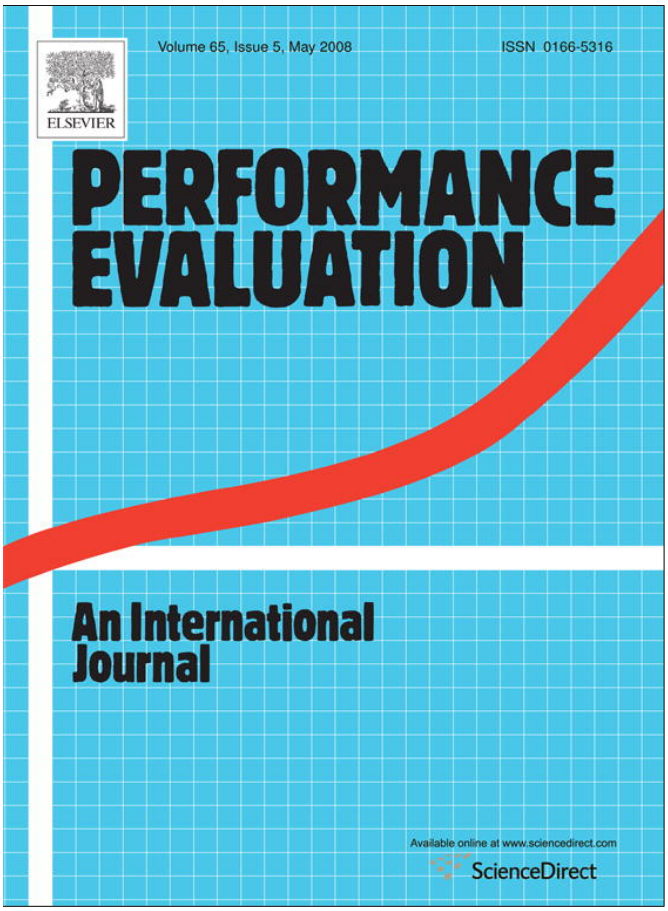

Available online at www.sciencedirect.com
ScienceDirect
Performance Evaluation 65 (2008) 366–381
PERFORMANCE EVALUATION
An International Journal
www.elsevier.com/locate/peva

A conditional probability approach to $M/G/1$-like queues

Alexandre Brandwajn[a,*], Hongyun Wang[b]

[a] Department of Computer Engineering, Baskin School of Engineering, University of California, Santa Cruz, CA 95064, USA
[b] Department of Applied Mathematics & Statistics, Baskin School of Engineering, University of California, Santa Cruz, CA 95064, USA

Received 29 August 2006; received in revised form 5 May 2007; accepted 20 September 2007
Available online 5 October 2007

Abstract

Following up on a recently renewed interest in computational methods for $M/G/1$-type processes, this paper considers an $M/G/1$-like system in which the service time distribution is represented by a Coxian series of memoryless stages. We present a novel approach to the solution of such systems. Our method is based on conditional probabilities, and provides a simple, computationally efficient and stable approach to the evaluation of the steady-state queue length distribution. We provide a proof of the numerical stability of our method. Without explicit use of matrix-geometric techniques or stochastic complementation, we are able to handle systems with state-dependent service and arrival rates. The proposed approach can be used to compute the queue length distribution for both finite and infinite $M/G/1$-like queues. In the case of an infinite, state-independent queue, our method allows us to show using elementary tools that the queue length distribution is asymptotically geometric. The parameter of the asymptotic geometric can be expressed through a simple set of equations, easily solved using fixed point iteration. Our approach is very thrifty in terms of memory requirements, easy to implement, and generally fast. Numerical examples illustrate the performance of the proposed method.

Keywords: $M/G/1$-like queues; Coxian distribution; State-dependent service; Conditional probability; Queue length distribution; Recurrent solution; Numerical stability; Computational efficiency; Asymptotic geometric distribution; Algorithms; Performance

1. Introduction

$M/G/1$-type queues are an essential tool in the performance analysis of computers and computer networks. Devices ranging from volumes in a storage subsystem to stations in an optical ring network have been represented as $M/G/1$ queues. Oftentimes, it is necessary to take into account the finite buffer space at such devices (e.g. to properly size the buffers), and, in some systems, the fact that the service rate may depend on the number of customers in the system (e.g. due to contention for shared resources). The efficient and numerically stable solution of such queues, in particular, the calculation of the stationary queue length probabilities, is therefore of importance. Recently, there has been a renewed interest in computational methods for $M/G/1$-type processes [27,30]. As discussed by Stathopoulos et al. [30], matrix analytic techniques introduced by Neuts [23,24] are often used to evaluate such processes. In particular, Ramaswami's algorithm [25,21] provides a numerically stable approach to the computation of steady-state probabilities in an $M/G/1$-type system. This algorithm is based on stochastic complementation [28], and its fastest implementation uses the FFT [22] (cf. [30]). The ETAQA method [27] has been proposed as an alternative for the

* Corresponding author. Tel.: +1 831 459 4023.
E-mail addresses: alexb@soe.ucsc.edu (A. Brandwajn), hongwang@ams.ucsc.edu (H. Wang).

0166-5316/$ - see front matter
doi:10.1016/j.peva.2007.09.002

computation of the moments of the queue length distribution (but not the distribution itself). Stathopoulos et al. [30] derive a new formulation for ETAQA and demonstrate its links with the Ramaswami's method.

The large body of previous work includes the work by Gaver et al. [14], who consider finite queues in randomly changing environments, and derive a numerical method involving recursive determination of certain matrices. Bright and Taylor [10] extend the logarithmic reduction algorithm proposed by Latouche and Ramaswami [18] to level-dependent infinite queues. They note possible numerical problems when recursively calculating matrices involved in the solution. Such problems are also mentioned in Gaver et al. [14]. Latouche shows in [17] that Newton's method applied to non-linear equations in Markov chains is quadratically convergent although not very attractive because of its computational complexity. Bini and Meini [6] extend the cyclic reduction techniques to infinite block matrices, and later propose an improvement based on FFT to lower computational cost and increase numerical stability [7]. Akar and Sohraby [1] propose an invariant subspace approach with at least quadratic convergence rates and potentially improved accuracy due to the avoidance of truncation. Ramaswami and Taylor [26] study the generalization of matrix-geometric stationary distribution to level-dependent quasi-birth-and-death processes. More recently, Bean et al. [3] consider quasistationary distributions for level-dependent processes and their computation using methods akin to the Latouche–Ramaswami algorithm. Ye [31] studies the theoretical properties of the Latouche–Ramaswami logarithmic reduction algorithm, pointing out numerical stability issues and offering an alternative based on a more stable algorithm for inverting a diagonally dominant matrix.

The asymptotic behavior of an infinite queue has been studied, among others, by Bean et al. [4] who consider quasi-birth-and-death processes with decomposable spaces, and use the matrix-geometric form of the steady state probability vector to derive a numeric method for computing the "caudal characteristic factor" of the process using matrices smaller than earlier traditional methods. Knessl et al. [15] consider a state-dependent $M/G/1$ where the interarrival and the service time distributions depend on the amount of unfinished work in the system. They apply perturbation methods to derive approximations for several measures pertaining to the unfinished work and the mean busy period in such a queue. In a related work, Knessl et al. [16] study a $GI/G/1$ queue with a similar form of state dependence using the same perturbation methods.

The reader is referred to the books by Latouche and Ramaswami [19], and by Bini et al. [5] for an overview of properties of matrix analytic approaches and numerical methods for quasi-birth-and-death problems and $M/G/1$-type Markov chains. Practical considerations and software implementation for several of the methods mentioned above are discussed by Bini et al. [8,9].

In this paper, we consider an $M/G/1$-like system in which the service time distribution is represented by a Coxian [12] series of memoryless stages. It is well known that a Coxian distribution can approximate arbitrarily closely any distribution, and a two-stage Coxian can be used to match the first two moments of any distribution [2] whose coefficient of variation is greater than 1. Several authors have considered algorithms for matching an arbitrary distribution by a Coxian, e.g. [11,29,13,20].

