Breaking the dimensionality curse in multi-server queues

by

Alexandre Brandwajn [1]

Thomas Begin [2]

[1] PALLAS International Corporation, San Jose, CA, USA

[2] LIP UMR CNRS - ENS Lyon - UCB Lyon 1 - Inria, Lyon, France

Abstract

Ph/Ph/c and and *Ph/Ph/c/N* queues can be viewed as a common model of multi-server facilities. We propose a simple approximate solution for the equilibrium probabilities in such queues based on a reduced state description in order to circumvent the well-known and dreaded combinatorial growth of the number of states inherent in the classical state description. The number of equations to solve in our approach increases linearly with the number of servers and phases in the service time distribution. A simple fixed-point iteration is used to solve these equations. Our approach applies both to open models with unrestricted buffer size and to queues with finite-size buffers.

The results of a large number of empirical studies indicate that the overall accuracy of the proposed approximation appears very good. For instance, the median relative error for the mean number in the queue over thousands of examples is below 0.1% and the relative error exceeds 5% in less than 1.5% of cases explored. The accuracy of the proposed approximation becomes particularly good for systems with more than 8 servers, and tends to become excellent as the number of servers increases.

Keywords: multi-server systems; *G/G/c* queue; state-dependent *Ph/Ph/c*-like queue; dimensionality curse; reduced state description; approximate solution.

1. Introduction

A number of areas of computer applications and systems offer examples of multi-servers facilities. For instance, many-core CPUs with 32 or more cores are around the corner [BOR11]. Parallel Access Volumes in mainframe storage [MER03] provide a potentially large number of "exposures" for simultaneous access to information. Call centers with hundreds of agents [GAN03] are an element of everyday life. In the area of fiber optical cables, WDM multiplexing allows over a hundred simultaneous signals on a single fiber.

Such systems can be naturally represented as multi-server queues in which requests arrive, queue for service if all servers are found busy, and eventually leave the system after receiving service from one of the servers. Unfortunately, if one realistically assumes general distributions of times between request arrivals and general service times, the resulting *G/G/c* queueing model does not possess a known analytical solution except in some special cases [ISH79, SMI83, BER90, ASM01]. Additionally, under higher loads, realistic models must account for finite buffer space (queueing room) which may prevent requests from joining the queue when the buffer is full.

A common approach is then to replace the "general" distributions by so-called "phase-type" distributions [JOH88, BOB05, OSO06] as any distribution can be approximated arbitrarily closely by a phase-type distribution (e.g., [JOH88]). This has the distinct advantage of leading to a system of linear equations if one is interested in the steady-state probability distributions in such a system. In queueing terms, the *G/G/c* queue is replaced by the *Ph/Ph/c* queue. The latter can be solved numerically using matrix geometric methods [RAM85, LAT93, BIN05]. This approach works great as long as the number of servers and/or phases in the arrival and especially service process is not too large. However, as mentioned above, the number of servers in many realistic examples varies from several tens to many hundreds, and the traditional phase-type approach is known to suffer from the "dimensionality curse" in that the number of states (and, hence, equations to solve in the linear system) grows combinatorially as the number of servers and /or phases increases. This precludes the direct use of this approach in many interesting and important areas.

In the area of approximate solutions to such systems, several authors attempt to summarize the properties of general distributions in *G/G/c* queues by their first 2 (rarely, 3) moments [HAR13]. Although the resulting approximations are usually simple to implement and their execution is fast, unfortunately, they fail to account for the intrinsic dependence of the *G/G/c* queue on higher-order properties of the distributions involved [GUP10, BEG13] (see also Appendix). Fluid queues represent another avenue for approximation based on the fact that, as the number of requests in a queueing system tends to infinity, one can consider the flow of discrete requests as a continuous flow and hence apply fluid mechanics equations to describe the system. These methods have been applied, for example, to represent call centers [KOO02, GAN03] and the *G/G/cN* queue [WHI04]. By their principle, these approximations appear best suited for the study of limiting processing capacities of such queues.

In the particular case when the arrivals can be treated as generated by a Poisson or quasi-Poisson source, there has been recent progress in obtaining computationally manageable approximations applicable to systems with hundreds of servers. Khazaei et al [KHA12] propose to use an adaptation of the embedded Markov Chain method in the case of a pure Poisson arrival process. They show the good accuracy of their numerical results for service time coefficients of variation not exceeding 1.4. The finite buffer size in their numerical results is relatively small and kept at less than half the number of servers. Their approach does not seem easy to apply to systems with state dependencies or more general arrival processes. Brandwajn and Begin [BRA14] introduce an approximation based on a reduced state description and demonstrate the accuracy of their approach for much larger range of coefficients of variation of the service time distribution (up to 7) and buffer sizes.

Clearly, in many situations the arrival process cannot be viewed as Poisson or quasi-Poisson. Therefore, in this paper we present an extension of the reduced state approximation to systems with phase-type distributions of the time between arrivals. We also extend the approximation to open queues i.e., queues with unlimited buffer size. While no human-made system possesses a truly unrestricted buffer size, such models are of practical interest when the physical buffer size is relatively large and the server utilization is not too close to saturation. In such cases the use of an open model may result in computational saving over a finite-buffer-size model. Interestingly, in open queues, our approximation happens to tend to the correct asymptotic rates of request arrivals and service rates as the number of requests tends to infinity.

To avoid arbitrary truncation in the open model, we transform the balance equations for the reduced state into equations for the conditional probabilities of the state of the arrival process and the reduced state of the service given the current number of requests. We then exploit the convergence of such conditional probabilities to their asymptotic values so as to enumerate the states only up to the practical asymptotic convergence point.

The use of conditional probabilities partitions the state space into independently normalized subspaces, which may contribute to numerical stability, and we employ them also in the case of finite buffers. We propose a simple fixed-point iteration to solve the conditional probability equations. Although we do not have a theoretical proof of convergence to a unique solution, the proposed iteration has never failed in the large number of cases explored.

Thus, the contributions of this paper include:

- An approximate solution of the *Ph/Ph/c* queue with unrestricted buffer,
- An approximate solution of the *Ph/Ph/c/N* queue with finite buffer and possible state dependencies.

Besides the addition of the state description to account for non-Poisson (or quasi-Poisson) arrival process, this paper extends the work presented in [BRA14] by using conditional probability equations in the solution, which simplifies the treatment of the queue with unrestricted buffer.

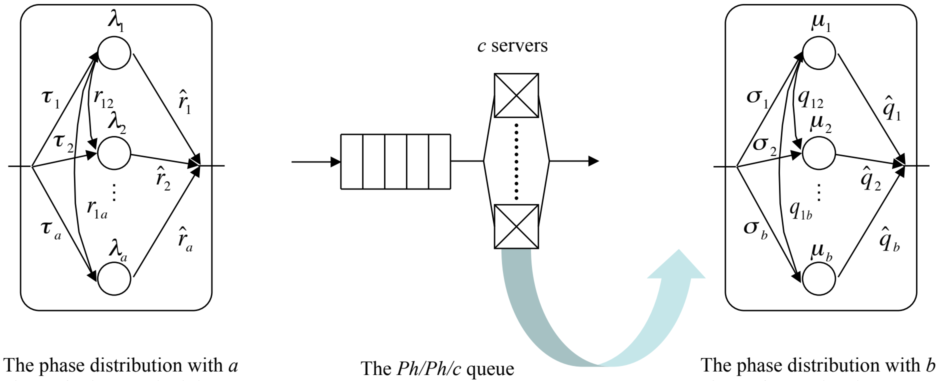
This paper is organized as follows. Section 2 is devoted to the approximate solution of the *Ph/Ph/c* queue with an infinite buffer. In Section 3 we consider a queue with a finite buffer and state-dependent distributions of interarrival times and service times. Section 4 presents numerical results to illustrate the accuracy of the proposed approximation. Finally, Section 5 concludes this paper.

