \frac{\Delta p_j^{(n)}}{\alpha(n)\varphi_j(n)}\right] &= \frac{1}{1+\Delta\beta^{(n)}}\left[\max_j \frac{\Delta\varphi_j^{(n)}}{\varphi_j(n)} - \min_j \frac{\Delta\varphi_j^{(n)}}{\varphi_j(n)}\right] \\ &\le \frac{1}{1+\Delta\beta^{(n)}}\left[\max_j \frac{\Delta p_j^{(n+1)}}{p(j \mid n+1)} - \min_j \frac{\Delta p_j^{(n+1)}}{p(j \mid n+1)}\right]\end{aligned}$$

and

$$\begin{aligned}1 + \min_j \frac{\Delta p_j^{(n)}}{\alpha(n)\varphi_j(n)} &= \frac{1}{1+\Delta\beta^{(n)}}\left[1 + \min_j \frac{\Delta\varphi_j^{(n)}}{\varphi_j(n)}\right] \\ &\ge \frac{1}{1+\Delta\beta^{(n)}}\left[1 + \min_j \frac{\Delta p_j^{(n+1)}}{p(j \mid n+1)}\right] \\ &> 0.\end{aligned}$$

Combining these results yields $g(n) \le g(n+1)$.

(b) Using (13), we conclude that

$$g(n) \ge \max_j \frac{\Delta p_j^{(n)}}{p(j \mid n)} - \min_j \frac{\Delta p_j^{(n)}}{p(j \mid n)} \ge \max_j \left|\frac{\Delta p_j^{(n)}}{p(j \mid n)}\right|.$$

Thus, Theorem 1 shows that our recurrent algorithm is numerically stable.

In the next section we discuss the computational complexity of our method and present a numerical result that illustrates its application.

4. Performance of the method

We start with a brief discussion of the computational complexity of our recurrent solution. In the case of a finite queueing room of size N, i.e. the PH/M/c/N-type queue, the total number of floating point operations can be evaluated in advance. It varies with the precise nature of the phase-type distribution considered, and can be expressed as $o(Na^k)$, where $k \le 3$. For the generalized Coxian distribution where (with the exception of the last stage) $r_{jl} = 0$ for any $l \ne j+1$, we have $k = 1$. Thus, for these types of distribution, which include the Erlang and the hyperexponential distributions, the complexity is $o(Na)$. The maximum value of $k = 3$ is reached for a general acyclic distribution.

In the case of an unrestricted queueing room, i.e. the PH/M/c-type queue, the first step is to solve for the limiting probabilities $\tilde{p}(j)$. This step may involve a slightly variable, but generally modest (say, a few tens), number of bisection points (cf. Appendix A.1). The complexity of each bisection point is $o(a^k)$, where the value of k depends on the type of distribution of the

https://doi.org/10.1239/jap/1331216835 Published online by Cambridge University Press

time between arrivals, as discussed for the PH/M/c/N-type queue. The second step involves our recurrence, and its complexity is $o(\tilde{n}a^k)$.

In both cases, the memory space requirements of our approach are limited: an array of a elements to hold the probabilities $p(j \mid n)$ for a single value of n at a time, two arrays of a elements for the coefficients φ_j and ξ_j, and a single array of $(N + 1)$ (or $(\tilde{n} + 1)$) elements to store the values of $\alpha(n)$.

Note that our recurrence requires no special arrangements for state-dependent systems. Note also that by using the conditional probabilities $p(j \mid n)$, as opposed to the regular state probabilities $p(j, n)$, we partition the state space into a set of subspaces individually normalized for each value of the number of requests n. This has the effect of scaling up the numerical values manipulated, thus delaying the onset of loss of precision due to floating-point underflow problems. Depending on the number of stages and the specific instance of the phase-type distribution, such problems might otherwise occur even for moderate numbers of servers. In Appendix A we give an example of application of our method to a larger number of servers and phases. An implementation of our algorithm is currently available online [4].

In Section 3 we presented a theoretical proof that the proposed recurrence is computationally stable, and, in the many numerical examples we have considered, we found the method to be numerically stable in practice.