Our method is based on conditional probabilities, and provides a simple, computationally efficient and numerically stable approach to the evaluation of the steady-state queue length distribution. Without any explicit use of matrix-geometric techniques or stochastic complementation, we are able to solve systems with state-dependent service and arrival rates. As a result, the proposed approach can be used for finite $M/G/1$-like queues as well. In the case of an infinite, state-independent queue, the form of the queue length distribution indicates that it is asymptotically geometric with a coefficient given as a solution of a simple set of equations, easily solved via fixed point iteration. To the best of our knowledge, the proposed method is novel.

This paper is organized as follows. In Section 2 we describe in more detail the queue considered and present our computational approach. Section 3 is devoted specifically to the case of an infinite state-independent queue and its asymptotically geometric queue length distribution. In Section 4 we outline the proof of numerical stability of our method. Section 5 presents numerical results to illustrate the behavior of our method, and Section 6 concludes this paper. Additionally, in the Appendix we show that the conditional probability on which we base our approach converges to a single fixed point in the case of an infinite state-independent queue.

2. $M/G/1$-like model and its recurrent solution

2.1. Model considered and its equations

The single-server queuing model considered is shown in Fig. 1.

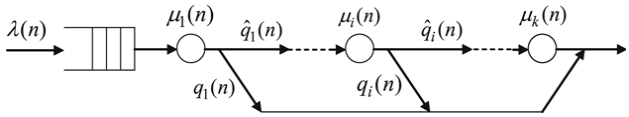

Fig. 1. $M/G/1$-type queue with Coxian-like service.

We assume that customers arrive from a quasi-Poisson source with rate $\lambda(n)$ where n denotes the current number of customers in the system. The service of a customer is represented by a state-dependent Coxian-like distribution [12] with a total of k stages. Both the completion rates of the memoryless stages and the exit probabilities in this distribution may depend on the number of customers in the queue. We denote by $\mu_i(n)$ the service rate of stage i and by $q_i(n)$ the probability that the customer completes its service following stage i, $i = 1, \ldots, k$. We let $\hat{q}_i(n) = 1 - q_i(n)$ denote the probability that the customer proceeds to stage $i + 1$ upon completion of stage i when there are n customers in the system. We assume that $\mu_i(n) > 0$, $i = 1, \ldots, k$, $0 < \hat{q}_i(n) \leq 1$ for $i < k$ and $\hat{q}_k(n) = 0$. Note that our formulation allows not only the mean value but also the nature of the service time distribution to change as the number of customers in the queue changes.

We consider the system of Fig. 1 in steady state, and we use the couple (n, i) where n $(n \geq 1)$ is the current number of customers and i $(i = 1, \ldots, k)$ is the current stage of service to describe the state of the system when it is not idle. We denote by $p(n, i)$ the corresponding steady state probability, and by $p(0)$ the probability that there are no customers in the system.

It is straightforward to obtain the balance equations for the corresponding state probabilities

$$p(1, 1)[\lambda(n) + \mu_1(n)] = \sum_{i=1}^{k} p(n + 1, i)\mu_i(n + 1)q_i(n + 1) + p(0)\lambda(0) \tag{1}$$

$$p(1, i)[\lambda(n) + \mu_i(n)] = p(n, i - 1)\mu_{i-1}(n)\hat{q}_{i-1}(n), \quad i > 1 \tag{2}$$

for $n = 1$, and

$$p(n, 1)[\lambda(n) + \mu_1(n)] = \sum_{i=1}^{k} p(n + 1, i)\mu_i(n + 1)q_i(n + 1) + p(n - 1, 1)\lambda(n - 1) \tag{3}$$

$$p(n, i)[\lambda(n) + \mu_i(n)] = p(n, i - 1)\mu_{i-1}(n)\hat{q}_{i-1}(n) + p(n - 1, i)\lambda(n - 1), \quad i > 1 \tag{4}$$

for $n > 1$.

These balance equations can be transformed into equations for the conditional probabilities of the service stage number given the current number of customers, and solved as a simple recurrence.

2.2. Recurrent solution

Our goal is to transform the standard balance equations (1)–(4) into easy-to-evaluate recurrence relations. We denote by $p(i \mid n)$ the conditional probability that the current service stage is i given that the number of customers is n, and by $p(n)$ the marginal probability that the number of customers in our $M/G/1$-like queue is equal to n. For all $n \geq 1$ and $i = 1, \ldots, k$ we have

$$p(n, i) = p(i \mid n)p(n). \tag{5}$$

From (5) and the balance equations (1)–(4), it is easy to see that $p(n)$ can be expressed as

$$p(n) = \frac{1}{G} \prod_{l=1}^{n} \lambda(l - 1)/u(l), \tag{6}$$

where G is a normalizing constant and

$$u(n) = \sum_{i=1}^{k} p(i \mid n)\mu_i(n)q_i(n) \tag{7}$$

$u(n)$ is the conditional rate of completion given n, and $1/u(n)$ can be viewed as the mean service time given that there are n customers in the system.

Substituting (5) and (6) into the balance equations, and using the fact that $p(n) > 0$ for all feasible values of n, we readily obtain the equations for the conditional probabilities $p(i \mid n)$

$$p(1 \mid 1)[\lambda(1) + \mu_1(1)] = \lambda(1) + u(1) \tag{8}$$

$$p(i \mid 1)[\lambda(1) + \mu_i(1)] = p(i-1 \mid 1)\mu_{i-1}(1)\hat{q}_{i-1}(1), \quad i > 1 \tag{9}$$

for $n = 1$, and

$$p(1 \mid n)[\lambda(n) + \mu_1(n)] = \lambda(n) + p(1 \mid n-1)u(n) \tag{10}$$

$$p(i \mid n)[\lambda(n) + \mu_i(n)] = p(i-1 \mid n)\mu_{i-1}(n)\hat{q}_{i-1}(n) + p(i \mid n-1)u(n), \quad i > 1 \tag{11}$$

for $n > 1$.

Since we must have

$$\sum_{i=1}^{k} p(i \mid n) = 1 \quad \text{for } n \geq 1, \tag{12}$$

we can express the solution of (8) and (9) as

$$p(i \mid n = 1) = \frac{1}{H} \prod_{j=2}^{i} \mu_{j-1}(1)\hat{q}_{j-1}(1)/[\lambda(1) + \mu_j(1)], \tag{13}$$

where H is a normalizing constant. The empty product is equal to 1 by convention, so that we have $p(1 \mid 1) = 1/H$. Note that Eqs. (8) and (9), together with the normalizing condition (12), imply that we must have $u(1) < \mu_1(1)$ in the steady state of our system.