2. Open model and its solution

We start by considering a classical *Ph/Ph/c* queue with an infinite buffer [HAR13]. We assume that the c servers are homogeneous, i.e., statistically identical, but not synchronized. As shown in Figure 1, the distribution of the times between arrivals comprises a exponential phases and the service time distribution has b exponential phases. We denote by σ_i the probability that service starts in phase i, by μ_i the completion rate for phase i ($i = 1,...,b$) of the service process, and by $\hat{q}_i$ the probability that the service process completes after phase i. We denote by τ_j, λ_j and $\hat{r}_j$ the corresponding quantities for phase j ($j = 1,...,a$) of the arrival process.

The classical approach to derive the steady-state probability of the number of requests (customers) in such a system is to consider a state description that includes the current number of requests in the system (n), the current phase of the arrival process (j), and the vector of the current number of requests in each phase of the service process ($\vec{m} = m_1,...,m_b$). It is clear that (for each value of n) such a full state description results in a combinatorial explosion of the number of balance equations one has to solve as the number of servers and service phases increases, compounded by the number of arrival phases.

As we are focusing on systems with large numbers of servers, to escape this issue, we use a reduced state description comprising the current number of users (n), the current phase of the arrival process (j) and the current phase of the service process for an arbitrarily selected server (i). Since for $n < c$ the selected server may be idle, we use the value $i = 0$ to denote its idle state. Let $p(n, j, i)$ be the corresponding steady-state probability where $n = 0,1...$, $j = 1,...,a$, $i = 0,...,b$. Denote by $p(n)$ the steady-state probability that there are n requests in the system, and by $p(j,i\,|\,n)$ the conditional probability of the current phase of the arrival process and the current service phase for the selected server given the current number of requests in the system. For $n = 0$, all servers are idle and only the arrival phase $j = 1,...,a$ is of significance.



The phase distribution with a phases for inter-arrival times

The *Ph/Ph/c* queue

The phase distribution with b phases for service times

Fig. 1. *Ph/Ph/c* queue with unrestricted buffer.

Using this reduced state description, it is a straightforward matter to obtain the balance equations for the probabilities $p(n,j,i)$. These equations involve the parameters of the arrival process and of the service distribution, as well as the conditional rate of completions of requests by servers **other than the chosen server** given the retained state description, which we denote by $\nu(n,j,i)$. Note that these balance equations, whose form is illustrated in the Appendix, are exact in the sense that their solution would produce the exact steady-state probabilities $p(n,j,i)$ if we knew the exact values for $\nu(n,j,i)$. This latter quantity, however, is not known explicitly and will be evaluated only approximately. Therein lies the approximation of our proposed solution.

The conditional rate of request completions by the **selected server** given the current number in the system can be expressed as

$$\varphi(n) = \sum_{j=0}^{a} \sum_{i=0}^{b} p(j,i\,|\,n)\mu_i \hat{q}_i \,. \tag{1}$$

Since the c severs are homogeneous, the overall conditional rate of request completions given n is simply

$$u(n) = c\varphi(n) \,. \tag{2}$$

Let $m = \min(n,c)$ be the number of busy servers with n customers in the system. To obtain an approximation for the unknown conditional rate of completions by other servers $\nu(n,j,i)$ we assume that the latter depends primarily on the current number of requests in the system n and not on the current phase of the arrival process or service at the selected server, when it is active, so that we have

$$\nu(n,j,i) \approx \frac{m-1}{m}u(n) = \left(c - \frac{c}{m}\right)\varphi(n) \text{ for } j = 1,\ldots,a \text{ and } i = 1,\ldots,b \,. \tag{3}$$

For $n < c$ and $i = 0$, i.e., when the selected server is not active, we use simply

$$\nu(n,j,0) \approx u(n) = c\varphi(n) \,. \tag{4}$$

This is the main approximation in our approach. In essence, we neglect the dependence on the current state of the arrival and service process in $\nu(n,j,i)$ so that the latter becomes a function only of the current number of requests in the system n. Intuitively, the neglected state information matters most when the number of servers is small. With larger numbers of servers, the knowledge of the current state of just one out of many servers or of the current phase of the arrival process conveys little information about the state of other servers. This is why we expect the approximation in formula (3) to improve as the number of servers increases.

The conditional rate of request arrivals given the current number in the system can be expressed as

$$w(n) = \sum_{j=1}^{a}\sum_{i=0}^{b} p(j,i\,|\,n)\lambda_j \hat{r}_j \,. \tag{5}$$

Hence, the steady-state probability that there are n requests in the system can be obtained in terms of $w(n)$ and $u(n)$ as

$$p(n) = \frac{1}{G}\prod_{k=1}^{n}\frac{w(k-1)}{u(k)},\;\; n = 0,1,\ldots, \tag{6}$$

where G is a normalizing constant.

Using the fact that $p(n,j,i) = p(j,i\,|\,n)p(n)$ together with formula (6) in the balance equations for $p(n,j,i)$ we can transform the latter into equations for the conditional probabilities $p(j,i\,|\,n)$. Because the quantities $w(n)$, $u(n)$ and $v(n,j,i)$ are expressed in terms of system parameters and of the probabilities $p(j,i\,|\,n)$, we obtain a self-contained system of equations for the conditional probabilities $p(j,i\,|\,n)$. These probabilities must be normalized for each value of n so that we must have

$$\sum_{j=1}^{a}\sum_{i=0}^{b} p(j,i\,|\,n) = 1,\;\; n = 0,1,\ldots \tag{7}$$

With an unrestricted buffer size, the resulting system of equations is in theory infinite since there is no upper bound on the value of the number of requests n. Assuming the system is ergodic, the equations for $p(j,i\,|\,n)$ must tend to a limit as n tends to infinity $p(j,i\,|\,n)\xrightarrow[n\to\infty]{}\tilde{p}(j,i)$ and similarly for the conditional rate of arrivals and completions $w(n)\xrightarrow[n\to\infty]{}\tilde{w}$ and $u(n)\xrightarrow[n\to\infty]{}\tilde{u}$ (as well as $v(n,j,i)\xrightarrow[n\to\infty]{}\tilde{v}(j,i)$). This is in agreement with the well-known result that the probability $p(n)$ is asymptotically geometric [TAK81]. We assume and our empirical results confirm that this is also the case when using the approximation given by (3) and (4). Denote by $\tilde{n}$ the value of n for which $p(j,i\,|\,n)$ and hence also $w(n)$ and $u(n)$ become sufficiently close to their respective asymptotic values, i.e., $|\,w(n) - \tilde{w}\,| < \delta$ and $|\,u(n) - \tilde{u}\,| < \delta$ where $\delta > 0$.

