Preliminary Results on a Simple Approach to G/G/c-Like Queues

Alexandre Brandwajn[1] and Thomas Begin[2]

[1] Baskin School of Engineering, University of California Santa Cruz, USA
[2] Université Pierre et Marie Curie, LIP6, France
alexb@soe.ucsc.edu, thomas.begin@lip6.fr

Abstract. In this paper we consider a multi-server queue with a near general arrival process (represented as an arbitrary state-dependent Coxian distribution), a near general state-dependent Coxian service time distribution and a possibly finite queueing room. In addition to the dependence on the current number of customers in the system, the rate of arrivals and the progress of the service may depend on each other. We propose a semi-numerical method based on the use of conditional probabilities to compute the steady-state queue length distribution in such a queueing system. Our approach is conceptually simple, easy to implement and can be applied to both infinite and finite $C_m/C_k/c$-like queues. The proposed method uses a simple fixed-point iteration. In the case of infinite queues, it avoids the need for arbitrary truncation through the use of asymptotic conditional probabilities.

This preliminary study examines the computational behavior of the proposed method with a Cox-2 service distribution. Our results indicate that it is robust and performs well even when the number of servers and the coefficient of variation of the service times are relatively high. The number of iterations to attain convergence varies from low tens to several thousand. For example, we are able to solve queues with 1024 servers and the coefficients of variation of the service time and of the time between arrivals set to 4 within 1100 iterations.

Keywords: Multi-server queue, general arrivals, general service times, steady-state queue length distribution, simple efficient semi-numerical solution.

1 Introduction

The use of multiple processing elements or servers to provide an overall high processing capacity is a frequently applied technique in many areas including multi-core processors, distributed systems, storage processors with multiple internal "engines", virtualization in operating systems, where multiple logical CPUs are defined, as well as the Internet where most popular Web sites use multiple"mirror" sites. The numbers of servers in these applications can readily exceed 16 and appears to be growing. Due to intrinsic characteristics of the service demands or the way service is provided, it is possible for the service times and/or inter-arrival times to exhibit high variability. In particular, modern

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 159–173, 2009.

CPUs, storage processors, as well as Web sites make extensive use of internal caches to reduce the expected service time for most requests. The mixture of cache hits and much less frequent cache misses naturally leads to service time distributions characterized by high variability. The potentially high variability of service times is not limited to computer applications [32].

At a high level, the applications described can be viewed as instances of the $G/G/c$ queueing system with a possibly high coefficient of variation of the service time, as well as of the time between arrivals. In real life, the maximum queue depth or buffer capacity is finite. Additionally, in many systems, the rate of service may depend on the current number of customers in the system, e.g. if system overheads increase as the number of customers increases in computer applications. State-dependent arrival rate allows us to represent, for instance, a queue subject to requests generated by a finite set of memoryless sources. In load balancing applications, it is also possible to have arrivals of requests that depend on the progress of service.

We consider a $G/G/c$-like system in which the distribution of the times between arrivals is represented by a Coxian [10] series of memoryless stages. The parameters of this Coxian distribution may depend on the number of customers in the system. The service times are represented by a Coxian distribution generalized to include state-dependent service rates and routing probabilities. Additionally, the rate of arrivals and the progress of the service may depend on each other. A number of authors have studied algorithms for matching an arbitrary distribution by a Coxian, e.g. [7, 25, 12, 19].

We base our method on conditional probabilities, which allows us to derive a computationally efficient semi-numerical approach to the evaluation of the steady-state queue length distribution. The proposed approach, applicable to both finite and infinite $C_m/C_k/c$-like queues, does not rely explicitly on matrix-geometric techniques [20, 22]. It is conceptually simple and appears numerically stable in practice even for large numbers of servers. Unlike certain other approaches (e.g. [23]), our method requires minimal mathematical sophistication and is easy to implement in a standard programming language, which should make it of interest to every-day performance analysts. Results obtained from our method have been verified using discrete-event simulation.

As it is well known (e.g. [30]), in the case of an infinite, state-independent $G/G/c$-like queue, the form of the queue length distribution is asymptotically geometric. Our method exploits this fact to avoid arbitrary truncation present in other methods [29, 27, 22]. For the $C_m/C_k/c$ queue, the coefficient of the asymptotic geometric distribution can be independently obtained from a simple set of equations.

In this paper we present preliminary experimental results on the computational behavior of the proposed approach in the particular case when the service time distribution comprises two stages (generalized Cox-2). It is well known that a standard state-independent two-stage Coxian can be used to match the first two moments of any distribution whose coefficient of variation is greater than

$1/\sqrt{2}$, and a Coxian distribution with an unlimited number of stages (used for the inter-arrival times) can approximate arbitrarily closely any distribution [1].

There is a large body of literature devoted to queues with multiple servers. The computation of the stationary queue length distribution of the $M/M/c$ or the $M/M/c/K$ queue is easy and well known [1]. However, no simple derivation seems to exist, even for the first moment of the queue length, when the service times are not exponentially distributed. For the $M/D/c$ queue, Saaty [24] presents a method to obtain the queue length distribution in steady state, and Cosmetatos [9] proposes an approximate formula to compute the mean waiting time in such a queue. Shapiro [28] considers the $M/E_2/c$ queue, and uses an original state description that leads to a set of differential equations for which he proposes a general solution framework. Mayhugh and McCormick [18], and Heffer [13] expand Shapiro's approach to the $M/E_k/c$ queue for arbitrary values of k. As pointed out by Tijms et al. [31], the solution of the resulting set of differential equations quickly becomes intractable as the value of k or the number of servers increases. Thus, Tijms et al. [31] propose an algorithm to approximately compute the steady-state queue length distribution for the $M/G/c$ queue with variation coefficients up to 3. Hokstad [15] attempts to use the method of supplementary variables to study the $M/K_2/c$ queue, and is able to obtain partial results for up to three servers. The method of supplementary variables is also used by Hokstad [14] and Cohen [8] to derive the stationary queue length distribution for the $M/G/2$ queue. Extensions of this method to a higher number of servers do not appear practical.