We now consider Eqs. (10) and (11) for increasing values of $n = 2, 3, \ldots$. For each n, the $p(i \mid n-1)$ are known from the preceding step, so that (10) and (11) is a system of k equations for the k conditional probabilities $p(i \mid n), i = 1, \ldots, k$. We note that Eqs. (10) and (11) involve $u(n)$, a linear combination of the conditional probabilities $p(i \mid n)$, and we can express the latter as

$$p(i \mid n) = b_i(n)\lambda(n) + c_i(n)u(n). \tag{14}$$

From (10) we have

$$b_1(n) = 1/[\lambda(n) + \mu_1(n)], \quad \text{and} \quad c_1(n) = p(1 \mid n-1)/[\lambda(n) + \mu_1(n)]. \tag{15}$$

Eq. (11) yields the following recurrence

$$\begin{aligned} b_i(n) &= b_{i-1}(n)\mu_{i-1}(n)\hat{q}_{i-1}(n)/[\lambda(n) + \mu_i(n)], \\ c_i(n) &= [c_{i-1}(n)\mu_{i-1}(n)\hat{q}_{i-1}(n) + p(i \mid n-1)]/[\lambda(n) + \mu_i(n)] \quad \text{for } i > 1. \end{aligned} \tag{16}$$

Having computed the coefficients $b_i(n)$ and $c_i(n)$ from the above recurrence, we determine $u(n)$ from the normalizing condition (12)

$$u(n) = \left[1 - \lambda(n) \sum_{i=1}^{k} b_i(n)\right] \Big/ \sum_{i=1}^{k} c_i(n). \tag{17}$$

For $n = 1$, we can use a similar approach or the solution form given by (13). Note that the recurrence itself (15) and (16) involves no subtractions to detract from numerical stability. A formal proof of stability is outlined in Section 4. Space requirements of our recurrent computation are very modest since we need only one set (for a single value of n) of the conditional probabilities $p(i \mid n)$ at any time. Even the values of $u(n)$ need not be kept if we embed our recurrence into the computation of the marginal probability $p(n)$. The latter is given by the familiar product-form formula (6) identical to that of a state-dependent $M/M/1$ queue with arrival rate $\lambda(n)$ and service rate $u(n)$. Clearly, if $\lambda(n)$ vanishes for some value of n, the customer population is limited and the recurrence has a "natural" stopping

point. If the population in the queue is unrestricted, as would be the case with a Poisson source, the computation will have to stop when values of $p(i \mid n)$ (and thus $u(n)$) for consecutive values of n no longer change more than some reasonable tolerance we fix as the convergence criterion. To the best of our knowledge, the proposed approach to the computation of the stationary queue length probabilities in an $M/G/1$-like queue is novel. The next section is devoted to the particular case of an infinite state-independent queue.

3. Infinite state-independent queue

We now restrict our attention to the particular case of an $M/G/1$-like queue with Poisson arrivals and standard Coxian service distribution. Specifically, we have $\lambda(n) = \lambda$, $\mu_i(n) = \mu_i$, and $q_i(n) = q_i$. Since the approach presented in the preceding section involves a recurrent computation of the conditional probabilities $p(i \mid n)$ for consecutive values of n, it might seem that, for the open queue, the method would require the solution of an infinite number of such recurrences. In this section we argue that, in practice, these conditional probabilities tend to quickly reach a limiting value, so that only a relatively small number of recurrences need to be solved. In the process, we demonstrate the asymptotically geometric form of the steady-state distribution $p(n)$ for large n.

For the queue considered, the equations for the conditional probability for $n = 1$ become

$$p(1 \mid n = 1)[\lambda + \mu_1] = \lambda + u(1) \tag{18}$$

$$p(i \mid n = 1)[\lambda + \mu_i] = p(i - 1 \mid n = 1)\mu_{i-1}\hat{q}_{i-1}, \quad i > 1, \tag{19}$$

and for $n > 1$ we have

$$p(1 \mid n)[\lambda + \mu_1] = \lambda + p(1 \mid n - 1)u(n) \tag{20}$$

$$p(i \mid n)[\lambda + \mu_i] = p(i - 1 \mid n)\mu_{i-1}\hat{q}_{i-1} + p(i \mid n - 1)u(n), \quad i > 1. \tag{21}$$

As n tends to infinity, the conditional probability $p(i \mid n)$ and $u(n)$ tend to their limiting values which we denote by $\tilde{p}(i)$ and $\tilde{u}$, respectively. A rigorous proof of this convergence is presented in the Appendix. Given the form of $p(n)$ in (6), the convergence to limiting values means that we have asymptotically for large n,

$$p(n + 1)/p(n) \approx \lambda/\tilde{u}, \tag{22}$$

i.e., the steady-state distribution is asymptotically geometric.

In our numerical experiments, we found that, typically, the solution of Eqs. (20) and (21) tends rapidly to the limiting distribution $\tilde{p}(i)$ as n increases. Consequently, we can use the following computational approach to obtain the queue length distribution for the infinite state-independent queue. Select a convergence criterion, e.g. $\|p(i \mid n) - p(i \mid n-1)\| < \varepsilon$ with some reasonable value of ε, and let $\tilde{n}$ be the value of n for which the convergence is attained. We can express the steady-state distribution $p(n)$ as

$$p(n) \approx \frac{1}{G} \begin{cases} \prod_{l=1}^{n} \lambda/u(l), & n \le \tilde{n} \\ \left[\prod_{l=1}^{\tilde{n}} \lambda/u(l)\right] (\lambda/\tilde{u})^{n-\tilde{n}}, & n > \tilde{n}. \end{cases} \tag{23}$$

The normalizing constant G can be written as

$$G = 1 + \sum_{n=1}^{\tilde{n}-1} \prod_{l=1}^{n} \lambda/u(l) + \left[\prod_{l=1}^{\tilde{n}} \lambda/u(l)\right] \frac{1}{1 - (\lambda/\tilde{u})}.$$

Thus, we use the recurrence described in Section 2.2 (13)–(17) for $n = 1, \ldots, \tilde{n}$ (i.e. simply until convergence has been reached in the computation of $p(i \mid n)$ and $u(n)$). Once convergence has been attained, we use the last computed value for $u(n)$ as the limiting value $\tilde{u}$ in the geometric tail of the distribution $p(n)$. With this approach, the expected

number of customers in the system can be written as

$$\bar{n} \approx \frac{1}{G}\left\{\sum_{n=1}^{\tilde{n}} np(n) + \left[\frac{\tilde{n}}{1-(\lambda/\tilde{u})} + \frac{(\lambda/\tilde{u})}{[1-(\lambda/\tilde{u})]^2}\right]\prod_{l=1}^{\tilde{n}} \lambda/u(l)\right\}.$$

Clearly, the system is stable as long as $\lambda < \tilde{u}$.

In practice, it has been our experience that convergence to $\tilde{u}$ tends occur quite rapidly, resulting typically in moderate values for $\tilde{n}$. In Section 5, we present numerical results to illustrate the behavior of our method.

Our recurrent algorithms produce the set of $u(n)$ for $n = 1, \ldots, \tilde{n}$, i.e., until convergence to $\tilde{u}$. If desired, the limiting value $\tilde{u}$ can also be computed independently as follows. The limiting conditional probabilities $\tilde{p}(i)$ and the limiting service rate value $\tilde{u}$ satisfy the following equations

$$\tilde{p}(1)[\lambda + \mu_1] = \lambda + \tilde{p}(1)\tilde{u} \tag{24}$$

$$\tilde{p}(i)[\lambda + \mu_i] = \tilde{p}(i-1)\mu_{i-1}\hat{q}_{i-1} + \tilde{p}(i)\tilde{u}, \quad i > 1. \tag{25}$$

Eqs. (24) and (25) can be easily (and, generally, rapidly) solved using a fixed-point iteration. Starting with an initial probability distribution $\tilde{p}^0(i)$ (we use a superscript to denote the iteration number) and the corresponding value $\tilde{u}^0$, we compute values at iteration j

$$\tilde{p}^j(1) \propto [\lambda + \tilde{p}^{j-1}(1)\tilde{u}^{j-1}]/[\lambda + \mu_1] \tag{26}$$

$$\tilde{p}^j(i) \propto [\tilde{p}^j(i-1)\mu_{i-1}\hat{q}_{i-1} + \tilde{p}^{j-1}(i)\tilde{u}^{j-1}]/[\lambda + \mu_i], \quad i > 1. \tag{27}$$

At each iteration we normalize the values so as to have $\sum_{i=1}^{k} \tilde{p}^j(i) = 1$. This allows us to determine the limiting values $\tilde{p}(i)$ and $\tilde{u}$. The latter is the service rate value for the asymptotically geometric state probability. We do not have a formal proof of convergence of this iterative scheme. In practice, it has converged without any damping in the iteration for all the cases investigated.

The next section is devoted to a formal proof that our recurrent algorithm is numerically stable.