Thus, for $n = 0,\ldots,\tilde{n}-1$ we use the equations for $p(j,i\,|\,n)$ and for $n \geq \tilde{n}$ we use the corresponding equations for the asymptotic conditional probabilities $\tilde{p}(j,i)$, i.e., the limit for $n \to \infty$ of the equations for $p(j,i\,|\,n)$. Both sets of equations can be readily solved using a fixed-point iteration as outlined in the Appendix. Hence, we obtain the values of $w(n)$ and $u(n)$ for $n = 0,\ldots,\tilde{n}-1$, and the asymptotic values $\tilde{w}$ and $\tilde{u}$, which allows us to compute the steady-state probability $p(n)$ using formula (6) as

$$p(n) \approx \begin{cases} \frac{1}{G}\prod_{k=1}^{n}\frac{w(k-1)}{u(k)}, & n < \tilde{n} \\ \\ \frac{1}{G}\left(\prod_{k=1}^{\tilde{n}}\frac{w(k-1)}{u(k)}\right)\cdot\left(\frac{\tilde{w}}{\tilde{u}}\right)^{n-\tilde{n}}, & n \geq \tilde{n} \end{cases} \tag{8}$$

Clearly, for this approach to be of practical interest, such asymptotic convergence must happen for values of $\tilde{n}$ that are not excessively large. Fortunately, as shown by our numerical results, these values tend to be reasonable. Note that the value of $\tilde{n}$ is determined dynamically in our method. In a practical implementation, we dimension the data structures to hold the quantities $p(j,i\,|\,n)$, $w(n)$ and $u(n)$ according to the results presented in Section 4.4, and we monitor dynamically the iteration process for asymptotic convergence so that the enumeration of values of the number of requests in the system n can be stopped. Note also that our approximation produces the correct values for the asymptotic arrival and service rates $\tilde{w}$ and $\tilde{u}$ (see Appendix). The next section is devoted to a model with finite buffer space and state dependencies.

3. Model with finite buffer and state dependencies

In this section we consider a model with a finite buffer and possible state dependencies. We denote by N the maximum number of requests (including those in service) that can be present in the system at any time. As before, we assume that the distribution of the times between arrivals comprises a exponential phases and the service time distribution has b exponential phases. The completion rates (intensities) of these exponential phases, as well as the probabilities of initial phase selection, moving from phase to phase and of completing after a phase may depend on the current number of users in the system, $n \leq N$. Thus, we let $\mu_i(n)$ be the completion rate for phase i ($i = 1,...,b$) of the service process, and $\hat{q}_i(n)$ the probability that the service process completes after phase i. We also let $\lambda_j(n)$ and $\hat{r}_j(n)$ be the corresponding quantities for phase j ($j = 1,...,a$) of the arrival process.

As before, we use a reduced state description $p(n,j,i)$ where n is the current number of users in the system, j is the current phase of the arrival process and i is the current phase of the service process for an arbitrarily selected server. We denote by $p(n)$ the steady-state probability that there are n requests in the system, and by $p(j,i\,|\,n)$ the conditional probability for the current phase of the arrival process and of the service process at the selected server given n.

With state-dependent phase intensities and transition probabilities, the conditional rate of request completions by the selected server given the current number in the system becomes

$$\varphi(n) = \sum_{j=0}^{a}\sum_{i=0}^{b} p(j,i\,|\,n)\mu_i(n)\hat{q}_i(n)\,. \tag{9}$$

Similarly, the conditional rate of request arrivals given n becomes

$$w(n) = \sum_{j=1}^{a} \sum_{i=0}^{b} p(j,i \mid n)\lambda_j(n)\hat{r}_j(n) . \tag{10}$$

As before, the overall conditional rate of request completions given n is $u(n) = c\varphi(n)$ and we use approximation (3) and (4) for the unknown conditional rate of completions by other servers. The steady-state probability that there are n requests in the system can be computed from formula (6) once we have obtained the conditional rates of arrivals and of completions $w(n)$ and $u(n)$. To compute these two quantities, just like previously, we use the identity $p(n, j,i) = p(j,i \mid n)p(n)$ and (6) in the balance equations for $p(n, j,i)$ to transform the latter into equations for the conditional probabilities $p(j,i \mid n)$.

However, unlike in the case of an open queue (unrestricted buffer space), the number of values of n to consider is finite (and thus there is no asymptotic convergence), and we have a particular equation for $n = N$. There are several ways in which a physical system may behave when the buffer space is full, e.g. the source of arrivals may become blocked until there is space in the buffer or the source may continue to generate requests which are simply lost. We consider the latter case in this paper although our method can be readily adapted to handle the case of blocked arrivals as well. The corresponding particular equation for $n = N$ is given in the Appendix.

With this assumption on system behavior when the buffer is full, it is important in some applications to determine the probability that an arriving request will be lost. This loss probability can be expressed as

$$p_{loss} = \frac{w(N)p(N)}{\sum_{n=0}^{N} w(n)p(n)} \tag{11}$$

and thus easily calculated once we have the conditional rates of arrivals $w(n)$ and the steady-state probabilities $p(n)$.

The next section is devoted to the accuracy and other aspects of the numerical behavior of the proposed approximation.

4. Numerical results

We study the accuracy of the proposed approach for a spectrum of values of system parameters. We present results for the following numbers of servers: $c = 8,16,32,64,128$ and $c = 256$. (For numbers of servers below 8, exact numerical solutions using the full state description [RAM85, LAT93, BIN05] are manageable and there is little need for approximations.) The offered load per server, i.e., the ratio of the submitted traffic to the number of servers, varies in steps of 0.1 from 0.5 to 0.9 for queues with unrestricted buffer, and from 0.5 to 1.5 for queues with finite buffer. We consider 5 buffer sizes for the latter, expressed in relation to the number of servers as $N = 1.5c,\ 2c,\ 2.5c,\ 3c$ and $N = 4c$ (recall that N is the maximum number of requests queued and in service that can be present in the system). We use distributions of the times between arrivals and of service times with four phases and coefficients of variation ranging from 0.5 to 4 in steps of 0.5. We utilize the method proposed by Bobbio et al. [BOB05] to generate the four-phase distributions with specified coefficients of variation. The behavior of the reduced state approximation with exponential inter-arrival times (*M/Ph/c* queue) has been studied in prior work [BRA14] and the proposed approach produces exact results in the case when the service time distribution is exponential. Therefore, we skip the value 1 for the coefficient of variation of both inter-arrival and service times.

Overall, we explored 1,470 examples for the open model, and 16,170 examples with finite buffers. The performance measures considered include the mean number of requests in the system, the wait probability (probability that an arriving request finds all servers busy), as well as the loss probability in the case of a finite buffer. We use the absolute value of the relative error (expressed as a percentage) between the exact and approximate values as measure of approximation accuracy. The "exact" values come from a numerical solution of the full balance equations for $c = 8$ and $c = 16$. For larger number of servers, we use discrete-event simulation with 15 independent replications of 50,000,000 completions each. Our choice of these simulation parameters, while somewhat arbitrary, is rooted in the idea that with independent replications it may be worthwhile to have larger values for the number of completions per replication to minimize "warm-up" effects. With the values chosen, the estimated confidence intervals at 95% confidence level are generally so small that we use only the mid-point value. Given the difficulty of estimating small quantities in simulations, we only consider relative errors for wait and loss probabilities when the "exact" values exceed 0.01. To summarize the accuracy of our approach, we consider the distribution of relative errors, as well as the median error values.