Results are even more difficult to obtain when the interarrival time distribution is also general. Ishikawa [16] uses the method of supplementary variables to derive the solution for the $G/E_3/3$ queue. De Smit [11] studies the $G/H/c$ queue but is not able to prove the existence of a solution in the general case, and reports experimental results limited to the $G/H_2/c$ queue. Ramaswami and Lucantoni [21] use the embedded Markov chain approach under the assumption of a phase-type distribution of service times. Their method requires the solution of a non-linear matrix equation. The high order of the matrices involved makes the solution impractical for a higher number of servers. Bertsimas [3] considers the $C_k/C_m/c$ queue and proposes a general method to solve the resulting infinite system of partial differential equations using generating functions. Asmussen and Moller [2] propose a technique to evaluate the distribution of the waiting time in a multi-server queue with phase-type service distributions. The latter two techniques do not appear easy to implement in practice. Several authors consider purely numerical approaches. Takahashi and Takami [29] and Seelen [27] present numerical methods for the $Ph/Ph/c$ queue. Their approach involves an iterative solution of the balance equations using successive aggregation/disaggregation steps. Seelen improves on the initial method proposed by Takashi and Takami by introducing an over-relaxation parameter to speed up convergence. As is often the case, the optimal value of this parameter is not known in advance and a poor choice may interfere with the convergence of the method. Additionally, both methods [29, 27] require arbitrary truncation for a queue with unlimited queueing room, which can introduce errors. Seelen et al. [26] provide a large number of numerical studies with many different distributions for both the interarrival and service times, the number of servers not exceeding 50. Rhee and Pearce [23] cast this type of queues as a quasi birth and death process [17], and propose a solution but provide no data on its numerical behavior. Some of the approaches proposed in the past even for simpler problems turned out to exhibit computational stability issues (e.g. [35]).

In the next section we describe the queue under study, and we outline our computational approach. We consider first the general case of state-dependent arrival and service rates. We also consider the specific case of an infinite queue where the arrival and service become independent of the number of customers as the latter increases, and the asymptotic queue length distribution for such a system. Section 3 is devoted to numerical results that illustrate the behavior of our method. Although we do not have a theoretical proof of convergence or numerical stability, our preliminary results indicate that the proposed method is computationally stable even with large numbers of servers. Section 4 concludes this paper.

2 Model and Its Solution

We consider the queueing system shown in Figure 1. We denote by n the current number of customers and by c the number of servers in this system. The times between arrivals of customers are represented by a series of m memoryless stages. We use the index j $(j = 1, \ldots, m)$ to refer to the current stage of the arrival process. The c servers are assumed to be homogeneous and the service times are represented as a Coxian-like distribution with k memoryless stages. We use the index i $(i = 1, \ldots, k)$ to refer to the current stage of the service process when there are customers in the system.. We describe the state of this system in steady state by the triple $(j, \overrightarrow{l}, n)$ where j $(j = 1, \ldots, m)$ is the current stage of the arrival process, $\overrightarrow{l} = (l_1, \ldots, l_k)$ is the vector giving the numbers of customers in stages 1 through k of their service, and n is the current number of customers in the system. Note that n refers to customers having completed the arrival process but not yet departed from the system. Note also that it is sufficient to consider stages 2 through k in the vector $\overrightarrow{l}$ since we have $\sum_{i=1}^{k} l_i = \min(n, c)$.

For the service time distribution, the completion rates of the stages and the probability of exiting after each expect the last stage may depend on the current number of users in the system as well as on the current stage of the arrival process. We denote by $\mu_i(n, j)$ the service rate of stage i $(i = 1, \ldots, k)$ and by $q_i(n, j)$ the probability that the customer completes its service following stage i when there are n customers in the system and the arrival process is in stage j. We let $\hat{q}_i(n, j) = 1 - q_i(n, j)$ denote the probability that the customer proceeds to stage $i + 1$ upon completion of stage i. We assume that $\mu_i(n, j) > 0$ for $i = 1, \ldots, k$, $0 < \hat{q}_i(n, j) \leq 1$ for $i = 1, \ldots, k - 1$ and $\hat{q}_k(n, j) = 0$. For the interarrival time distribution, we denote by $\lambda_j(n, \overrightarrow{l})$ the completion rate of stage j $(j = 1, \ldots, m)$ when the current number of customers in the system is n, the state of the servers is given by $\overrightarrow{l}$, and by $p_j(n, \overrightarrow{l})$ the probability to complete the arrival process following stage j. $\hat{p}_j(n, \overrightarrow{l}) = 1 - p_j(n, \overrightarrow{l})$ denotes the probability that the customer arrival process proceeds to stage $j+1$ upon completion of the preceding stage. We have $0 < \hat{p}_j(n, \overrightarrow{l}) \leq 1$ for $j = 1, \ldots, m-1$, and $\hat{p}_m(n, \overrightarrow{l}) = 0$ for all values of n. Note that, in the case when there is no state dependency, the arrival process considered can be viewed as simply a renewal process with a Coxian interrenewal distribution. Note also that the stages described may correspond to actual stages of processing and service, or may be just a device to represent non-exponential distributions.