4. Stability of recurrent solution

Our goal in this section is to show that our recurrent solution described in Section 2 is computationally stable. For our purposes here, we find it convenient to rewrite the recurrence as

$$p(i \mid n)[\lambda(n) + \mu_i(n)] = p(i-1 \mid n)\mu_{i-1}(n)\hat{q}_{i-1}(n) + \lambda(n)\delta_{i,1} + p(i \mid n-1)u(n), \tag{28}$$

where we use the following notation

$$p(i \mid 0) = \begin{cases} 1, & i = 1 \\ 0, & i > 1, \end{cases} \qquad p(0 \mid n) = 0 \quad \text{and} \quad \delta_{i,1} = \begin{cases} 1, & i = 1 \\ 0, & i \neq 1. \end{cases}$$

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

$$p(i \mid n) = b_i(n)\lambda(n) + c_i(n)u(n) \tag{29}$$

with the coefficients $b_i(n)$ and $c_i(n)$ given by

$$b_i(n)[\lambda(n) + \mu_i(n)] = b_{i-1}(n)\mu_{i-1}(n)\hat{q}_{i-1}(n) + \delta_{i,1} \tag{30}$$

$$c_i(n)[\lambda(n) + \mu_i(n)] = c_{i-1}(n)\mu_{i-1}(n)\hat{q}_{i-1}(n) + p(i \mid n-1). \tag{31}$$

By convention, we let $b_0(n) = c_0(n) = 0$. $u(n)$ is given by (17), i.e.

$$u(n) = \left[1 - \lambda(n)\sum_{i=1}^{k} b_i(n)\right] \Big/ \sum_{i=1}^{k} c_i(n).$$

We assume that the Coxian service time distribution has indeed k stages, so that

$$\mu_i(n) > 0, \quad i = 1, \ldots, k, \qquad 0 < \hat{q}_i(n) \leq 1 \quad \text{for } i < k \quad \text{and} \quad \hat{q}_k(n) = 0.$$

We start by showing that our recurrence for $p(i \mid n)$ produces positive values.

Lemma 1. *If* $p(1 \mid n-1) > 0$ *and* $p(i \mid n-1) \geq 0$ *for* $i > 1$, *then we have*

(1) $b_i(n) > 0$ *and* $c_i(n) > 0$ *for* $i \geq 1$,
(2) $u(n) > 0$, *and*
(3) $p(i \mid n) > 0$ *for* $i \geq 1$.

Proof. (1) Follows directly from (30) and (31).
(2) Summing (30) over all values of i and using the fact that $\hat{q}_k(n) = 0$, we have

$$\sum_{j=1}^{k} b_j(n)[\lambda(n) + \mu_j(n)] = \sum_{j=1}^{k} b_j(n)\mu_j(n)\hat{q}_j(n) + 1.$$

Rearranging the terms and using the fact that $q_j(n) = 1 - \hat{q}_j(n)$, we get

$$1 - \lambda(n)\sum_{j=1}^{k} b_j(n) = \sum_{j=1}^{k} b_j(n)\mu_j(n)q_j(n) > 0, \qquad \text{and hence } u(n) > 0.$$

(3) Follows directly from the results of (1) and (2).

Lemma 1 tells us that $p(i \mid n) > 0$ for $i \geq 1$ and $n \geq 1$. □

We now consider a set of perturbed conditional probabilities (the perturbation corresponding to floating point round-off errors)

$$\tilde{p}(i \mid n-1) = p(i \mid n-1) + \Delta p_i^{(n-1)}.$$

Since the conditional probability is normalized at each step of the recurrence, the perturbations $\Delta p_i^{(n-1)}$ must satisfy $\sum_{i=1}^{k} \Delta p_i^{(n-1)} = 0$. We also assume that the perturbations are small so that we have $\tilde{p}(i \mid n-1) > 0$. (30) shows that the perturbation does not affect $b_i(n)$. Let $\tilde{c}_i(n) = c_i(n) + \Delta c_i^{(n-1)}$ be the solution of (31) when $p(i \mid n-1)$ is replaced by $\tilde{p}(i \mid n-1)$. $\Delta p_i^{(n-1)}$ and $\Delta c_i^{(n)}$ are related by

$$\Delta c_i^{(n)}[\lambda(n) + \mu_i(n)] = \Delta c_{i-1}^{(n)}\mu_{i-1}(n)\hat{q}_{i-1}(n) + \Delta p_i^{(n-1)}. \tag{32}$$

In Lemma 2 we bound relative perturbations in $p_i(n-1)$ in terms of relative perturbations in $c_i(n)$.

Lemma 2. $\Delta p_i^{(n-1)}$ *and* $\Delta c_i^{(n)}$ *satisfy*

$$\max_j \frac{\Delta c_j^{(n)}}{c_j(n)} \leq \max_j \frac{\Delta p_j^{(n-1)}}{p(j \mid n-1)}$$
$$\min_j \frac{\Delta c_j^{(n)}}{c_j(n)} \geq \min_j \frac{\Delta p_j^{(n-1)}}{p(j \mid n-1)}.$$

Proof. See Appendix. □

Let $\tilde{p}(i \mid n) = p(i \mid n) + \Delta p_i^{(n)}$ be the conditional probabilities at n corresponding to $\tilde{p}(i \mid n-1)$. $\tilde{p}(i \mid n)$ is expressed in (29) as

$$\tilde{p}(i \mid n) = b_i(n)\lambda(n) + \tilde{c}_i(n)\tilde{u}(n),$$

where

$$\tilde{u}(n) = \left[1 - \lambda(n)\sum_{i=1}^{k} b_i(n)\right] \Big/ \sum_{i=1}^{k} \tilde{c}_i(n).$$

The next lemma relates $\Delta p_i^{(n)}$ to $\Delta c_i^{(n)}$.

Lemma 3. *$\Delta p_i^{(n)}$ satisfies*

$$\frac{\Delta p_i^{(n)}}{u(n)c_i(n)} = \frac{1}{1+\Delta\beta^{(n)}}\left[\frac{\Delta c_i^{(n)}}{c_i(n)} - \Delta\beta^{(n)}\right],$$

where

$$\Delta\beta^{(n)} = \sum_{j=1}^{k} \Delta c_j^{(n)} \Big/ \sum_{j=1}^{k} c_j(n).$$

Proof. See Appendix. □

We now have the elements to prove the stability of our recurrent algorithm. Consider a function "measuring" the magnitude of the relative error in $p(i \mid n)$

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

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

(1) $g(n) \le g(n-1)$

(2) $\max_j |\frac{\Delta p_j^{(n)}}{p(j|n)}| \le g(n)$.

Proof. (1) $\sum_{j=1}^{k} \Delta p_j^{(n)} = 0$ implies

$$\max_j \frac{\Delta p_j^{(n)}}{p(j \mid n)} \ge 0 \quad \text{and} \quad \min_j \frac{\Delta p_j^{(n)}}{p(j \mid n)} \le 0. \tag{33}$$

Using (29) and the results of Lemma 1, we have

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

Substituting into function $g(n)$ yields

$$g(n) \le \frac{\max_j \frac{\Delta p_j^{(n)}}{u(n)c_j(n)} - \min_j \frac{\Delta p_j^{(n)}}{u(n)c_j(n)}}{1 + \min_j \frac{\Delta p_j^{(n)}}{u(n)c_j(n)}}.$$

From the results of Lemmas 2 and 3 it follows

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

and

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

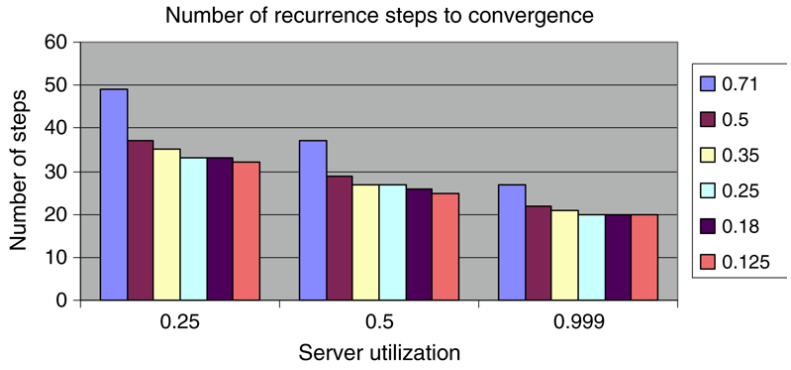

Fig. 2. Speed of convergence for Erlang-k service distribution.