We now describe the numerical results obtained for the quantities considered.

4.1. Accuracy for the mean number in the queue

We start by considering the accuracy of the method for the mean number of requests in the system. Figure 2 shows the distribution of errors and the median error as a function of the number of servers for queues with unrestricted buffer. We observe that the percentage of cases in which the error exceeds 5% falls rapidly from 46% with 8 servers to 6% with 64 servers and all the way to 0% with 256 servers. At the same time, the percentage of observed errors exceeding 15% decreases from 25% with 8 servers down to 0.4% with 64 servers and 0% for 128 or more servers. It is worthwhile noting that errors exceeding 15% observed with 8 servers tend to occur for larger values of the coefficient of variation of the service time and higher server utilizations.

Thus, as expected on intuitive grounds, the accuracy of the proposed method increases rapidly as the number of servers grows. The median error decreases from 4% with 8 servers to less than 0.4 % with 16 or more servers.

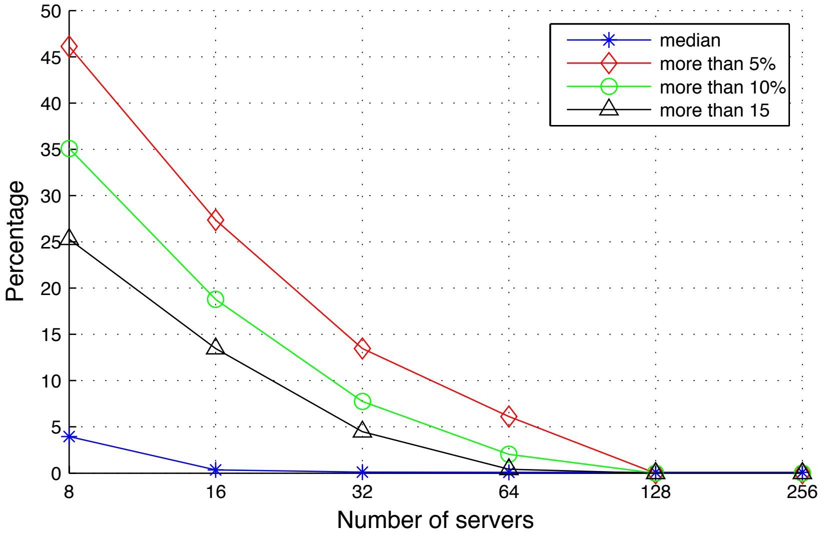


Fig. 2. Distribution of the percentage relative errors in the approximate solution for the mean number of requests in a *Ph/Ph/c* queue.

Figure 3 summarizes the results obtained for the mean number of requests in the system for queues with finite buffer. In the large set of examples explored (over 16,000) we found virtually no cases in which the relative error exceeds 10%. We observe that the infrequent larger errors tend to occur for smaller numbers of servers and when the coefficient of variation of the time between arrivals is small and the coefficient of variation of the service time is large. Overall, in some 99% of cases, the relative error remains below 5%. The median error remains below 0.1% throughout all cases explored.

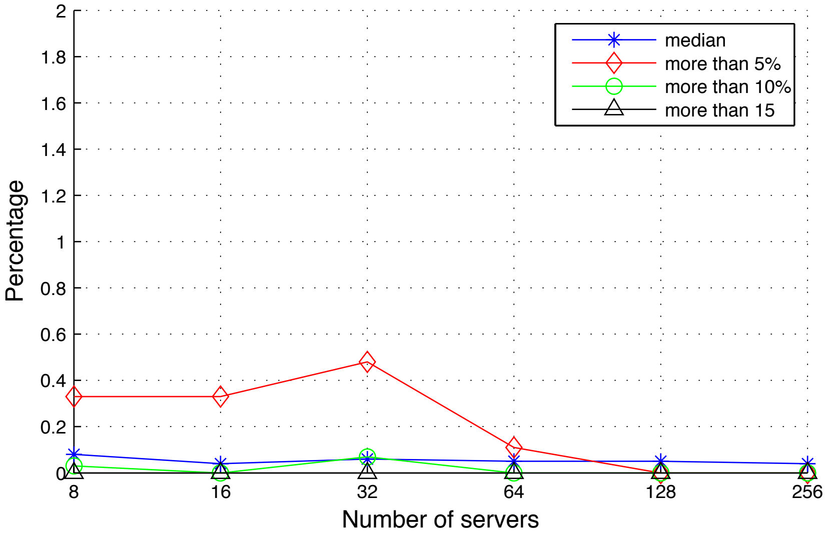


Fig. 3. Distribution of the percentage relative errors in the approximate solution for the mean number of requests in a *Ph/Ph/c/N* queue.

It is interesting to examine the accuracy of the proposed approximation as a function of the offered load per server. As an example, we show in Figure 4 the results obtained with $c = 64$ servers and other parameters spanning the spectrum of values for the coefficients of variations of the distributions of the time between arrivals and of the service time, as well as buffer sizes for finite-buffer queues, described above. We observe that the mean relative error for the mean number of requests in the system tends to peak when the offered load per server is around 0.95 (we believe that queues with unrestricted buffer approach their limit of practical validity when the load exceeds 0.9). In our results, for the open queue with 64 servers, the mean relative error peaks at some 7%, while for queues with a finite buffer the corresponding mean relative error peaks at less than 1%. The median values of the relative error peak at around 4% for the open queue and less than 0.5% for finite-buffer queues.

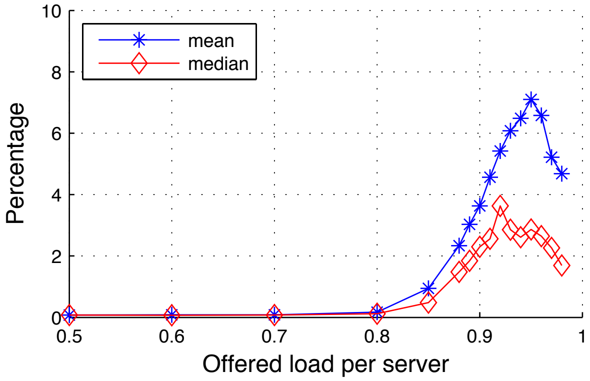


Fig. 4a. Percentage relative errors in the approximate solution for the mean number of requests in a *Ph/Ph/c* queue with *c*=64 servers as a function of the offered load per server.

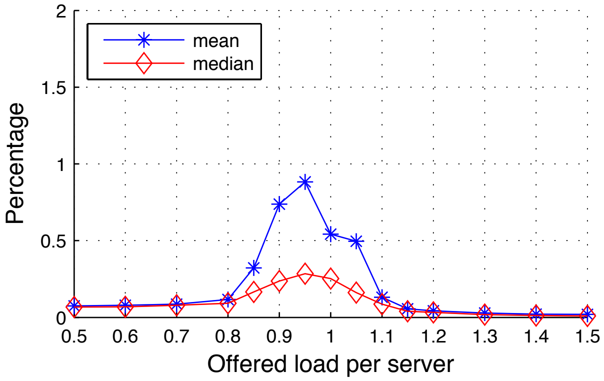


Fig. 4b. Percentage relative errors in the approximate solution for the mean number of requests in a *Ph/Ph/c/N* queue with *c*=64 servers as a function of the offered load per server.