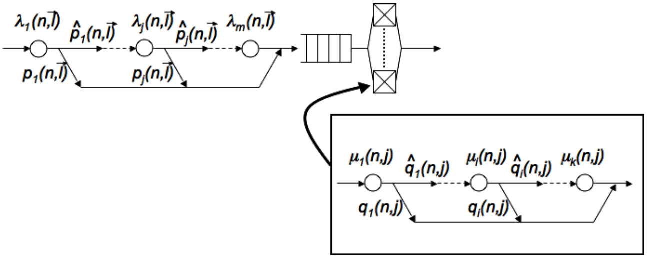


Fig. 1. $C_m/C_k/c$-like queue considered

We let $p(j, \overrightarrow{l}, n)$ be the stationary probability that the system is in the state described by $(j, \overrightarrow{l}, n)$. Denote by $p(j, \overrightarrow{l}|n)$ the corresponding conditional probability that the stage of arrival process is j and that the state of the servers is described by $\overrightarrow{l}$ given that the current number of customers in the system is n. Denote also by $p(n)$ the steady-state probability that there are n customers in the system. Clearly, assuming that $p(n) > 0$, we must have

$$p(j, \overrightarrow{l}, n) = p(j, \overrightarrow{l}|n)p(n). \tag{1}$$

For each value of n, we must also have

$$\sum_{j=1}^{m} \sum_{\overrightarrow{l}} p(j, \overrightarrow{l}|n) = 1. \tag{2}$$

It is a straightforward matter to derive the balance equations for the probabilities $p(j, \overrightarrow{l}, n)$ both in the case of a finite and infinite queueing room. It is not difficult to see that the rate of customer arrivals given n can be expressed as

$$\alpha(n) = \sum_{j=1}^{m} \sum_{\overrightarrow{l}} p(j, \overrightarrow{l}|n)\lambda_j(n, \overrightarrow{l})p_j(n, \overrightarrow{l}), \text{ for } n \geq 0. \tag{3}$$

Similarly, the rate of service completion given that there are n customers in the system can be expressed as

$$\nu(n) = \sum_{j=1}^{m} \sum_{\overrightarrow{l}} p(j, \overrightarrow{l}|n) \sum_{i=1}^{k} l_i \mu_i(n, j) q_i(n, j), \text{ for } n > 0. \tag{4}$$

Hence, $p(n)$, the steady-state probability that there are n customers in the system, is given by

$$p(n) = \frac{1}{G} \prod_{k=1}^{n} \alpha(k-1)/\nu(k), \text{ for } n \geq 0. \tag{5}$$

In formula (5), G is a normalizing constant chosen so that $\sum_{n \geq 0} p(n) = 1$. In other words, the probability $p(n)$ in our $C_m/C_k/c$-like system is the same as the steady-state probability of the number of customers in a simple birth and death process with birth (arrival) rate $\alpha(n)$ and death (service) rate $\nu(n)$. This result can be derived by summing the balance equations for the steady-state probabilities $p(j, \overrightarrow{l}, n)$ over all values of j and $\overrightarrow{l}$, and using the fact that $p(j, \overrightarrow{l}, n) = p(j, \overrightarrow{l}|n)p(n)$ [cf. [6]]. Thus we have (implicit in formula (5))

$$\begin{aligned} p(n-1)/p(n) &= \nu(n)/\alpha(n-1) \\ p(n+1)/p(n) &= \alpha(n)/\nu(n+1). \end{aligned} \tag{6}$$

To obtain the equations for these conditional probabilities $p(j, \overrightarrow{l}|n)$, it suffices to use formula (1) together with (6) in the balance equations for $p(j, \overrightarrow{l}, n)$. In the case of a finite queueing room of size N, there are several possible assumptions regarding the behavior of the arrival process at the high limit resulting in special boundary equations for $n = N$ (and possibly $n = N - 1$).

We now focus on the case of an unrestricted queueing room. One possible approach is to simply truncate the equations at some arbitrary high value for n. A more elegant approach is possible if the parameters of the arrival process $\lambda_j(n, \overrightarrow{l})$, $p_j(n, \overrightarrow{l})$, as well as those of the service process $\mu_i(n, j)$, $q_i(n, j)$ become independent of the number of users starting with some value of $n = n_0$ so that we have $\lambda_j(n, \overrightarrow{l}) = \widetilde{\lambda}_j(\overrightarrow{l})$, $p_j(n, \overrightarrow{l}) = \widetilde{p}_j(\overrightarrow{l})$, $\mu_i(n, j) = \widetilde{\mu}_i(j)$, $q_i(n, j) = \widetilde{q}_i(j)$ for $n \geq n_0$. Under these conditions, and assuming that the system under consideration is ergodic, one can expect that the conditional probabilities $p(j, \overrightarrow{l}|n)$ tend to a limit as n increases: $\lim_{n \to \infty} p(j, \overrightarrow{l}|n) = \widetilde{p}(j, \overrightarrow{l})$.

As a result, starting with a sufficiently high value of n, say $n \geq \widetilde{n}$ (clearly, $\widetilde{n} > n$), we have for $||p(j, \overrightarrow{l}|n) - \widetilde{p}(j, \overrightarrow{l})|| < \delta$ for $\delta > 0$, and the arrival and departure rates $\alpha(n)$ and $\nu(n)$ become sufficiently close to their limiting values, which we denote by $\widetilde{\alpha}$ and $\widetilde{\beta}$

$$\widetilde{\alpha} = \sum_{j=1}^{m} \sum_{\overrightarrow{l}: l_1 + \ldots + l_k = c} \widetilde{p}(j, \overrightarrow{l}) \widetilde{\lambda}_j(\overrightarrow{l}) \widetilde{p}_j(\overrightarrow{l}) \tag{7}$$

$$\widetilde{\nu} = \sum_{j=1}^{m} \sum_{\overrightarrow{l}: l_1 + \ldots + l_k = c} \widetilde{p}(j, \overrightarrow{l}) \sum_{i=1}^{k} l_i \widetilde{\mu}_i(j) \widetilde{q}_i(j). \tag{8}$$