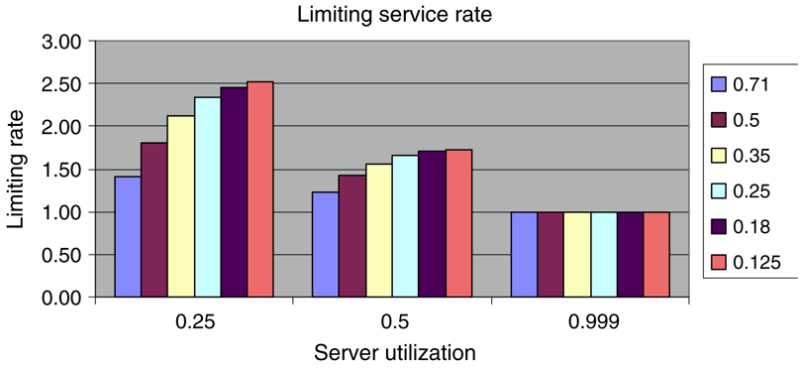

Fig. 3. Limiting service rate for Erlang-k service distribution.

Combining these results we get $g(n) \leq g(n-1)$.

(2) Using (33), we conclude that

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

Theorem 1 demonstrates that our recurrent algorithm is numerically stable. □

In the next section we present numerical results that illustrate the computational behavior of our method.

5. Numerical results

We start by illustrating the behavior of our method in the case of infinite state-independent queues. Unless specified otherwise, the convergence criterion used in our results for infinite queues was $|1 - u(n)/u(n-1)| < \varepsilon$ for all $n \geq 1$ with $\varepsilon = 10^{-15}$. In all examples presented in this section, the mean service time was kept at 1. Fig. 2 shows the number of recurrence steps before convergence is reached when the service time is an Erlang distribution of order k for $k = 2$, 4, 8, 16, 32, and 64. More specifically, we take $\mu_1 = \cdots = \mu_k = k$ and $\hat{q}_1 = \cdots = \hat{q}_{k-1} = 1$. The corresponding coefficient of variation of the Erlang distribution is $1/\sqrt{k}$, i.e., 0.71, 0.5, 0.35, 0.25, 0.18, and 0.125 in our example. We consider three values of server utilization: 0.25, 0.5, and 0.999, the latter value corresponds to a system that is barely ergodic. In Fig. 3, we present the corresponding value of the service rate at convergence, i.e., the limiting value $\tilde{u}$ (in the asymptotically geometric state probability distribution $p(n)$). Figures are labeled with the coefficient of variation of the service time distribution.

We observe that, in this example, convergence occurs within a few tens of recurrence steps. This seems to be the case for many systems. Interestingly, the value of the limiting service rate $\tilde{u}$ is generally very different from the inverse of the expected service time, and only approaches the latter near server saturation.

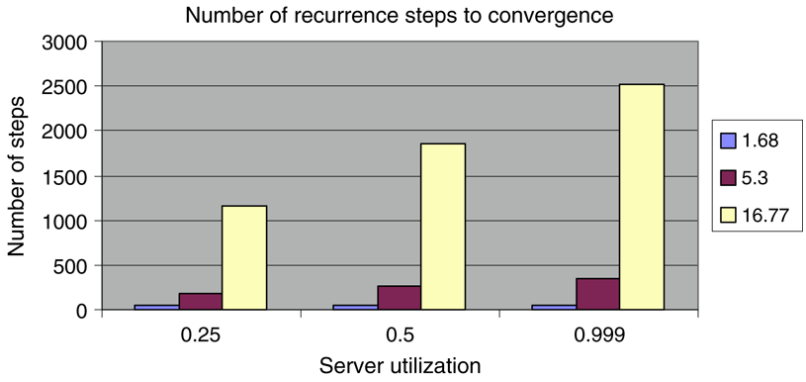

Fig. 4. Example of slower convergence.

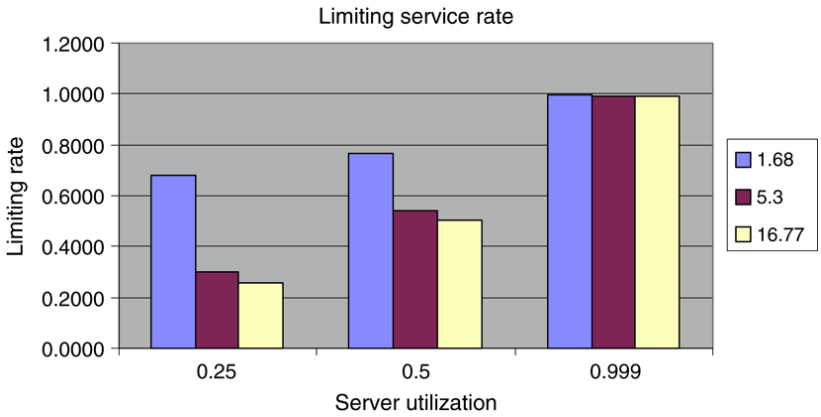

Fig. 5. Limiting service rate for slower convergence case.

Although convergence to the limiting service rate value tends to occur rapidly for many systems, there are cases when considerably more recurrence steps are needed to achieve the stringent convergence level we used. Fig. 4 shows the number of recurrence steps in one of the worst cases we have encountered. The Coxian service time distribution used in this example corresponds to the following parameters: $k = 9$, $\mu_1 = 2$, $\mu_2 = \cdots \mu_k = 2(k-1)\hat{q}_1$, $\hat{q}_2 = \cdots = \hat{q}_{k-1} = 1$, and the values of $\hat{q}_1$ are 0.1, 0.01 and 0.001. The resulting coefficient of variation of the service time distribution is 1.68, 5.30, and 16.77, respectively, and we use it to label the results. In Fig. 5, we have represented the corresponding limiting service rates.

We observe that the number of iteration steps needed to satisfy the specified convergence criterion reaches 2500 in this case.

Note that although the number of recurrence steps in this example increases with the coefficient of variation of the service time distribution, other high variability distributions exhibit much faster convergence. Figs. 6 and 7 show the results for a two-stage Coxian with $\mu_1 = 2$ and the following sets of parameter values for the second stage and the probability that stage 2 will follow the completion of stage 1:

set 1: $\mu_2 = 0.3, \hat{q}_1 = 0.15$;
set 2: $\mu_2 = 0.1, \hat{q}_1 = 0.05$;
set 3: $\mu_2 = 0.01, \hat{q}_1 = 0.05$;
set 4: $\mu_2 = 0.001, \hat{q}_1 = 0.0005$.

The resulting coefficient of variation is 1.83, 3.16, 10, and 31.62, respectively.

Here we observe that the number of recurrence steps remains below one hundred even for the highest variability case and very close to server saturation. As before, the limiting service rate is quite different from the inverse of the mean service time (1, in our case), except close to saturation. Interestingly, for the high variability distribution

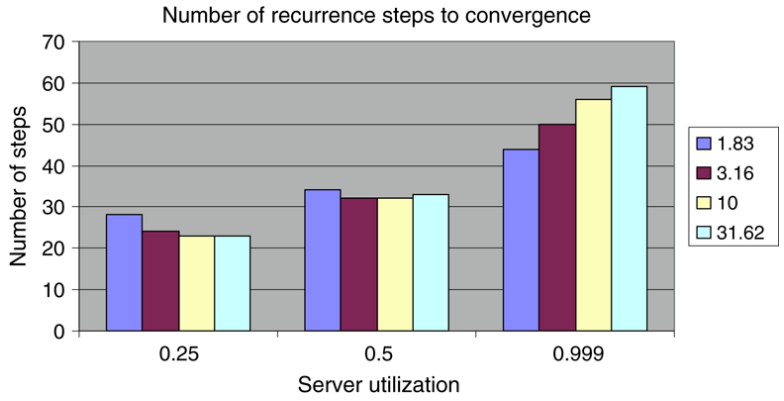

Fig. 6. Speed of convergence for high variability distribution.