4.2. Loss probability with finite buffers

Figure 5 illustrates the accuracy of the proposed approach for the loss probability in the case of queues with finite buffers. We observe that the percentage of cases in which the error exceeds 5% decreases from some 8% for 8 servers to 3% with 32 servers and all the way to about 1% with 256 servers. The median error remains well below 0.5% in all the cases studied.

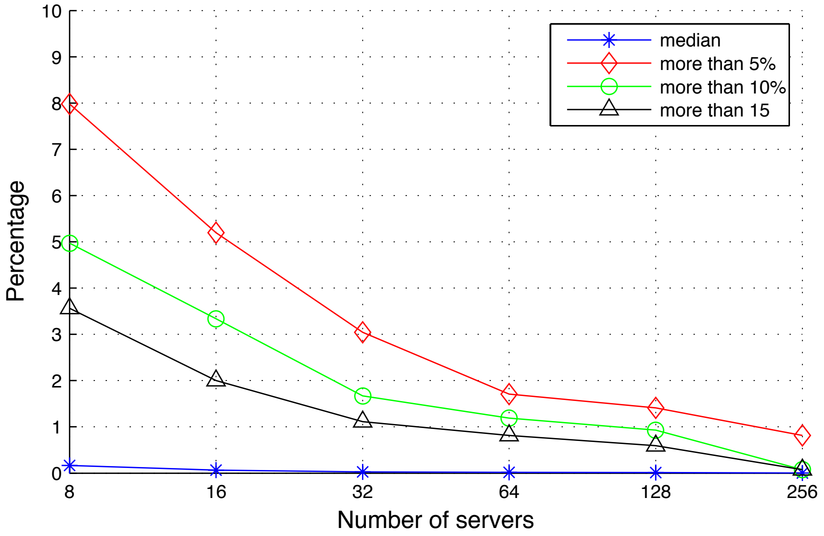


Fig. 5. Distribution of the percentage relative errors in the approximate solution for the loss probability in a *Ph/Ph/c/N* queue.

4.3. Wait probability

Table 1 summarizes the results obtained for the probability that an arriving request has to wait in the case of queues with unrestricted buffer. In the over 1,400 examples studied, the relative errors remain below 5% in over 96% of the cases explored. The mean error is around 1% and the median error is below 0.5%.

Table 1. Distribution of the percentage relative errors in the approximate solution for the wait probability in a *Ph/Ph/c* queue.

Average	Median	<5%	5-10%	10-15%	>15%
1.02%	0.22%	96.05%	2.28%	0.87%	0.80%

Table 2 displays analogous results for queues with finite buffers. Here we observe that the relative error remains below 5% in over 98% of the 16,170 cases studied. The mean error is below 1% and the median error is less than 0.1% confirming the impressive accuracy of the proposed approach in the case of finite buffer space.

Table 2. Distribution of the percentage relative errors in the approximate solution for the wait probability in a *Ph/Ph/c/N* queue.

Average	Median	<5%	5-10%	10-15%	>15%
0.60%	0.08%	98.23%	1.27%	0.25%	0.25%

4.4. Speed of asymptotic convergence for queues with unrestricted buffer

As discussed in Section 2, in the case of unrestricted buffer size our approximation relies on the convergence of the conditional probabilities $p(j,i\,|\,n)$ to their asymptotic values $\tilde{p}(j,i)$ in order to transform an infinite set of equations into a finite one without arbitrary truncation. Table 3 shows the mean and the median values of the ratio $\tilde{n}/c$ with a rather stringent $\delta = 10^{-11}$. Recall that δ corresponds to the point at which we consider that asymptotic convergence has been achieved for the conditional rates $w(n)$ and $u(n)$. We observe that the mean and median values of $\tilde{n}$ grow less than linearly as the number of servers increases. Clearly, the value of $\tilde{n}$ increases as the number of servers increases, but relative to the number of servers, the rate of growth slows down markedly as c increases.

Table 3. Relative mean and median values found for $\tilde{n}$.

	c=8	c=16	c=32	c=64	c=128	c=256
Mean	33.4	22.3	16.0	12.6	9.9	8.0
Median	35.9	22.4	15.9	12.8	9.9	8.2

4.5. Model with state dependencies

The good accuracy of the proposed approximation appears to extend to the case when the intensity of the arrivals and the service phases depend on the current number of requests in the system. As an example, Figure 6 compares the results obtained for the mean number of request in the system using our approximation and using an exact numerical solution of the full balance equations for several levels of the offered load in the queue. This example corresponds to a system with $c = 32$ servers and a finite queueing room of $N = 128$. The service process has a coefficient of variation $c_S = 3$. The completion rate of each service phase is a function of the current number of requests in the system such that the service rate decreases linearly from full speed for a single user down to 50% of its full speed as the number of users reaches full system capacity N. The arrival process is quasi-Poisson with rate $\lambda(n) = \phi \times (1 - n/(3N))$, i.e., it represents a set of $3N$ identical exponential request sources. Figure 5 illustrates the good accuracy of the approximation for the mean number of requests in the system as a function of the maximum offered load ϕ.

Overall, in the many examples explored, the accuracy of the proposed method appears excellent, especially for larger numbers of servers.

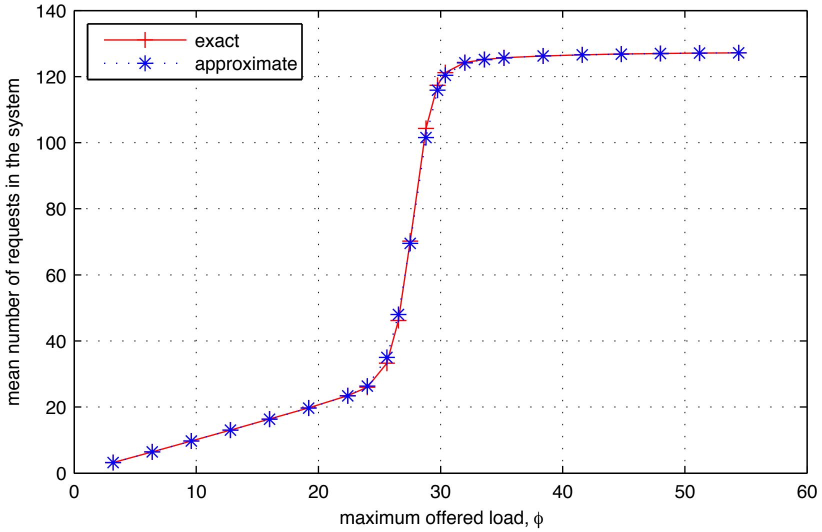


Fig. 6. Mean number of requests in *Ph/Ph/c/N* queue with $c = 32$, $N = 4c$, $c_S = 3$ and state dependencies for the service process and the arrival process for various levels of workload.

5. Conclusions

We propose a simple approximate solution for *Ph/Ph/c* queues, which uses a reduced state description in order to circumvent the combinatorial growth of the number of states inherent in the classical state description used for such queueing systems. The number of equations to solve in our approach grows only linearly with the number of servers and phases in the distributions of times between arrivals and of service times. A simple fixed-point iteration is used to solve these equations. Although we do not have a theoretical proof of convergence of the fixed-point iteration, in the very many examples explored, it never failed to converge within a reasonable number of iterations.