Thus, we can express the steady-state distribution $p(n)$ as

$$p(n) \approx \frac{1}{G} \begin{cases} \prod_{k=1}^{n} \alpha(k-1)/\nu(k), & n \le \widetilde{n} \\ \prod_{k=1}^{\widetilde{n}} \alpha(k-1)/\nu(k)(\widetilde{\alpha}/\widetilde{\nu})^{n-\widetilde{n}}, & n > \widetilde{n} \end{cases} \tag{9}$$

Following a common convention, empty products are set to one. The normalizing constant G can be written as

$$G \approx 1 + \sum_{n=1}^{\widetilde{n}-1} \prod_{k=1}^{n} \alpha(k-1)/\nu(k) + \left[\prod_{k=1}^{\widetilde{n}} \alpha(k-1)/\nu(k)\right] \frac{1}{1-(\widetilde{\alpha}/\widetilde{\nu})}, \tag{10}$$

and the expected number of customers in the system can be expressed as

$$\overline{n} \approx \frac{1}{G} \left\{ \sum_{n=1}^{\widetilde{n}} np(n) + \left[\frac{\widetilde{n}}{1-(\widetilde{\alpha}/\widetilde{\nu})} + \frac{(\widetilde{\alpha}/\widetilde{\nu})}{[1-(\widetilde{\alpha}/\widetilde{\nu})]^2}\right] \prod_{k=1}^{\widetilde{n}} \alpha(k-1)/\nu(k) \right\}. \tag{11}$$

We note that the form of the solution for $p(n)$ given in formula (9) clearly shows that the steady-state distribution is asymptotically geometric with "traffic intensity" $\widetilde{\alpha}/\widetilde{\nu}$.

Thus, we solve the set of equations for the conditional probabilities $p(j, \overrightarrow{l}|n)$ for all values of n, subject to the normalizing condition given by (2). In the case of an infinite queue, the values to consider are $n = 0, \ldots, \widetilde{n}$, and in the case of a finite queueing room, all values of $n = 0, \ldots, N$. Because the equations for $p(j, \overrightarrow{l}|n)$ involve in general the conditionals for $n-1$ and $n+1$, it does not seem possible to solve these equations as a simple recurrence, as would be the case for an $M/G/1$-like queue (cf. [5]).

However, a simple-minded and simple to implement fixed-point iteration can be used to solve these equations as follows. We use a superscript to denote the iteration number. We start with some set of initial values $p^0(j, \overrightarrow{l}|n)$ for $n = 0, \ldots, n_{max}$ (where $n_{max} = N$ in the case of a finite queueing room, and $n_{max} = \widetilde{n}^0$, an initial estimate of $\widetilde{n}$, in the case of an infinite queue), and we consider the possible states in the order of increasing n, enumerating all server states $\overrightarrow{l}$ compatible with the value of n, and j, the latter varying the fastest. We compute new values for the conditional probabilities directly from the corresponding equations. For each value of n, we normalize the newly computed values so that $\sum_{j=1}^{m} \sum_{\overrightarrow{l}} p^i(j, \overrightarrow{l}|n) = 1$ once we have updated all the values for all $(j, \overrightarrow{l})$, but in the iteration we use the latest (not necessarily normalized) values as soon as they become available. Following the normalization, we can compute new values for the conditional rate of request arrivals $\alpha^i(n)$ and the rate of completions $\nu^i(n)$.

In the case of an infinite queue (under the assumptions discussed earlier in this section) we dynamically determine the "cutoff" value $\widetilde{n}^i$ as the value of n for which $|1 - \alpha^i(n-1)/\alpha^i(n)| < \epsilon$, as well as $|1 - \nu^i(n-1)/\nu^i(n)| < \epsilon$, and we consider that at this point the limiting values have been reached for the conditional probabilities at iteration i. Note that by selecting the value of ϵ as desired we control the accuracy with which the convergence to limiting conditional probabilities is determined. Thus our method provides an automatic limitation for the values of n based on the accuracy of convergence to limiting values, as opposed to arbitrary truncation used in several other methods. In practice, in most cases, the convergence to limiting values tends to occur quickly (i.e., for moderate values of $\widetilde{n}^i$), so that the steady-state distribution can be determined with high accuracy at a limited computational expense. For a finite queueing room, the maximum value for n is the size of the queueing room N, and there is no asymptotic convergence involved. The fixed-point iteration itself stops when the values of the conditional probabilities at consecutive iterations differ less than a specified convergence tolerance, e.g.$||1-p^{i-1}(j, \overrightarrow{l}|n)/p^i(j, \overrightarrow{l}|n)|| < \delta$.

The fact that we use newly computed values for $p^i(j, \overrightarrow{l}|n)$ as soon as they become available not only reduces the space requirements of our method to a single set of arrays to hold the values of $p^i(j, \overrightarrow{l}|n)$, $\alpha^i(n)$ and $\nu^i(n)$, but also appears to speed up the convergence. We have not been successful in developing a theoretical proof of convergence for the proposed approach.

In our initial study of the properties of this approach, we performed a large number of test runs concentrated on the particular case of a Cox-2 service distribution, for which $p(j, \overrightarrow{l}|n)$ can be replaced by $p(j, l_2|n)$. In our test runs, the proposed approach has always converged, typically within a relatively small number of iterations although the number of iterations tends to increase as the number of servers and the service time variability increase. The choice of the initial distribution $p^0(j, l_2|n)$ seems to have a limited effect. For each value of n, the computational complexity of every iteration scales linearly with the number of servers since the latter determines the number of values to consider for the current number of customers in their second stage of service, l_2. As discussed in the next section, for the unrestricted queue, the value of $\widetilde{n}$, and hence the number of values of n to consider, appears to increase less than linearly as the number of servers increases. Obviously, in the case of a finite queueing room, we have $n = 0, \ldots, N$ at each iteration.