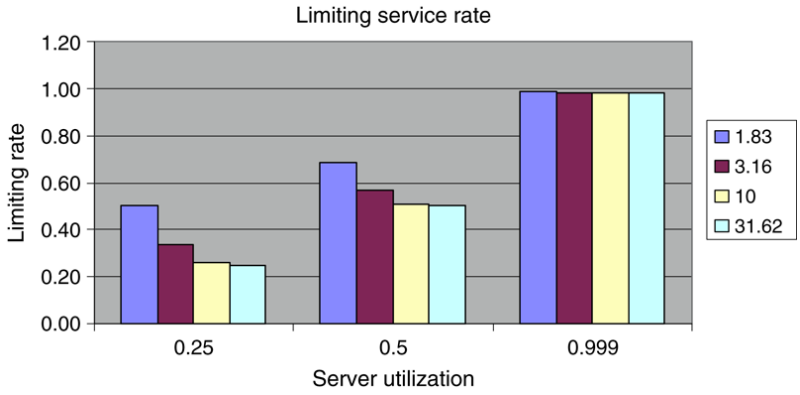

Fig. 7. Limiting service rate for high variability distribution.

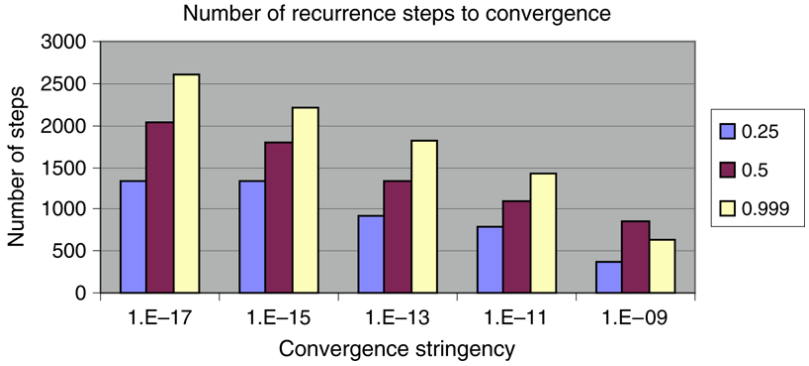

Fig. 8. Convergence speed versus convergence stringency.

the limiting service rate value $\tilde{u}$ is lower than the inverse of the mean service time (unlike for the Erlang service distribution).

While the value of ε in the convergence criterion used in our examples yields accurate results both for the expected number in the system and for selected state probabilities, it may seem rather stringent. In our next example, we study the influence of the convergence stringency on the number of recurrence steps. The results shown in Fig. 8 have been obtained for a service time distribution similar to that used in Figs. 4 and 5, except that the total number of stages in the Coxian distribution is 7. The coefficient of variation of the service time distribution is 17.08. The results in Fig. 8 are labeled by the server utilization.

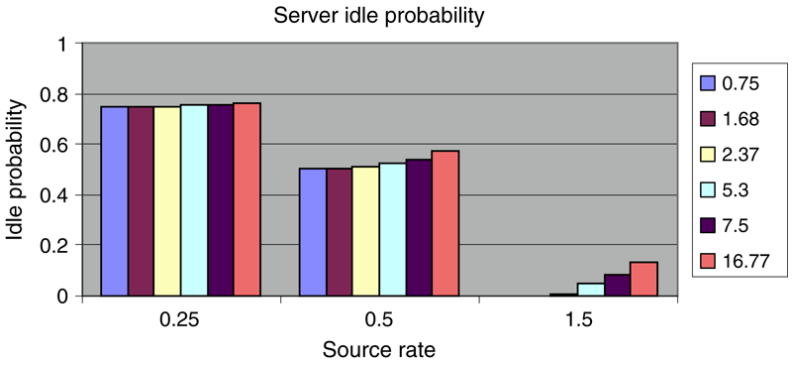

Fig. 9. Expected number of customers in finite source system.

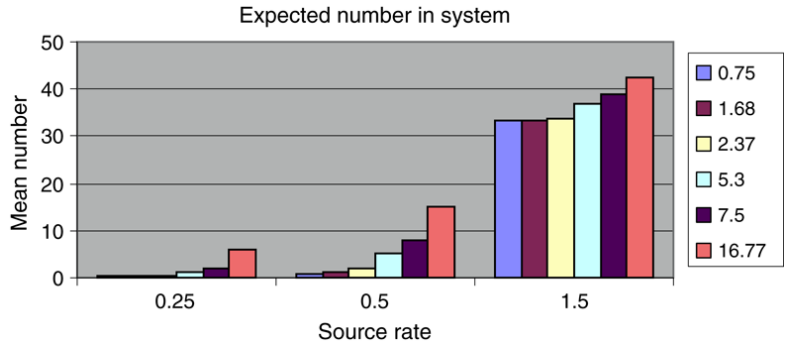

Fig. 10. Probability server is idle in finite source system.

We notice that, at higher server utilization levels, the convergence stringency may have an important effect on the number of recurrence steps required. For the convergence criterion used in these examples, $|1 - u(n)/u(n-1)| < \varepsilon$ for all $n \geq 1$, it has been our experience that $\varepsilon = 10^{-13}$ is sufficient to achieve accurate results for most systems with server utilization of up to 0.98. For server utilizations not exceeding 0.7, $\varepsilon = 10^{-11}$ tends to be sufficient. Note that for many systems the number of recurrence steps to convergence is quite low even with more stringent values for ε, and the solution time in all cases is negligible. The computational effort at each recurrence step is minimal. Note also that the storage requirements of our approach are very low. If we embed the recurrence steps inside the computation of state probabilities $p(n)$ for increasing values of n, there is no need to store more than the last set of the conditional probabilities $p(i \mid n)$ and the last two values of the service rate $u(n)$. It is also sufficient to store only the values of $p(n)$ (and any derived quantities) one is interested in. Additionally, we note that, even in the cases where a larger number of recurrence steps is needed, we have not encountered numerical problems. Since we know the limiting service rate $\tilde{u}$ from the solution of Eqs. (24) and (25), we can double check for any loss of accuracy when the recurrent computation of $u(n)$ ends due to convergence.

As a last example, we consider a system with state-dependent arrivals where $\lambda(n) = (N - n)\alpha$. This form of the arrival rate corresponds to a set of N customers each generating a request with rate α. The results shown in Figs. 9 and 10 pertain to a system with $N = 100$ and the nine-stage high variability service time distribution used in Figs. 4 and 5. In this finite-source example, we consider more values for $\hat{q}_1$: 0.5, 0.1, 0.05, 0.01, 0.005, and 0.001. The resulting coefficient of variation for the service time distribution ranges from 0.75 to 16.77, and is used as a label in Figs. 8 and 9.

Fig. 9 displays the expected number of customers in the system as a function of the maximum arrival rate $N\alpha$. Fig. 10 shows the corresponding probability that the server is idle in such a system versus the same maximum arrival rate.

It is interesting to note the effect of service time variability on the expected number of customers in the system, as well as on the probability that the system is idle. As the coefficient of variation of the service time distribution increases so does the probability that the server is idle. This effect becomes more visible as the maximum arrival rate

increases. The expected number of customers increases with service time variability but the relative increase with service time variability becomes smaller as the arrival rate increases.