5. Conclusions

In this paper we proposed a simple recurrent solution of PH/M/c/N-type and PH/M/c-type queues. The queues considered may have state-dependent arrival and service processes. For systems with finite buffers, the proposed method can handle queues with lost arrivals, as well as with blocking.

We derived our recurrent solution by considering the conditional probabilities of the arrival process given the number of requests in the system. The resulting recurrence yields the state-dependent arrival rates, and, hence, the steady-state probabilities for the number of requests both at arbitrary times and at instants of arrival. The solution is exact, and involves no iteration. In the case of an unrestricted queueing room, the solution involves the computation of the limiting geometric factor ($\tilde{\rho} = \tilde{\alpha}/u^*$) and the limiting conditional probabilities before the actual recurrence. A simple bisection can be used to accomplish this task with high accuracy.

The proposed solution is reasonably scalable as the number of servers and phases increases. For a generalized Coxian distribution of the interarrival times, its computational complexity is $o(Ma)$, and it is $o(Ma^3)$ for an arbitrary acyclic phase-type distribution, where a is the number of phases and M is the maximum population level for the recurrence, i.e. N for the PH/M/c/N-type and $\tilde{n}$ for the PH/M/c-type queue. It is interesting to note that, for phase-type distributions, many distribution fitting methods use the generalized Coxian distribution (or some simplified form thereof) [8], [16], [21], [24], [37], [41], for which the complexity of the proposed recurrent solution grows linearly with the number of stages.

The use of conditional probabilities, as opposed to the joint probabilities of the current arrival phase and the number of customers in the system, has the distinct advantage of dividing the state space into subspaces individually normalized for each population level. This scales up the values manipulated by the computation, thus delaying the onset of numerical issues related to floating-point underflow.

We presented a theoretical proof that our recurrent solution is computationally stable, and our numerical trials indicate that the method is numerically stable in practice even with large numbers of phases. Our recurrent solution requires minimal memory space. It is also very simple to implement in a standard computer language.

https://doi.org/10.1239/jap/1331216835 Published online by Cambridge University Press

Appendix A

A.1. Limiting probabilities with unrestricted queueing room

To devise a simple solution for (8), let us treat the limiting rate of arrivals $\tilde{\alpha}$ as a parameter and write the equation as

$$\tilde{p}(j) = \frac{1}{\lambda_j^* + u^* - \tilde{\alpha}} \left[\sum_{l=1}^{j-1} \lambda_l^* r_{lj}^* \tilde{p}(l) + \tau_j^* u^* \right], \qquad j = 1, \dots, a. \tag{14}$$

For $\tilde{\alpha} < u^* + \min_j \lambda_j^*$, $\tilde{p}(j)$ is a strictly increasing function of $\tilde{\alpha}$. For $\tilde{\alpha} = 0$, summing (14) over all values of j, and dividing by u^*, we obtain

$$\sum_{j=1}^{a} \tilde{p}(j) = 1 - \frac{1}{u^*} \sum_{j=1}^{a} \lambda_j \hat{r}_j^* < 1.$$

On the other hand, as $\tilde{\alpha} \to u^* + \min_j \lambda_j$, $\sum_{j=1}^{a} \tilde{p}(j) \to \infty > 1$. Since $\sum_{j=1}^{a} \tilde{p}(j)$ is a continuous strictly increasing function of $\tilde{\alpha}$, there must be a unique value of $\tilde{\alpha}$ such that $\sum_{j=1}^{a} \tilde{p}(j) = 1$. Given the geometric nature of the limiting distribution of the number of requests in the system, we must have $\tilde{\alpha} < u^*$ if the queue is to be stable. Thus, the limiting solution can be obtained through a simple bisection looking for a value $\tilde{\alpha} \in (0, u^*)$ for which $\sum_{j=1}^{a} \tilde{p}(j) = 1$.

A.2. Application example

To illustrate the application of our method, we consider a system with 32 servers, a Pareto-like distribution of the time between arrivals, and a finite buffer of 64 requests. Such bursty distributions have been reported in I/O subsystems (see, e.g. [22]), as well as computer networks (see, e.g. [14] and [23]). The number of servers chosen might correspond, for instance, to the number of 'exposures' in the case of parallel access volumes [12] in a mainframe I/O subsystem. The phase-type representation of the Pareto-like distribution was obtained using the PhFit software [21]. It contains 16 phases, six of which are used for the heavy-tail part of the distribution. We represent in Figure 1 the results obtained using our method at close to 80% server utilization, including the steady-state average-time probabilities $p(n)$ and the probabilities of the state upon arrival, $P_A(n)$.