In the case of an infinite queueing room, it is possible to know independently, from the solution of the corresponding equations, the limiting distribution $\widetilde{p}(j, l_2)$. We find that using this limiting distribution for $p^0(j, l_2|n)$ (truncated and normalized for $n < c$), tends to speed up the iteration. It can be readily computed from the equations for $\widetilde{p}(j, l_2)$ using a simple fixed-point iteration.

In the next section, we present numerical results to illustrate the behavior of our method for a number of values of queue parameters, including service times with high variability (coefficient of variation of over 10) and number of servers c ranging from 4 to 256.

3 Numerical Results

In this section we present numerical results to illustrate the good convergence properties of our method, as well as its ability to solve systems both with high number of servers and high service time variability. In most examples we consider

Table 1. Parameters of selected service time distributions

Dist.	Mean	Coeff.Var.	Skewness	Kurtosis	μ_1	μ_2	$\hat{q}_1$
I	1	0.8	1.80	5.05	4.25	1.308	1.000
II	1	2.0	3.06	12.77	1000.0	0.400	0.399
III	1	4.0	6.01	48.28	1000.0	0.118	0.117
IV	1	8.0	12.01	192.43	1000.0	0.031	0.031
V	1	16.0	24.02	769.55	1000.0	0.008	0.008

three levels of server utilization: 0.25, 0.5, 0.99, which correspond to 25%, 50% and 99% of the c servers busy, respectively. The Cox-2 distributions used to represent the service times in our examples are given in Table 1. Note that skewness and kurtosis relate to moments of order 3 and 4 of a probability distribution. Results in the following figures are then labeled by the corresponding coefficient of variation of the service time distribution. The mean service time is kept at one in all cases. We used discrete-event simulation to confirm the accuracy of our results for a selected set of cases.

With infinite queueing room, the "cutoff" point for the determination of $\widetilde{n}^i$ was obtained using $\epsilon = 10^{-11}$. The overall iteration convergence criterion used was $||1-\nu^{i-1}(n)/\nu^i(n)|| < \delta$ and $||1-\alpha^{i-1}(n)/\alpha^i(n)|| < \delta$ with $\delta = 10^{-5}$. These values were used for all examples presented in this paper.

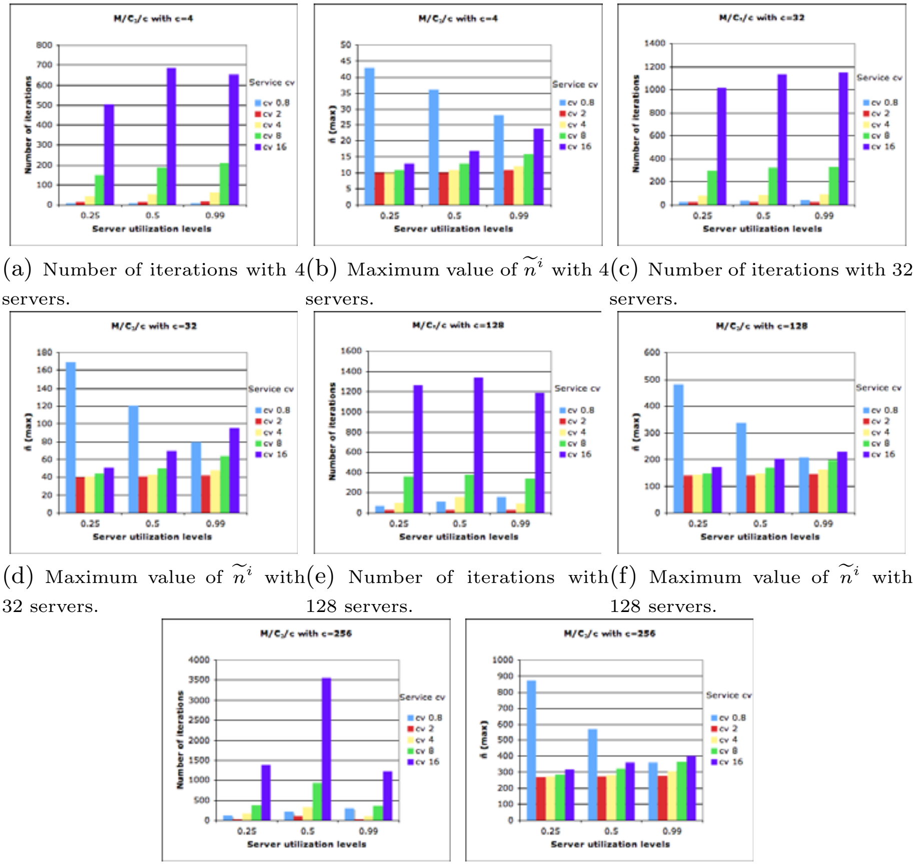
3.1 The $M/G/c$-Like Queue

In our first set of results we consider an infinite state-independent queue with Poisson arrivals, i.e., an $M/C_2/c$ queue. Figures 2a through 2h show the number of iterations needed to achieve convergence as well as the largest values of $\widetilde{n}^i$ observed during the iteration process (thus indicating the number of equations solved and storage requirements.)

We observe that the number of iterations is generally tame, ranging from no more than around 200 for coefficients of variation up to 8 and 4 servers, to below 1000 with 256 servers. In our examples, the number of iterations tends to increase as the coefficient of variation of the service time increases, although, as we discuss later in this section, the results can be quite sensitive to higher order parameters of the service time distribution. When the coefficient of variation of the service time is equal to 16, the number of iterations ranges from around 700 to below 4000. The convergence of $p(j, l_2|n)$ to the limiting distribution $\widetilde{p}(j, l_2)$ as n increases tends to occur relatively quickly. The maximum values of $\widetilde{n}^i$ attained during the iteration range from low tens to below 1000 in the "worst" case for the queue considered, viz. for 256 servers and coefficient of variation of the service set to 16.