Like with any approximation, it is important to understand its accuracy and its domain of applicability. Unfortunately, we have not been able to obtain theoretical bounds for the errors of our approximation. This is why we undertook a systematic empirical study of its behavior over a large spectrum of examples attempting to cover cases that tend to cause problems. We considered both relatively small and much larger numbers of servers, small and large buffer sizes (in the case of finite buffers), as well as a spectrum of values for the offered load. We concentrated our examples on medium to high load where most of the errors tend to occur. We considered a relatively large set of values for the coefficients of variation of the inter-arrival and service time distributions concentrating on values well above one which, again, tend to cause problems for approximation methods.

We used common performance indices such as the mean number of requests in the system, the wait probability, as well as the loss probability in the case of queues with finite buffers to assess the accuracy of our approximation. Our results indicate that the accuracy of the approximation is generally very good for 8 or more servers and tends to improve rapidly as the number of servers grows. The median relative error for the mean number in the queue over the totality of examples considered (17,640 distinct scenarios) is below 0.1% and the relative error exceeds 5% in less than 1.5% of the scenarios considered.

For the wait probability and the loss probability, the median error is below 0.5% and the observed errors exceed 5% in less than 4% of the cases explored. Overall, the accuracy of the proposed approximation appears very good.

6. References

[ASM01] Asmussen, S., and Moller, J. R. Calculation of the Steady State Waiting Time Distribution in *GI/PH/c* and *MAP/PH/c* Queues, Queueing Systems, Vol. 37, (2001), pp. 9-29.

[BEG13] Begin, T., and Brandwajn, A. A note on the accuracy of several existing approximations for *M/Ph/m* queues, IEEE HSNCE, (2013).

[BER90] Bertsimas, D., An Analytic Approach to a General Class of *G/G/s* Queueing Systems, Operations Research, Vol. 38 (1), (1990), pp. 139-155.

[BIN05] Bini, D., Latouche, G., and Meini, B. Numerical methods for structured Markov chains, Oxford: Oxford University Press, (2005).

[BOB05] Bobbio, A., Horvath, A., and Telek, M. Matching three moments with minimal acyclic phase type distributions, Stochastic Models, Vol. 21, (2005), pp. 303-326.

[BOL05] Bolch, G., Greiner, S., Meer, H., and Trivedi, K., Queueing Networks and Markov Chains. Second Edition, Wiley-Interscience, 2005.

[BOR11] Borkar, S., and Chien, A. A. The future of microprocessors. Communications of the ACM, Vol. 54 (5), (2011), pp. 67-77.

[BRA14] Brandwajn, A., and Begin, T. Reduced complexity in *M/Ph/c/N* queues, Performance Evaluation, Vol. 78, (2014), pp. 42-54.

[GAN03] Gans, N., Koole, G., and Mandelbaum, A. Telephone call centers: Tutorial, review, and research prospects. Manufacturing & Service Operations Management, Vol. 5 (2), (2003), pp. 79-141.

[GUP10] Gupta, V., Harchol-Balter, M., Dai, J., and Zwart, B. On the inapproximability of *M/G/K*: why two moments of job size distribution are not enough, Queueing Systems, Vol. 64 (1), (2010), pp. 5-48.

[HAR13] Harchol-Balter, M. Performance Modeling and Design of Computer Systems: Queueing Theory in Action, Cambridge University Press, (2013).

[ISH79] Ishikawa, A. On the equilibrium solution for the Queueing System *GI/Ek/m*, TRU Mathematics, Vol. 15, (1979), pp. 47-66.

[JOH88] Johnson, M.A., and Taaffe, M.R. The denseness of phase distributions, School of Industrial Engineering, Purdue University, (1988).

[KHA12] Khazaei, H., Misic, J., and Misic, V. B. Performance analysis of cloud computing centers using *m/g/m/m+ r* queuing systems, IEEE Transactions on Parallel and Distributed Systems, Vol. 23(5), (2012), pp. 936-943.

[KOO02] Koole, G., and Mandelbaum, A. Queueing models of call centers: An introduction, Annals of Operations Research, Vol. 113 (1), (2002), pp. 41-59.

[LAT93] Latouche, G., and Ramaswami, V. A logarithmic reduction algorithm for quasi-birth-and-death processes, Journal of Applied Probability, Vol. 30, (1993), pp. 650-674.

[MER03] Meritt, A. S., Staubi, J. A., Trowell, K. M., Whistance, G., and Yudenfriend, H. M. z/OS support of the IBM Total Storage enterprise storage server, IBM Systems Journal, Vol. 42 (2), (2003), pp. 280-301.

[OSO06] Osogami, T., and Harchol-Balter, M. Closed form solutions for mapping general distributions to quasi-minimal PH distributions, Performance Evaluation, Vol. 63 (6), (2006), pp. 524-552.

[RAM85] Ramaswami, V., and Lucantoni, D.M. Algorithms for the multi-server queue with phase type service, Stochastic Models, Vol. 1, (1985), pp. 393-417.

[SMI83] 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, Vol. 15 (2), (1983), pp. 392-419.

[TAK81] Takahashi , Y. Asymptotic Exponentiality of the Tail of the Waiting-Time Distribution in a *PH/PH/c Queue*, Advances in Applied Probability, Vol. 13 (3), (1981), pp. 619-630.

[WHI04] Whitt, W. A diffusion approximation for the G/GI/n/m queue. Operations Research, Vol. 52 (6), (2004), pp. 922-941.

7. Appendix

7.1 Balance equations and equations for conditional probabilities

Denote by r_{jk} the probability that phase j of the arrival process is followed by phase k, and by q_{il} the probability that phase i of the service process is followed by phase l.

The balance equations for the reduced state description $p(n,j,i)$ considered in this paper have the following form

For $n = 2,...,c-1$, $j = 1,...,a$

$$\begin{aligned} &p(n,j,0)[\lambda_j+\nu(n,j,0)]=\sum_{k=1}^{a}p(n-1,k,0)\lambda_k\hat{r}_k\tau_j\left(\frac{c-n}{c-n+1}\right)+\sum_{k=1}^{a}p(n,k,0)\lambda_k r_{kj}\\ &+\sum_{l=1}^{b}p(n+1,j,l)\mu_l\hat{q}_l+p(n+1,j,0)\nu(n+1,j,0)\end{aligned}\tag{A1}$$

and for $i=1,...,b$

$$\begin{aligned} &p(n,j,i)[\lambda_j+\mu_i+\nu(n,j,i)]=\sum_{k=1}^{a}p(n-1,k,0)\lambda_k\hat{r}_k\tau_j\left(\frac{1}{c-n+1}\right)\sigma_i+\sum_{k=1}^{a}p(n-1,k,i)\lambda_k\hat{r}_k\tau_j\\ &+\sum_{l=1}^{b}p(n,j,l)\mu_l q_{li}+\sum_{k=1}^{a}p(n,k,i)\lambda_k r_{kj}+p(n+1,j,i)\nu(n+1,j,i)\end{aligned}\tag{A2}$$

As mentioned in Section 2, $\nu(n,j,i)$ denotes the conditional rate of completions of requests by servers other than the chosen server given the current system state (n,j,i).