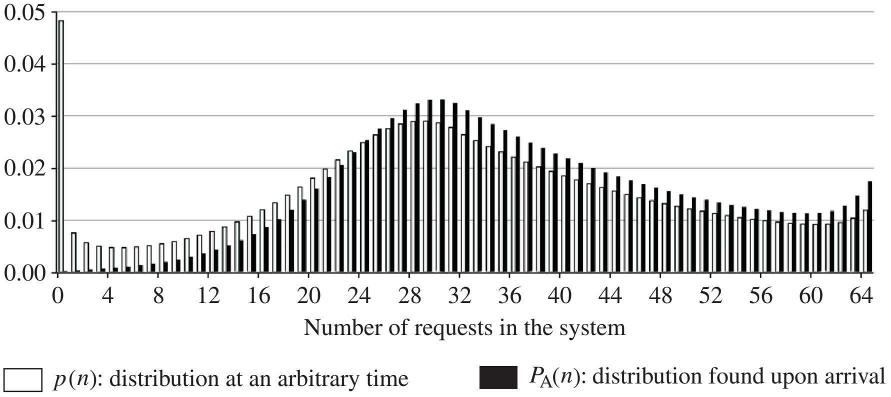

FIGURE 1: Distributions of the number of requests at arbitrary times and upon arrival.

https://doi.org/10.1239/jap/1331216835 Published online by Cambridge University Press

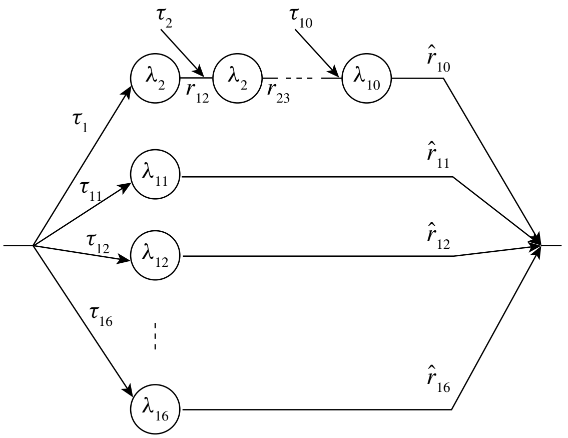

FIGURE 2: Pareto-like distribution for the time between arrivals used in the numerical example.

TABLE 1.

Probabilities		Phase rate	
τ_1	$4.962\,997\,89 \times 10^{-2}$	λ_1	$7.029\,376\,15$
τ_2	$6.556\,227\,66 \times 10^{-2}$	λ_2	$5.602\,915\,23$
τ_3	$6.080\,485\,26 \times 10^{-2}$	λ_3	$5.014\,299\,42$
τ_4	$3.953\,060\,23 \times 10^{-2}$	λ_4	$4.398\,542\,22$
τ_5	$8.433\,362\,70 \times 10^{-2}$	λ_5	$4.111\,812\,39$
τ_6	$1.114\,454\,42 \times 10^{-1}$	λ_6	$3.091\,590\,50$
τ_7	$4.346\,580\,02 \times 10^{-2}$	λ_7	$2.594\,319\,15$
τ_8	$1.137\,791\,44 \times 10^{-2}$	λ_8	$2.526\,358\,99$
τ_9	$3.875\,069\,20 \times 10^{-2}$	λ_9	$2.456\,465\,93$
τ_{10}	$2.301\,720\,16 \times 10^{-1}$	λ_{10}	$2.259\,072\,87$
τ_{11}	$5.270\,715\,06 \times 10^{-7}$	λ_{11}	$2.234\,017\,04 \times 10^{-5}$
τ_{12}	$8.119\,158\,05 \times 10^{-6}$	λ_{12}	$1.691\,547\,94 \times 10^{-4}$
τ_{13}	$1.084\,296\,20 \times 10^{-4}$	λ_{13}	$9.825\,538\,80 \times 10^{-4}$
τ_{14}	$1.424\,765\,17 \times 10^{-3}$	λ_{14}	$5.498\,235\,48 \times 10^{-3}$
τ_{15}	$1.867\,582\,84 \times 10^{-2}$	λ_{15}	$3.059\,037\,19 \times 10^{-2}$
τ_{16}	$2.447\,091\,29 \times 10^{-1}$	λ_{16}	$1.700\,401\,80 \times 10^{-1}$
$r_{12}, r_{23}, \ldots = \hat{r}_{10}, \hat{r}_{11}, \ldots = 1$			