3.2 The $M/G/c/N/N$-Like Queue

In our second set of results we consider a similar queue subject to state dependent memoryless arrivals, i.e., the rate of arrivals when there are n requests in the queue (including the ones in service) is given by $\lambda(n) = (N-n)\gamma$. Such a model corresponds to a set of N sources of requests as shown in Figure 3. Each source generates a new request after an exponentially distributed time $1/\gamma$ following the completion of its previous service period.



(a) Number of iterations with 4 servers.

(b) Maximum value of $\widetilde{n}^i$ with 4 servers.

(c) Number of iterations with 32 servers.

(d) Maximum value of $\widetilde{n}^i$ with 32 servers.

(e) Number of iterations with 128 servers.

(f) Maximum value of $\widetilde{n}^i$ with 128 servers.

(g) Number of iterations with 256 servers.

(h) Maximum value of $\widetilde{n}^i$ with 256 servers.

Fig. 2. Behavior of the method for a multi-server queue for service time distributions from Table 1 as a function of the number of servers c and the server utilization level

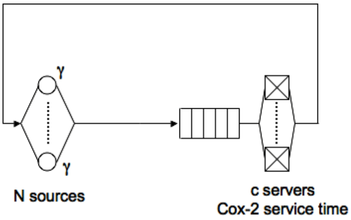
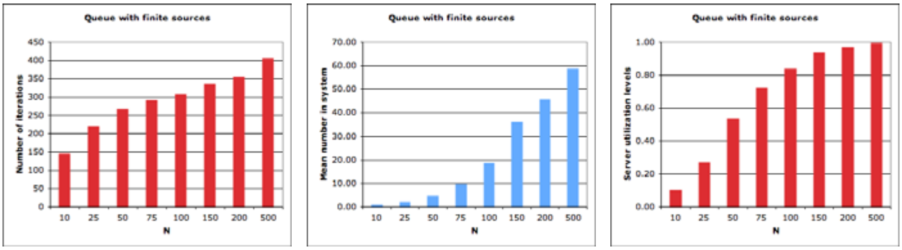


Fig. 3. Multi-server queue with N sources

The results shown in Figure 4 pertain to a queue with 8 servers and a coefficient of variation of the service time of 8. Figures 4a, 4b and 4c show the number of iterations, the expected number of customers in the system (queued and in service) $\bar{n}$, as well as the utilization level (fraction of servers busy), respectively,



(a) Number of iterations (b) Mean number of customers (c) Server utilization levels

Fig. 4. Behavior of the method for a multi-server queue with $c = 8$ servers and service time distribution Dist. IV ($cv = 8$, cf. Table 1) as a function of the number of sources N.

for numbers of sources ranging from 10 to 500. The value of γ is kept at 0.1. We observe that the number of iterations to achieve convergence tends to increase with the number of sources, but remains, in the example considered, below 500 in all cases.

3.3 The *G*/*G*/*c*-Like Queue

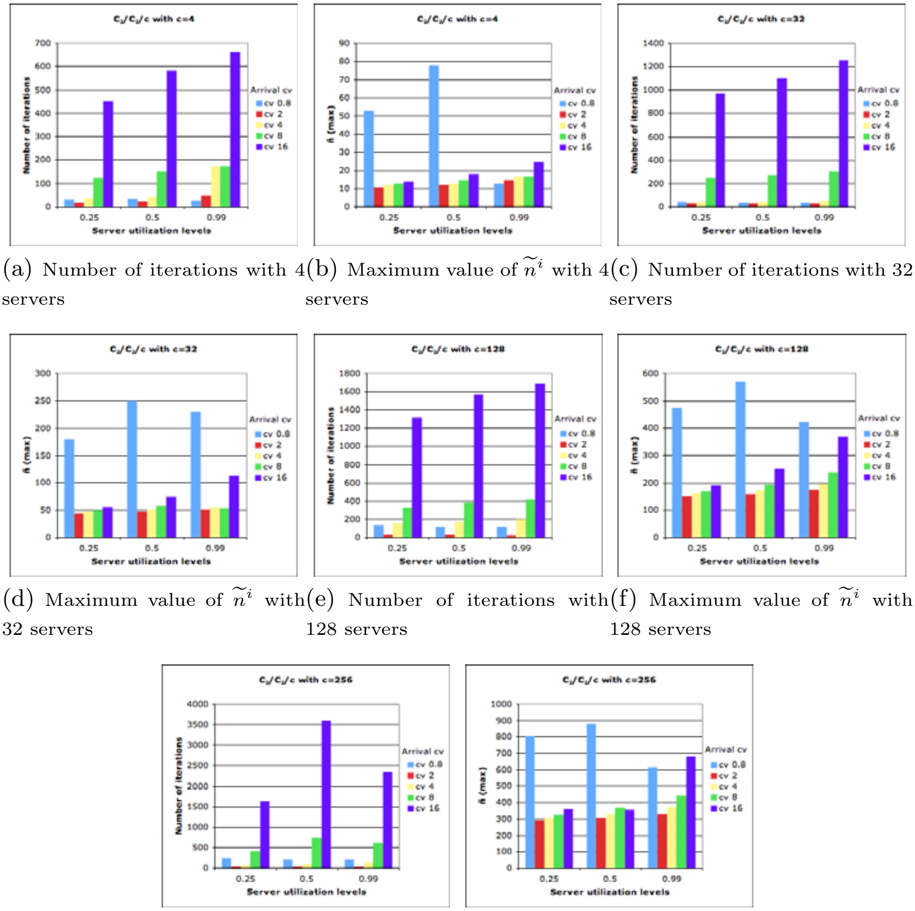
In Figure 5 we have represented results for a $C_2/C_2/c$ queue with infinite queueing room in which the time between consecutive arrivals is a Cox-2 distribution with a coefficient of variation of 4. The parameters of the service time distribution are given in Table 1. The generic parameters of the distributions of the interarrival times used in our examples are given in Table 2. The values given in this table correspond to a mean time between arrivals of one. For other values of the mean interarrival time used in our examples, the rates of the stages of the arrival process change in proportion to the inverse of that mean, while the stage transition probabilities remain constant.