Similarly, the balance equations for $n>c$, $j=1,...,a$, $i=1,...,b$ have the form

$$\begin{aligned} &p(n,j,i)[\lambda_j+\mu_i+\nu(n,j,i)]=\sum_{k=1}^{a}p(n-1,k,i)\lambda_k\hat{r}_k\tau_j+\sum_{l=1}^{b}p(n,j,l)\mu_l q_{li}+\sum_{k=1}^{a}p(n,k,i)\lambda_k r_{kj}\\ &+p(n+1,j,i)\nu(n+1,j,i)+\sum_{l=1}^{b}p(n+1,j,l)\mu_l\hat{q}_l\sigma_i\end{aligned}\tag{A3}$$

There are, additionally, analogous balance equations for the various boundary cases. Note that these balance equations are exact in the sense that, if we knew the exact values for the conditional completion rates $\nu(n,j,i)$, the solution of these balance equations would produce the exact steady-state probabilities $p(n,j,i)$. The approximation in our method stems from the fact that we compute an approximate value for $\nu(n,j,i)$. Also, in the case of an open model, some minimal level of approximation is introduced through the use of the asymptotic convergence to stop the enumeration of increasing values of n. As described in Section 2, the balance equations are then transformed into equations for the conditional probabilities $p(j,i\,|\,n)$.

We present below the conditional probability equations used in the solution of the open model.

For $n=0$, $j=1,...,a$

$$p(j,0\,|\,0)\lambda_j=[\sum_{i=1}^{b}p(j,i\,|\,1)\mu_i\hat{q}_i+p(j,0\,|\,1)\nu(0,j,i)]w(0)/u(1)+\sum_{k=1}^{a}p(k,0\,|\,0)\lambda_k r_{kj}\,.\tag{B1}$$

For $n=1$, $j=1,...,a$

$$\begin{aligned} &p(j,0\,|\,1)[\lambda_j+\nu(1,j,0)]=[\sum_{i=1}^{b}p(j,i\,|\,2)\mu_i\hat{q}_i+p(j,0\,|\,2)\nu(2,j,0)]w(n)/u(n+1)\\ &+\sum_{k=1}^{a}p(k,0\,|\,0)\lambda_k\hat{r}_k\tau_j(1-1/c)u(1)/w(0)+\sum_{k=1}^{a}p(k,0\,|\,1)\lambda_k r_{kj}\end{aligned}\quad ,\tag{B2}$$

and for $i=1,...,b$

$$\begin{aligned} p(j,i\,|\,1)[\lambda_j+\mu_i] &= \sum_{k=1}^{b} p(k,0\,|\,0)\lambda_k \hat{r}_k \tau_j (1/c)\sigma_i u(1)/w(0) + \sum_{l=1}^{b} p(j,l\,|\,1)\mu_l q_{li} \quad . \\ &+ p(j,i\,|\,2)\nu(2,j,i)w(1)/u(2) + \sum_{k=1}^{a} p(k,1\,|\,1)\lambda_k r_{kj} \end{aligned} \tag{B3}$$

For $n = 2,\dots,c-1$, $j = 1,\dots,a$

$$\begin{aligned} p(j,0\,|\,n)[\lambda_j+\nu(n,j,0)] &= [\sum_{l=1}^{b} p(j,l\,|\,n+1)\mu_l \hat{q}_l + p(j,0\,|\,n+1)\nu(n+1,j,0)]w(n)/u(n+1) \\ &+ \sum_{k=1}^{a} p(k,0\,|\,n-1)\lambda_k \hat{r}_k \tau_j \left(\frac{c-n}{c-n+1}\right) u(n)/w(n-1) + \sum_{k=1}^{a} p(k,0\,|\,n)\lambda_k r_{kj} \end{aligned} \tag{B4}$$

and for $i = 1,\dots,b$

$$\begin{aligned} & p(j,i\,|\,n)[\lambda_j+\mu_i+\nu(n,j,i)] = \\ & [\sum_{k=1}^{a} p(k,0\,|\,n-1)\lambda_k \hat{r}_k \tau_j \left(\frac{1}{c-n+1}\right)\sigma_i + \sum_{k=1}^{a} p(k,i\,|\,n-1)\lambda_k \hat{r}_k \tau_j]u(n)/w(n-1) \\ & + \sum_{l=1}^{b} p(j,l\,|\,n)\mu_l q_{li} + p(j,i\,|\,n+1)\nu(n+1,j,i)w(n)/u(n+1) + \sum_{k=1}^{a} p(k,i\,|\,n)\lambda_k r_{kj} \end{aligned} \tag{B5}$$

Note that we have $\nu(n,j,0) = 0$ and $p(j,0\,|\,n) = 0$ for all $n \ge c$ since no server can be idle when the number of requests is at least equal to the number of servers.

For $n = c$, $j = 1,\dots,a$, $i = 1,\dots,b$

$$\begin{aligned} p(j,i\,|\,c)[\lambda_j+\mu_i+\nu(c,j,i)] &= [\sum_{k=1}^{a} p(k,0\,|\,c-1)\lambda_k \hat{r}_k \sigma_i + \sum_{k=1}^{a} p(k,i\,|\,c-1)\lambda_k \hat{r}_k]\tau_j u(c)/w(c-1) \\ &+ [p(j,i\,|\,c+1)\nu(c+1,j,i) + \sum_{l=1}^{b} p(j,l\,|\,c+1)\mu_l \hat{q}_l \sigma_i]w(c)/u(c+1) \\ &+ \sum_{l=1}^{b} p(j,l\,|\,c)\mu_l q_{li} + \sum_{k=1}^{a} p(k,i\,|\,c)\lambda_k r_{kj} \end{aligned} \tag{B6}$$

For $n > c$, $j = 1,\dots,a$, $i = 1,\dots,b$

$$\begin{aligned} p(j,i\,|\,n)[\lambda_j+\mu_i+\nu(n,j,i)] &= \sum_{k=1}^{a} p(k,i\,|\,n-1)\lambda_k \hat{r}_k \tau_j u(n)/w(n-1) \\ &+ [p(j,i\,|\,n+1)\nu(n+1,j,i) + \sum_{l=1}^{b} p(j,l\,|\,n+1)\mu_l \hat{q}_l \sigma_i]w(n)/u(n+1) \\ &+ \sum_{l=1}^{b} p(j,l\,|\,n)\mu_l q_{li} + \sum_{k=1}^{a} p(k,i\,|\,n)\lambda_k r_{kj} \end{aligned} \tag{B7}$$

In the case of a finite buffer with lost arrivals, the equation for $n = N$, $j = 1,\dots,a$, $i = 1,\dots,b$ is of the form

$$\begin{aligned} p(j,i\,|\,N)[\lambda_j(1-\hat{r}_j\tau_j)+\mu_i+\nu(N,j,i)] &= \sum_{k=1}^{a} p(k,i\,|\,N-1)\lambda_k \hat{r}_k \tau_j u(N)/w(N-1) \\ &+ \sum_{l=1}^{b} p(j,l\,|\,N)\mu_l q_{li} + \sum_{k=1}^{a} p(k,i\,|\,N)\lambda_k r_{kj} + \sum_{k=1,k\ne j}^{a} p(k,i\,|\,N)\lambda_k \hat{r}_k \tau_j \end{aligned} \tag{B8}$$