The Pareto-like distribution used in our example is depicted in Figure 2, with the numerical values of the probabilities and phase rates given in Table 1.

A.3. Details of the recurrent computation

We look for a solution to (3) of the form $p(j \mid n) = \varphi_j(n)\alpha(n)+\xi_j(n)u(n)$. For our purpose here, we let $p(j \mid N+1) = \tau_j$ for all j. Equation (3) then implies that

$$\varphi_j(n) = \frac{1}{[\lambda_j(n)+u(n)]}\left\{p(j \mid n+1) + \sum_{l=1}^{j-1} \lambda_l(n) r_{lj}(n) \varphi_l(n)\right\}, \qquad j = 1, \ldots, a, \quad (15)$$

and

$$\xi_j(n) = \frac{1}{[\lambda_j(n) + u(n)]}\left\{\sum_{l=1}^{j-1} \lambda_l(n) r_{lj}(n) \xi_l(n) + \tau_j(n)\right\}, \qquad j = 1, \dots, a. \tag{16}$$

The unknown $\alpha(n)$ is determined from the normalizing condition (5):

$$\alpha(n) = \left[1 - u(n)\sum_{j=1}^{a} \xi_j(n)\right] \Big/ \sum_{j=1}^{a} \varphi_j(n). \tag{17}$$

The computation for $n = N-1$ in the case of communication blocking proceeds in an analogous way.

Note that we are able to use a simple recurrence owing to the assumed acyclic nature of the phase-type distribution of the time between arrivals. If this restriction is lifted, the $\varphi_j(n)$ and $\xi_j(n)$ could not be evaluated sequentially as shown in (15) and (16).

Our recurrent computation can be summarized as follows.

Algorithm 1.

```
for $n$ from $N$ to 0 do
    • Compute coefficients $\varphi_j(n)$ and $\xi_j(n)$ for $j$ from 1 to $a$.
    • Compute $\alpha(n)$ using (17).
    • Compute $p(j \mid n)$ for $j$ from 1 to $a$ using (6).
endfor
Compute $p(n)$ and $P_{\mathrm{A}}(n)$ using (1) and (7).
```

A.4. Proofs of Lemmas 2 and 3

Proof of Lemma 2. Suppose that $\max_j \Delta\varphi_j^{(n)}/\varphi_j(n)$ is attained at $j = m$. Then, for $j = m$, we can write (12) as

$$\left[\frac{\Delta\varphi_m^{(n)}}{\varphi_m(n)}\right]\varphi_m(n)[\lambda_m(n) + u(n)] - \sum_{l=1}^{m-1}\left[\frac{\Delta\varphi_l^{(n)}}{\varphi_l(n)}\right]\varphi_l(n)\lambda_l(n) r_{lm}(n)$$
$$= \left[\frac{\Delta p_m^{(n+1)}}{p(m \mid n+1)}\right] p(m \mid n+1),$$

which leads to

$$\left[\frac{\Delta\varphi_m^{(n)}}{\varphi_m(n)}\right]\left\{\varphi_m(n)[\lambda_m(n) + u(n)] - \sum_{l=1}^{m-1} \varphi_l(n)\lambda_l(n) r_{lm}(n)\right\} \le \left[\frac{\Delta p_m^{(n+1)}}{p(m \mid n+1)}\right] p(m \mid n+1).$$

On the other hand, at $j = m$, (10) becomes

$$\varphi_m(n)[\lambda_m(n) + u(n)] - \sum_{l=1}^{m-1} \varphi_l(n)\lambda_l(n) r_{lm}(n) = p(m \mid n+1).$$

https://doi.org/10.1239/jap/1331216835 Published online by Cambridge University Press