Table 2. Generic parameters of selected distributions of time between arrivals

Dist.	Mean	Coeff.Var.	Skewness	Kurtosis	λ_1	λ_2	$\hat{p}_1$
I	1	0.8	1.80	5.05	4.248	1.308	1.000
II	1	2.0	3.36	15.31	10.00	0.375	0.338
III	1	4.0	6.66	59.30	10.00	0.107	0.096
IV	1	8.0	13.33	236.96	10.00	0.028	0.025
V	1	16.0	26.66	948.05	10.00	0.007	0.006

We give in Tables 1 and 2 the precise parameters of the distributions used because the results can be sensitive to higher order parameters of both the interarrival time distribution and the service time distribution [4, 33, 34]. This sensitivity extends to the performance of our method, as well as the steady-state probability distribution for the $G/G/c$ queue itself.

The number of iterations ranges typically from 200 with 4 servers to 500 with 256 servers when the coefficient of variation of the service time does not exceed 8. With the coefficient on variation of the service time set to 16, the number of iterations ranges from about 700 with 4 servers to 3500 with 256 servers. The



(a) Number of iterations with 4 servers

(b) Maximum value of $\widetilde{n}^i$ with 4 servers

(c) Number of iterations with 32 servers

(d) Maximum value of $\widetilde{n}^i$ with 32 servers

(e) Number of iterations with 128 servers

(f) Maximum value of $\widetilde{n}^i$ with 128 servers

(g) Number of iterations with 256 servers

(h) Maximum value of $\widetilde{n}^i$ with 256 servers

Fig. 5. Behavior of the method for a multi-server queue for inter-arrivals time distributions from Table 2 as a function of the number of servers c and the server utilization level

maximum values of $\widetilde{n}^i$ attained during the iteration range from low tens to below 1000 in the "worst" case, which happens to be in this case for 256 servers and coefficient of variation of the service set to 0.8.

Using a "proof-of-concept" implementation in C running on a 2.99 GHz Intel processor, for the case of 128 servers at 0.99 server utilization level, we measured execution times ranging from 3.97 s with $cv = 8$ to around 0.31 s with a lower coefficient of variation of 2 ($cv = 2$). As mentioned before, lower numbers of servers tend to result in faster execution, so that with 32 servers the corresponding results range from 0.72 s to 0.16 s. With 4 servers the execution times of our simple implementation range from about 0.14 s for $cv = 8$ to 0.03 s for $cv = 2$. Note that the vast majority of execution times are below one second.

The convergence stringency used throughout this paper, viz. $\epsilon = 10^{-11}$ and $\delta = 10^{-5}$ appears generally sufficient. When focusing on individual state probabilities in our trials, we used more stringent values: $\epsilon = 10^{-15}$ and $\delta = 10^{-8}$. There seems to be limited difference in the results obtained.

Overall, our method appears to be computationally robust, reasonably fast and quite scalable as the number of servers and the variability of service and interarrival times increase. The next section is devoted to the conclusions of this paper.

4 Conclusion

We consider a semi-numerical method to compute the steady-state distribution of the number of users in a $C_m/C_k/c$-like system where the distributions of the times between arrivals and the service times are represented by Coxian series of memoryless stages. The parameters of both Coxian distributions may depend on the current number of customers in the system. Additionally, arrivals and the progress of the service may depend on each other. We base our approach explicitly on conditional probabilities. This allows us to derive a conceptually simple and computationally efficient semi-numerical approach to the evaluation of the steady-state queue length distribution.

The proposed method can be used to solve both infinite and finite $G/G/c$-like queues of the type considered. In the case of an infinite $C_m/C_k/c$ queue whose parameters don't depend on the current number of customers, the form of the queue length distribution is asymptotically geometric. Our method exploits this fact to avoid arbitrary truncation of the balance equations. Instead, we dynamically determine, with as much stringency as desired, the convergence to asymptotic values, and use the latter in our solution. The coefficient of the geometric distribution is a by-product of our iterative solution. It can also be obtained independently, without solving the whole queue, using a simple set of equations, easily solved via fixed-point iteration.

In this preliminary study, we examined empirically the computational properties of this method in the case of a Cox-2 service distribution. Our experimental evidence indicates that the proposed method is numerically stable in practice. In our numerical examples we have explored the behavior of our approach for a range of values of the number of servers in the queue (4 to 256), as well as for several coefficients of variation of the time between arrivals and of the service times. Our results indicate that the proposed method performs well even when the number of servers is relatively high (256 in our examples) and so is the coefficient of variation (up to 16 in our examples). In the many cases we considered, the method has never failed to converge within a reasonable number of iterations. The number of iterations to attain convergence depends on the parameters of the $C_m/C_2/c$ queue considered, and varies, in our examples, from low tens to several thousand. It tends to increase for queues with high coefficients of variation of the service time and high number of servers. In additional tests, not reported in Section 3, we were able to solve queues with 1024 servers, the number of iterations not exceeding 1100 for the coefficients of variation of the service time and of the time between arrivals set to 4.

Our results underscore the potential importance of higher order moments of the interarrival time and service time distributions in the steady-state probability distribution for the number of customers in the $G/G/c$ queue. This topic is discussed in more detail in another paper.

Overall, the proposed method is conceptually simple, easy to implement, and readily applicable to both finite and infinite systems. It requires minimal mathematical sophistication. Our preliminary results indicate that it robust, fast, and scales reasonably well with the number of servers. These qualities should make the method attractive to performance analysts "in the trenches" when dealing with systems that can be modeled as mutliserver queues.