In the case of an unrestricted queue (open system), the balance equations tend to the following asymptotic form for $j = 1,...,a$, $i = 1,...,b$

$$\begin{aligned} &\tilde{p}(j,i)[\lambda_j + \mu_i + \tilde{v}(j,i)] = \sum_{k=1}^{a} \tilde{p}(k,i)\lambda_k \hat{r}_k \tau_j \tilde{u} / \tilde{w} \\ &+[\tilde{p}(j,i)\tilde{v}(j,i) + \sum_{l=1}^{b} \tilde{p}(j,l)\mu_l \hat{q}_l \sigma_i]\tilde{w} / \tilde{u} + \sum_{l=1}^{b} \tilde{p}(j,l)\mu_l q_{li} + \sum_{k=1}^{a} \tilde{p}(k,i)\lambda_k r_{kj} \end{aligned} \tag{B9}$$

where we must have $\sum_{j=1}^{a} \sum_{i=1}^{b} \tilde{p}(j,i) = 1$.

7.2 Outline of fixed-point iterative solution

While one can easily design a simple fixed-point iteration to solve equation (B9) directly, it is worthwhile noting that is intuitively clear that the asymptotic distribution $\tilde{p}(j,i)$ must be the same as the asymptotic distribution in a *Ph/Ph/1* queue with the same service time distribution as the original *Ph/Ph/c* queue but the distribution of the time between arrivals modified so that the intensity of each phase (λ_j) is divided by the number of servers c . In other words, for the asymptotic distribution, the servers can be viewed as functioning independently and separately, each handling an equal share of the workload. Moreover, it is also intuitively clear that for $n \to \infty$, the servers become "decoupled" from the arrival process so that the distribution $\tilde{p}(j,i)$ must have a product form: $\tilde{p}(j,i) = f(j)g(i)$ where $\sum_{j=1}^{a} f(j) = 1$ and $\sum_{i=1}^{b} g(i) = 1$. These two observations lead to a significantly simplified computation of the asymptotic distribution where one can iterate between the computation of successive approximations of $f(j)$ and of $g(i)$.

A simple-minded solution of the set of equations (B1) through (B8) may proceed as follows.

1. Start with a feasible initial distribution $p^0(j,i\,|\,n)$, $n = 0,1,...$
For an open queue, it is helpful to find first the asymptotic distribution $\tilde{p}(j,i)$ and use it as the initial distribution for all $n \geq c$. For $n < c$, and in the case of a finite buffer, any reasonable initial distribution should work.

2. Compute initial values for derived quantities: $w^0(n)$, $u^0(n)$ and $v^0(n,j,i)$.

3. At iteration ℓ consider values of n in increasing order $n = 0,1,...$

a. For each value of n, compute new non-normalized values $\hat{p}^{\ell}(j,i\,|\,n)$ directly from the corresponding equation (B1) through (B8) using values available from the previous iterations ($p^{\ell-1}(j,i\,|\,n+1)$, $u^{\ell-1}(n+1)$, $w^{\ell-1}$ (n+1), $u^{\ell}(n)$, $w^{\ell}(n)$, etc), already normalized values from the current iteration ($p^{\ell}(j,i\,|\,n-1)$, $w^{\ell}(n)$), as well as any available newly obtained non-normalized values $\hat{p}(j,i\,|\,n)$.

b. Normalize the newly computed values $\hat{p}^{\ell}(j,i\,|\,n)$ to obtain $p^{\ell}(j,i\,|\,n) = \hat{p}^{\ell}(j,i\,|\,n) / \sum_{j=1}^{b} \sum_{i=0}^{a} \hat{p}^{\ell}(j,i\,|\,n)$ and compute new values for derived quantities $w^{\ell}(n)$, $u^{\ell}(n)$ and $v^{\ell}(n,j,i)$.

c. In the case of an open queue, stop the enumeration of $n > c$ when the values for $p^{\ell}(j,i\,|\,n)$ become sufficiently close to the values for $p^{\ell}(j,i\,|\,n-1)$ (asymptotic convergence).

4. If the newly computed values for $w^{\ell}(n)$ and $u^{\ell}(n)$ are sufficiently close to the values from the preceding iteration $w^{\ell-1}(n)$ and $u^{\ell-1}(n)$, i.e., $\max\{|\,w^{\ell}(n) - w^{\ell-1}(n)\,|\} < \varepsilon$ and $\max\{|\,u^{\ell}(n) - u^{\ell-1}(n)\,|) < \varepsilon$ with $\varepsilon > 0$, stop the iteration. Otherwise, go to Step 3.

5. Compute the steady-state probability distribution $p(n)$ from formula (6), as well any derived performance metrics.

7.3 Speed of convergence of the fixed-point iterative solution

The proposed solution of the equations for $p(j,i\,|\,n)$ is a rather straightforward fixed-point iteration, presented as a simple "proof of concept". Nonetheless, it may be interesting to examine the number of iterations required to attain convergence. The latter varies with specific queue parameters but, on average, seems to depend mostly on the number of servers in the system. Interestingly, the median number of iterations needed is sufficiently close for queues with unrestricted and with finite buffer that they can be displayed together. Table 4 summarizes the median number of iterations relative to (i.e., divided by) the number of servers with $\varepsilon = 10^{-8}$.

Table 4. Relative median number of iterations before convergence is found.

	c=8	c=16	c=32	c=64	c=128	c=256
Median	143.1	143.3	136.3	127.6	118.2	109.5

We observe that, while the number of iterations clearly grows with the number of servers, the rate of increase tends to be less than linear, especially for larger number of servers.

7.4 Example of comparison with simple approximations

As mentioned in the introduction, the few existing simple approximations fail to account for potentially important distributional dependencies in the *G/G/c* queue. Figure 7 illustrates the resulting relative error for the mean number of requests in the system in a queue with unrestricted buffer with $c = 16$ servers, the coefficient of variation of the time between arrivals $c_a = 3$ and the coefficient of variation of the service time $c_s = 0.5$ for two simple approximations: the Allen-Cunneen approximation and the approximation proposed by Kulbatzki [BOL05].

We observe that both simple approximations produce significant errors as the load per server approaches 0.8 while the relative error in our approximation remains small. Similar large errors occur in many other examples.

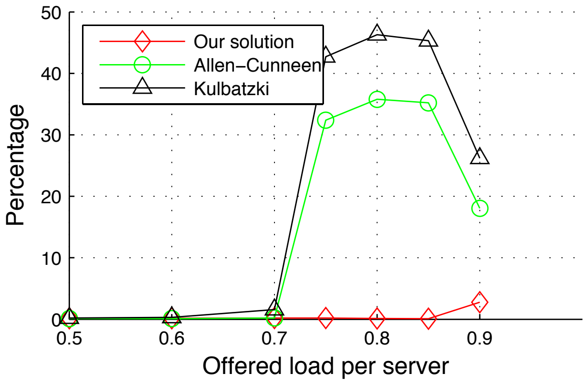


Fig. 7. Relative error in the mean number of requests in *Ph/Ph/c* queue with $c = 16$, $c_S = 0.5$, and $c_a = 3$ for various levels of workload.