Combining these two results yields

$$\max_j \frac{\Delta\varphi_j^{(n)}}{\varphi_j(n)} = \frac{\Delta\varphi_m^{(n)}}{\varphi_m(n)} \le \frac{\Delta p_m^{(n+1)}}{p(m \mid n+1)} \le \max_j \frac{\Delta p_j^{(n+1)}}{p(j \mid n+1)}.$$

In an analogous manner, we obtain

$$\min_j \frac{\Delta\varphi_j^{(n)}}{\varphi_j(n)} \ge \min_j \frac{\Delta p_j^{(n+1)}}{p(j \mid n+1)}.$$

Proof of Lemma 3. We have

$$\begin{aligned}
\Delta\alpha(n) &= \tilde{\alpha}(n) - \alpha(n)\\
&= \frac{1 - u(n)\sum_{j=1}^{a}\xi_j(n)}{\sum_{j=1}^{a}[\varphi_j(n) + \Delta\varphi_j^{(n)}]} - \frac{1 - u(n)\sum_{j=1}^{a}\xi_j(n)}{\sum_{j=1}^{a}\varphi_j(n)}\\
&= \frac{-\big(1 - u(n)\sum_{j=1}^{a}\xi_j(n)\big)\big(\sum_{j=1}^{a}\Delta\varphi_j^{(n)}\big)}{\big(\sum_{j=1}^{a}\varphi_j(n)\big)\big(\sum_{j=1}^{a}\varphi_j(n) + \sum_{j=1}^{a}\Delta\varphi_j^{(n)}\big)}\\
&= -\alpha(n)\frac{\Delta\beta^{(n)}}{1 + \Delta\beta^{(n)}},\\
\Delta p_j^{(n)} &= \tilde{p}(j \mid n) - p(j \mid n)\\
&= [\varphi_j(n) + \Delta\varphi_j^{(n)}][\alpha(n) + \Delta\alpha(n)] - \alpha(n)\varphi_j(n)\\
&= \alpha(n)\Delta\varphi_j^{(n)} + \Delta\alpha(n)[\varphi_j(n) + \Delta\varphi_j^{(n)}]\\
&= \alpha(n)\left[\Delta\varphi_j^{(n)} - \frac{\Delta\beta^{(n)}}{1 + \Delta\beta^{(n)}}[\varphi_j(n) + \Delta\varphi_j^{(n)}]\right]\\
&= \frac{\alpha(n)}{1 + \Delta\beta^{(n)}}[\Delta\varphi_j^{(n)} - \Delta\beta^{(n)}\varphi_j(n)].
\end{aligned}$$

A.5. PH/M/*c*/*N* queue with blocking

We consider here a communications type of blocking where the arrival process is stopped altogether when the number of requests in the system becomes N. The arrival process then restarts after the departure of a request. For this type of blocking, the equations for the top two values of n change. For $n = N$, there are no arrivals, the probabilities $p(j \mid N)$ are meaningless, and $\alpha(N) = 0$ (of course, $p(N)$ is in general nonzero.) For $n = N - 1$, we obtain

$$\begin{aligned}
&p(j \mid N-1)[\lambda_j(N-1) + u(N-1)]\\
&\quad = \sum_{l=1}^{j-1}\lambda_l(N-1)r_{lj}(N-1)p(l \mid N-1)\\
&\qquad + \tau_j(N-2)u(N-1) + \tau_j(N-1)\alpha(N-1), \qquad j = 1, \dots, a. \qquad (18)
\end{aligned}$$

The steady-state probabilities $p(n)$ are obtained, as before, from (1). Note that (18) is similar to (4) except that it applies to $n = N - 1$. For $n = 0, \dots, N - 2$, (3) applies as before. The probabilities upon arrival are given by (7). The probability that the source is blocked is simply $p(N)$.

https://doi.org/10.1239/jap/1331216835 Published online by Cambridge University Press

The blocking described above corresponds either to a closed network of two stations with a total population of N requests, or a node in a computer network with a buffer size of N packets where the source generating the packets is prevented from overflowing the buffer.