Overall, in all the examples we studied, we have found the proposed method to be numerically well behaved, and generally very fast. Additionally, our method is quite easy to implement in a language like C, C++, Java or FORTRAN, with minimal memory space requirements.

6. Conclusion

We have proposed an approach, believed to be novel, to the solution of $M/G/1$-like queues in which the service time distribution is represented by a Coxian series of memoryless stages. Our method is based on elementary conditional probabilities, and provides a simple, computationally efficient and stable recurrence that allows us to obtain state-dependent service rates, and, hence, the steady-state queue length distribution. Unlike existing methods, the proposed approach does not rely on the use of matrix-geometric techniques or stochastic complementation. It can handle generalized queues in which both the arrival rates and the parameters of the service time distribution, including the probabilities of moving from stage to stage, may be state dependent. Our method can be applied to both finite and infinite $M/G/1$-like queues. We believe that our method combines conceptual simplicity, ease of implementation and computational efficiency.

In the case of an infinite, state-independent queue, the explicit product form of the queue length distribution in our method allows us to show that it is asymptotically geometric. Additionally, we have given a simple set of equations that defines the limiting service rate for this geometric distribution, and we have presented a simple fixed-point iteration to solve it.

We have provided a proof of the numerical stability of our algorithm in the general case, and a proof of its convergence to the unique non-negative fixed point in the case of a state-independent infinite queue. We have included several examples to illustrate the performance of our method in the case of an infinite $M/G/1$-like queue. We have considered both low and high service time variability, and server utilizations ranging from low to very close to saturation. Our results for infinite queues suggest that, in general, only a moderate number of recurrence steps is needed to reach the limiting service rate with high accuracy. In many cases, just a few tens of recurrence steps may be sufficient. We have also included an example to illustrate the results obtained with our method in the case of a finite number of customers and state-dependent arrivals.

The proposed method is in general fast, very thrifty in terms of memory space requirements, conceptually simple, and easy to implement. It is stable even for systems with server utilization very close to saturation. Its simplicity and numeric stability should make it attractive to performance analysts. An extension of this approach to $M/G/1$-like systems with service interruptions is under study.

Acknowledgments

The authors wish to thank the anonymous referees for their remarks and suggestions which helped improve this paper.

Appendix A

A.1. Proof of lemmas

Proof of Lemma 2. Suppose $\max_j \frac{\Delta c_j^{(n)}}{c_j(n)}$ is attained at $j = m$. At $i = m$, we write (32) as

$$\left[\frac{\Delta c_m^{(n)}}{c_m(n)}\right] c_m(n)[\lambda(n) + \mu_m(n)] - \left[\frac{\Delta c_{m-1}^{(n)}}{c_{m-1}(n)}\right] c_{m-1}(n)\mu_{m-1}(n)\hat{q}_{m-1}(n) = \left[\frac{\Delta p_m^{(n-1)}}{p(m \mid n-1)}\right] p(m \mid n-1)$$

which leads to

$$\left[\frac{\Delta c_m^{(n)}}{c_m(n)}\right] \{c_m(n)[\lambda(n) + \mu_m(n)] - c_{m-1}(n)\mu_{m-1}(n)\hat{q}_{m-1}(n)\} \leq \left[\frac{\Delta p_m^{(n-1)}}{p(m \mid n-1)}\right] p(m \mid n-1).$$

On the other hand, at $i = m$, (31) becomes

$$c_m(n)[\lambda(n) + \mu_m(n)] - c_{m-1}(n)\mu_{m-1}(n)\hat{q}_{m-1}(n) = p(m \mid m-1).$$

Combining these two results, we obtain

$$\max_j \frac{\Delta c_j^{(n)}}{c_j(n)} = \frac{\Delta c_m^{(n)}}{c_m(n)} \le \frac{\Delta p_m^{(n-1)}}{p(m \mid n-1)} \le \max_j \frac{\Delta p_j^{(n-1)}}{p(j \mid n-1)}.$$

Using an analogous approach, we can show that

$$\min_j \frac{\Delta c_j^{(n)}}{c_j(n)} \ge \min_j \frac{\Delta p_j^{(n-1)}}{p(j \mid n-1)}. \quad \square$$

Proof of Lemma 3.

$$\begin{aligned}
\Delta u(n) &= \tilde{u}(n) - u(n) \\
&= \frac{1 - \lambda(n)\sum_{j=1}^{k} b_j(n)}{\sum_{j=1}^{k}[c_j(n) + \Delta c_j^{(n)}]} - \frac{1 - \lambda(n)\sum_{j=1}^{k} b_j(n)}{\sum_{j=1}^{k} c_j(n)} \\
&= \frac{-\left(1 - \lambda(n)\sum_{j=1}^{k} b_j(n)\right)\left(\sum_{j=1}^{k} \Delta c_j^{(n)}\right)}{\left(\sum_{j=1}^{k} c_j(n)\right)\left(\sum_{j=1}^{k} c_j(n) + \sum_{j=1}^{k} \Delta c_j^{(n)}\right)} = -u(n)\frac{\Delta\beta^{(n)}}{1 + \Delta\beta^{(n)}} \\
\Delta p_i^{(n)} &= \tilde{p}(i \mid n) - p(i \mid n) \\
&= [u(n) + \Delta u(n)]\,[c_i(n) + \Delta c_i^{(n)}] - u(n)c_i(n) \\
&= u(n)\Delta c_i^{(n)} + \Delta u(n)[c_i(n) + \Delta c_i^{(n)}] \\
&= u(n)\left[\Delta c_i^{(n)} - \frac{\Delta\beta^{(n)}}{1 + \Delta\beta^{(n)}}[c_i(n) + \Delta c_i^{(n)}]\right] = \frac{u(n)}{1 + \Delta\beta^{(n)}}[\Delta c_i^{(n)} - \Delta\beta^{(n)} c_i(n)]. \quad \square
\end{aligned}$$

Proof of Lemma 4. Using (29) and the fact that $\frac{p(j|n) - \lambda b_j}{p(j|n)} \le 1 - \lambda \min_j b_j = \alpha$, we obtain

$$\max_j \frac{\Delta p_j^{(n)}}{p(j \mid n)} = \max_j \left(\frac{p(j \mid n) - \lambda b_j}{p(j \mid n)} \cdot \frac{\Delta p_j^{(n)}}{u(n)c_i(n)}\right) \le \alpha \max_j \frac{\Delta p_j^{(n)}}{u(n)c_i(n)}$$

$$\min_j \frac{\Delta p_j^{(n)}}{p(j \mid n)} = \min_j \left(\frac{p(j \mid n) - \lambda b_j}{p(j \mid n)} \cdot \frac{\Delta p_j^{(n)}}{u(n)c_i(n)}\right) \ge \alpha \min_j \frac{\Delta p_j^{(n)}}{u(n)c_i(n)}.$$

Substituting into function $g(n)$ yields

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

Then $g(n) \le \alpha g(n-1)$ follows from the proof of Theorem 1. $\square$

A.2. Convergence to fixed point for infinite state-independent queue

Consider now the infinite state-independent queue of Section 3. For an infinite state-independent queue, all parameters are independent of n. As a results, the coefficients b_i defined in (30) are also independent of n. Let $\alpha = 1 - \lambda \min_j b_j$. From Lemma 1 we get $0 < \alpha < 1$, and we have

Lemma 4. *For an infinite state-independent queue, $g(n)$ satisfies*

$$g(n) \leq \alpha g(n-1).$$

Proof of Lemma 4. See Appendix A.1. □

For notational convenience, we let $\vec{p}(n) = (p(1 \mid n), p(2 \mid n), \ldots, p(k \mid n))^T$. We have

Theorem 2. *The sequence $\{\vec{p}(n),\ n = 1, 2, \ldots, \}$ converges to a unique fixed point.*

Proof of Theorem 2. The key to the proof is to view $\vec{p}(n+1)$ as $\vec{p}(n)$ perturbed:

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

Using the results of Theorem 1 and Lemma 4, we have

$$\|\vec{p}(n+1) - \vec{p}(n)\|_\infty = \max_j |\Delta p_j^{(n)}| \leq \max_j \left|\frac{\Delta p_j^{(n)}}{p(j \mid n)}\right| \leq g(n) \leq \alpha^n g(1)$$

and

$$\|\vec{p}(N+m) - \vec{p}(N)\|_\infty \leq \sum_{n=N}^{N+m-1} \|\vec{p}(n+1) - \vec{p}(n)\|_\infty \leq \left(\sum_{n=N}^{N+m-1} \alpha^n\right) g(1) \leq \alpha^N \frac{g(1)}{1-\alpha}.$$

Thus, $\{\vec{p}(n),\ n = 1, 2, \ldots, \}$ is a Cauchy sequence with respect to the infinity norm, and converges with respect to this norm to a limit $\vec{p}(\infty)$. Since $\vec{p}(\infty)$ is the limit of both $\vec{p}(n)$ and $\vec{p}(n+1)$, it must be a fixed point. □

Now we show that our recurrent algorithm has only one non-negative fixed point. Suppose $\vec{p}(n) = (p(1),\ p(2), \ldots, p(k))^T$ is a fixed point of our recurrent algorithm. Substituting into (28), we have

$$p(i)(\lambda + \mu_i) = p(i-1)\mu_{i-1}\hat{q}_{i-1} + \lambda\delta_{i,1} + p(i)u, \quad \text{where } u = \sum_{j=1}^{k} p(j)\mu_j q_j. \tag{34}$$

Let us treat u as a parameter and write $p(i)$ as

$$p(i) = [p(i-1)\mu_{i-1}\hat{q}_{i-1} + \lambda\delta_{i,1}]/(\lambda + \mu_i - u).$$

To ensure that $p(i)$ is non-negative, we must have $u < \min_j(\lambda + \mu_j)$. It is clear that for $u < \min_j(\lambda + \mu_j)$, $p(i)$ is a strictly increasing function of u. When $u = 0$, summing (34) over all values of i and dividing by λ yields $\sum_{i=1}^k p(i) = 1 - \frac{1}{\lambda}\sum_{i=1}^k p(i)\mu_i q_i < 1$. As $u \to \min_j(\lambda + \mu_j)$, $\sum_{i=1}^k p(i) \to +\infty > 1$. Since $\sum_{i=1}^k p(i)$ is a continuous and strictly increasing function of u, there is a unique value of u such that $\sum_{i=1}^k p(i) = 1$ is satisfied. This means that the non-linear equation (34) has a unique non-negative solution.

References

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

[2] A.O. Allen, Probability, Statistics, and Queuing Theory, 2nd edition, Academic Press, 1990.

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

[4] N.G. Bean, J.-M. Li, P.G. Taylor, Caudal characteristics of QBDs with decomposable phase spaces, in: G. Latouche, P.G. Taylor (Eds.), Advances in Algorithmic Methods for Stochastic Models, Notable Publications, NJ, 2000, pp. 37–56.

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

[6] D. Bini, B. Meini, On the solution of a nonlinear matrix equation arising in queueing problems, SIAM J. Matrix Anal. Appl. 17 (1996) 906–926.
[7] D.A. Bini, B. Meini, Improved cyclic reduction for solving queueing problems, Numer. Algorithms 15 (1997) 57–74.
[8] D.A. Bini, B. Meini, S. Steffè, B. van Houdt, Structured Markov chain solver: The algorithms, in: Proceedings of the SMCTOOLS Workshop, Pisa, 2006.
[9] D.A. Bini, B. Meini, S. Steffè, B. van Houdt, Structured Markov chain solver: Software tools, in: Proceedings of the SMCTOOLS Workshop, Pisa, 2006.
[10] L.W. Bright, P.G. Taylor, Calculating the equilibrium distribution in level dependent quasi-birth-and-death processes, Commun. Statist. Stochastic Models 11 (1995) 497–525.
[11] W. Bux, U. Herzog, The phase concept: Approximation of measured data and performance analysis, in: Proceedings of the International Symposium on Computer Performance Modeling, Measurement and Evaluation, North Holland, Yorktown Heights, NY, 1977, pp. 23–38.
[12] D.R. Cox, Walter L. Smith, Queues, John Wiley, New York, 1961.
[13] M. Faddy, Penalised maximum likelihood estimation of the parameters in a coxian phase-type distribution, in: Matrix-Analytic Methods: Theory and Application: Proceedings of the Fourth International Conference, Adelaide, Australia, 2002, pp. 107–114.
[14] P. Gaver, A. Jacobs, G. Latouche, Finite birth-and-death models in randomly changing environments, Adv. Appl. Probab. 16 (1984) 715–731.
[15] C. Knessl, B. Matkowsky, Z. Schuss, C. Tier, Asymptotic analysis of a state-dependent $M/G/1$ queueing system, SIAM J. Appl. Math. 46 (1986) 483–505.
[16] C. Knessl, C. Tier, B. Matkowsky, Z. Shuss, A state-dependent $GI/G/1$ queue, Eur. J. Appl. Math. 5 (1994) 217–241.
[17] G. Latouche, Newton's iteration for non-linear equations in Markov chains, IMA J. Numer. Anal. 14 (1994) 583–598.
[18] G. Latouche, V. Ramaswami, A logarithmic reduction algorithm for quasi-birth-and-death processes, J. Appl. Probab. 30 (1993) 650–674.
[19] G. Latouche, V. Ramaswami, Introduction to matrix analytic methods in stochastic modeling, in: ASA-SIAM Series on Statistics and Applied Probability, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.
[20] S. McLean, M. Faddy, P. Millard, Using Markov models to assess the performance of a health and community care system, in: Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems, 2006, pp. 777–782.
[21] B. Meini, Solving $M/G/1$-type Markov chains: Recent advances and applications, Commun. Statist. Stochastic Models 14 (1998) 479–496.
[22] B. Meini, An improved FFT-based version of Ramaswami's formula, Commun. Statist. Stochastic Models 13 (1997) 223–238.
[23] M.F. Neuts, Matrix-geometric Solutions in Stochastic Models, John Hopkins University Press, Baltimore, MD, 1981.
[24] M.F. Neuts, Structured Stochastic Matrices of $M/G/1$-type and Their Applications, Marcel Dekker, New York, NY, 1989.
[25] V. Ramaswami, A stable recursion for the steady state vector in Markov chains of $M/G/1$-type, Commun. Statist. Stochastic Models 4 (1988) 183–189.
[26] V. Ramaswami, P.G. Taylor, Some properties of the rate operators in level dependent quasi-birth-and-death processes with a countable number of phases, Commun. Statist. Stochastic Models 12 (1996) 143–164.
[27] A. Riska, E. Smirni, Exact aggregate solutions for $M/G/1$-type Markov processes, in: Proceedings of the ACM SIGMETRICS Conference, Marina Del Rey, CA, USA, June 2002, pp. 86–96.
[28] A. Riska, E. Smirni, $M/G/1$-type Markov processes: A tutorial, in: Performance Evaluation of Complex Computer Systems: Techniques and Tools, in: LNCS, vol. 2549, Springer Verlag, 2002, pp. 36–63.
[29] Y. Sasaki, H. Imai, M. Tsunoyama, et al., Approximation of probability distribution functions by Coxian distribution to evaluate multimedia systems, Systems and Computers in Japan 35 (2) (2004) 16–24.
[30] A. Stathopoulos, A. Riska, Z. Hua, et al., Bridging ETAQA and Ramaswami's formula for the solution of $M/G/1$-type processes, Performance Eval. 62 (1–4) (2005) 331–348.
[31] Q. Ye, On Latouche–Ramaswami's logarithmic reduction algorithm for quasi-birth-and-death processes, Commun. Statist. Stochastic Models 18 (2002) 449–467.