References

1. Allen, A.O.: Probability, Statistics, and Queueing Theory with Computer Science Applications, 2nd edn. Academic Press, London (1990)
2. Asmussen, S., Moller, J.R.: Calculation of the Steady State Waiting Time Distribution in $GI/PH/c$ and $MAP/PH/c$ Queues. Queueing Systems 37, 9–29 (2001)
3. Bertsimas, D.: An Analytic Approach to a General Class of $G/G/s$ Queueing Systems. Operations Research 38(1), 139–155 (1990)
4. Bondi, A.B., Whitt, W.: The influence of service-time variability in a closed network of queues. Performance Evaluation 6(3), 219–234 (1986)
5. Brandwajn, A., Wang, H.: A Conditional Probability Approach to $M/G/1$-like Queues. Performance Evaluation 65(5), 366–381 (2008)
6. Brandwajn, A.: Equivalence and Decomposition in Queueing Systems - A Unified Approach. Performance Evaluation 5, 175–185 (1985)
7. Bux, W., Herzog, U.: The Phase Concept: Approximation of Measured Data and Performance Analysis. In: Proceedings of the International Symposium on Computer Performance Modeling, Measurement and Evaluation, Yorktown Heights, NY, pp. 23–38. North-Holland, Amsterdam (1977)
8. Cohen, J.W.: On the the $M/G/2$ Queueing Model. Stochastic Processes and Their Applications 12, 231–248 (1982)
9. Cosmetatos, G.P.: Approximate Explicit Formulae for the Average Queueing Time in the Process $(M/D/r)$ and $(D/M/r)$. INFOR 13, 328–331 (1975)
10. Cox, D.R., Smith, W.L.: Queues. John Wiley, New York (1961)
11. De Smit, J.H.A.: The Queue $GI/M/s$ with Customers of Different Types or the Queue $GI/Hm/s$. Advances in Applied Probability 15(2), 392–419 (1983)
12. Faddy, M.: 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, pp. 107–114 (2002)
13. Heffer, J.C.: Steady State Solution of the $M/Ek/c$ (infinty, FIFO) queueing system. INFOR. 7, 16–30 (1969)
14. Hokstad, P.: On the Steady-State Solution of the $M/G/2$ Queue. Advances in Applied Probability 11(1), 240–255 (1979)
15. Hokstad, P.: The steady-state solution ok the $M/K2/m$ Queue. Advances in Applied Probability 12(3), 799–823 (1980)
16. Ishikawa, A.: On the equilibrium solution for the Queueing System $GI/Ek/m$. TRU Mathematics 15, 47–66 (1979)

17. Latouche, G., Ramaswami, V.: Introduction to Matrix Analytic Methods in Stochastic Modeling, ASA (1999)
18. Mayhugh, J.O., McCormick, R.E.: Steady State Solution of the Queue $M/Ek/r$. Management Science, Theory Series 14(11), 692–712 (1968)
19. McLean, S., Faddy, M., Millard, P.: 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, pp. 777–782 (2006)
20. Neuts, M.F.: Matrix-geometric solutions in stochastic models. An algorithmic approach. Courier Dover Publications (1994)
21. Ramaswami, V., Lucantoni, D.M.: Stationary waiting time distribution in queues with phase type service and in quasi-birth-and-death-processes. Stochastic Models 1, 125–136 (1985)
22. Ramaswami, V., Lucantoni, D.M.: Algorithms for the multi-server queue with phase type service. Stochastic Models 1, 393–417 (1985)
23. Rhee, K.H., Pearce, C.E.M.: On Some Basic Properties of the Inhomogeneous Quasi-Birth-And-Death Process. Comm. Korean Math. Soc. 12(1), 177–191 (1997)
24. Saaty, T.L.: Elements of Queueing Theory, with Application. The Annals of Mathematical Statistics 34(4), 1610–1612 (1963)
25. Sasaki, Y., Imai, H., Tsunoyama, M., et al.: Approximation of probability distribution functions by Coxian distribution to evaluate multimedia systems. Systems and Computers in Japan 35(2), 16–24 (2004)
26. Seelen, L.P., Tijms, H.C., Van Hoorn, M.H.: Tables for Multi-Server Queues. North-Holland, Amsterdam (1984)
27. Seelen, L.P.: An Algorithm for $Ph/Ph/c$ Queues. European Journal of the Operations Research Society 23, 118–127 (1986)
28. Shapiro, S.: The M-Server Queue with Poisson Input and Gamma-Distributed Service of Order Two. Operations Research 14(4), 685–694 (1966)
29. Takahashi, Y., Takami, Y.: A Numerical Method for the Steady-State Probabilities of a $GI/G/s$ Queueing system in a General Class. Journal of the Operations Research Society of Japan 19, 147–157 (1976)
30. Takahashi, Y.: Asymptotic Exponentiality of the Tail of the Waiting-Time Distribution in a $PH/PH/c$ Queue. Advances in Applied Probability 13(3), 619–630 (1981)
31. Tijms, H.C., Van Hoorn, M.H., Federgruen, A.: Approximations for the Steady-State Probabilities in the $M/G/c$ Queue. Advances in Applied Probability 13(1), 186–206 (1981)
32. van Dijk, N.M.: Why queuing never vanishes. European Journal of Operational Research 99(2), 463–476 (1997)
33. Whitt, W.: The Effect of Variability in the $GI/G/s$ Queue. Journal of Applied Probability 17(4), 1062–1071 (1980)
34. Wolff, R.W.: The Effect of Service Time Regularity On System Performance. In: Computer Performance, pp. 297–304. North Holland, Amsterdam (1977)
35. Ye, Q.: On Latouche-Ramaswami's logarithmic reduction algorithm for quasi-birth-and-death processes. Comm. Stat. & Stochastic Models 18, 449–467 (2002)