References

[1] Akar, N. and Sohraby, K. (1997). An invariant subspace approach in M/G/1 and G/M/1 type Markov chains. *Comm. Statist. Stoch. Models* **13,** 381–416.

[2] Allen, A. O. (1990). *Probability, Statistics, and Queueing Theory*, 2nd edn. Academic Press, Boston.

[3] Bean, N. G., Pollett, P. K. and Taylor, P. G. (2000). Quasistationary distributions for level-dependent quasi-birth-and-death processes. *Comm. Statist. Stoch. Models* **16,** 511–541.

[4] Begin, T. and Brandwajn, A. (2011). Solutions to queueing sytems. Available at http://queueing-systems.ens-lyon.fr.

[5] Bini, D. A., Latouche, G. and Meini, B. (2005). *Numerical Methods for Structured Markov Chains*. Oxford University Press.

[6] Bini, D. A., Meini, B., Steffé, S. and Van Houdt, B. (2006). Structured Markov chain solver: algorithms. In Proc. SMCtools '06, ACM, New York, 6 pp.

[7] Bini, D. A., Meini, B., Steffé, S. and Van Houdt, B. (2006). Structured Markov chain solver: software tools. In Proc. SMCtools '06, ACM, New York, 10 pp.

[8] Bobbio, A., Horváth, A. and Telek, M. (2005). Matching three moments with minimal acyclic phase type distributions. *Stoch. Models* **21,** 303–326.

[9] Bolch, G., Greiner, S., Meer, H. and Trivedi, K. (2005). *Queueing Networks and Markov Chains*, 2nd edn. Wiley-Interscience, Hoboken, NJ.

[10] Brandwajn, A. and Wang, H. (2008). A conditional probability approach to M/G/1-like queues. *Performance Evaluation* **65,** 366–381.

[11] Bright, L. W. and Taylor, P. G. (1995). Calculating the equilibrium distribution in level dependent quasi-birth-and-death processes. *Comm. Statist. Stoch. Models* **11,** 497–525.

[12] Castets, G. A., Leplaideur, D., Bras, J. A. and Galang, J. (2001). *IBM Enterprise Storage Server* (IBM Publ. SG24-5665). IBM Corporation.

[13] Chaudhry, M. L., Gupta, U. C. and Goswami, V. (2004). On discrete-time multiserver queues with finite buffer: GI/Geom/m/N. *Comput. Operat. Res* **31,** 2137–2150.

[14] Crovella, M. E. and Bestavros, A. (1997). Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Trans. Networking* **5,** 835–846.

[15] Elwalid, A. I., Mitra, D. and Stem, T. E. (1991). Statistical multiplexing of Markov modulated sources: theory and computational algorithms. In *Proc. ITC-13* (Copenhagen, 1991), pp. 495–500.

[16] Feldmann, A. and Whitt, W. (1997). Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. In INFOCOM '97, Vol. 3 (Kobe, Japan, April 1997), pp. 1096–1104.

[17] Gaver, D. P., Jacobs, P. A. and Latouche, G. (1984). Finite birth-and-death models in randomly changing environments. *Adv. Appl. Prob* **16,** 715–731.

[18] Harris, C. M., Brill, P. H. and Fischer, M. J. (2000). Internet-type queues with power-tailed interarrival times and computational methods for their analysis. *INFORMS J. Comput.* **12,** 261–271.

[19] Haverkort, B. R. and Ost, A. (1997). Steady-state analysis of infinite stochastic petri nets: comparing the spectral expansion and the matrix-geometric method. In *Proc. 7th Internat. Workshop Petri Nets and Performance Models* (France, June 1997), pp. 36–45.

[20] Hokstad, P. (1975). The G/M/m queue with finite waiting room. *J. Appl. Prob.* **12,** 779–792

[21] Horváth, A. and Telek, M. (2002). PhFit: a general phase-type fitting tool. In *Computer Performance Evaluation: Modelling Techniques and Tools* (Lecture Notes Comput. Sci. **2324**), eds T. Field *et al.*, Springer, Berlin, pp. 82–91.

[22] Hsu, W. W. and Smith, A. J. (2003). Characteristics of I/O traffic in personal computer and server workloads. *IBM Systems J.* **42,** 347–372.

[23] Jiang, H. and Dovrolis, C. (2005). Why is the internet traffic bursty in short time scales? In *ACM SIGMETRICS Performance Evaluation Review* (Proc. SIGMETRICS 2005), ACM, New York, pp. 241–252.

[24] Khayari, R. A., Sadre, R. and Haverkort, B. R. (2003). Fitting world-wide web request traces with the EM-algorithm. *Performance Evaluation* **52,** 175–191.

[25] Kiefer, J. and Wolfowitz, J. (1955). On the theory of queues with many servers. *Trans. Amer. Math. Soc.* **78,** 1–18.

[26] Kleinrock, L. (1975). *Queueing Systems*, Vol. I. John Wiley, New York.

[27] Kleinrock, L. (1976). *Queueing Systems*, Vol. II. John Wiley, New York.

https://doi.org/10.1239/jap/1331216835 Published online by Cambridge University Press

[28] Latouche, G. (1994). Newton's iteration for non-linear equations in Markov chains. *IMA J. Numerical Anal.* **14,** 583–598.

[29] Latouche, G. and Ramaswami, V. (1993). A logarithmic reduction algorithm for quasi-birth-and-death processes. *J. Appl. Prob.* **30,** 650–674.

[30] Latouche, G. and Ramaswami, V. (1999). *Introduction to Matrix Analytic Methods in Stochastic Modeling*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

[31] Mitrani, I. and Chakka, R. (1995). Spectral expansion solution for a class of Markov models: application and comparison with the matrix-geometric method. *Performance Evaluation* **23,** 241–260.

[32] Mitrani, I. and Mitra, D. (1992). A spectral expansion method for random walks on semi-infinite strips. In *Iterative Methods in Linear Algebra*, North-Holland, Amsterdam, pp. 141–149.

[33] Naoumov, V., Krieger, U. R. and Wagner, D. (1996). Analysis of a multi-server delay-loss system with a general Markovian arrival process. In *Matrix-Analytic Methods in Stochastic Models*, eds S. Chakravarthy and A. S. Alfa, Marcel Dekker, New York, pp. 43–66.

[34] Neuts, M. F. (1989). *Structured Stochastic Matrices of M/G/1 Type and Their Applications*. Marcel Dekker, New York.

[35] Neuts, M. F. (1994). *Matrix-Geometric Solutions in Stochastic Models*. Dover Publications, New York.

[36] O'Cinneide, C. A. (1990). Characterization of phase-type distributions. *Comm. Statist. Stoch. Models* **6,** 1–57.

[37] Osogami, T. and Harchol-Balter, M. (2006). Closed form solutions for mapping general distributions to quasi-minimal PH distributions. *Performance Evaluation* **63,** 524–552.

[38] Ramaswami, V. and Taylor, P. G. (1996). Some properties of the rate operators in level dependent quasi-birth-and-death processes with a countable number of phases. *Comm. Statist. Stoch. Models* **12,** 143–164.

[39] Riska, A. and Smirni, E. (2002). M/G/1-type Markov processes: a tutorial. In *Performance Evaluation of Complex Systems: Techniques and Tools* (Lecture Notes Comput. Sci. **2459**) eds M. Calzarossa and S. Tucci, Springer, London, pp. 36–63.

[40] Scheller-Wolf, A. and Sigman, K. (1997). Delay moments for FIFO GI/GI/s queues. *Queueing Systems* **25,** 77–95.

[41] Thummler, A., Buchholz, P. and Telek, M. (2006). A novel approach for phase-type fitting with the EM algorithm. *IEEE Trans. Dependable Secure Comput* **3,** 245–258.

[42] Tran, H. T. and Do, T. V. (2000). Computational aspects for steady state analysis of QBD processes. *Periodica Polytechnica Electrical Eng.* **44,** 179–200.

[43] Vangal, S. R. *et al.* (2008). An 80-tile sub-100-W TeraFLOPS processor in 65 nm CMOS. *IEEE J. Solid-State Circuits* **43,** 29–41.

[44] Ye, Q. (2002). On Latouche–Ramaswami's logarithmic reduction algorithm for quasi-birth-and-death processes. *Stoch. Models* **18,** 449–467.

https://doi.org/10.1239/jap/1331216835 Published online by Cambridge University Press