Reduced complexity in *M/Ph/c/N* queues

Alexandre Brandwajn
Baskin School of Engineering
University of California Santa Cruz
USA
alexb@soe.ucsc.edu

Thomas Begin
LIP UMR CNRS - ENS Lyon -
UCB Lyon 1 - INRIA 5668
France
thomas.begin@ens-lyon.fr

ABSTRACT

Many real-life systems can be modeled using the classical *M/G/c/N* queue. A frequently-used approach is to replace the general service time distribution by a phase-type distribution since the *M/Ph/c/N* queue can be described by familiar balance equations. The downside of this approach is that the size of the resulting state space suffers from the "dimensionality curse", i.e., exhibits combinatorial growth as the number of servers and/or phases increases.

To circumvent this complexity issue, we propose to use a reduced state description in which the state of only one server is represented explicitly, while the other servers are accounted for through their rate of completions. The accuracy of the resulting approximation is generally good and, moreover, tends to improve as the number of servers in the system increases. Its computational complexity in terms of the number of states grows only linearly in the number of servers and phases.

Keywords: Multiple servers, general service, finite buffer, *M/Ph/c/N* queue, reduced-state approximation, linear complexity.

1. INTRODUCTION

A large number of real-life systems (such as call centers, multi-core processors, etc.) can be modeled as instances of the classical *M/G/c/N* queue (i.e., an *M/G/c* queue with a maximum of N requests in the system) if the pattern of request arrivals is relatively well behaved and can be represented by a quasi-Poisson process. The exact analytical solution of this queueing model is not known except in some special cases, and a limited number of approximations have appeared in the literature.

Only a small handful of papers are devoted to the approximate computation of the steady-state queue length distribution for an *M/G/c/N* queue. Hokstad (1978) and Miyazawa (1986) use approximate generating functions to obtain the steady-state queue length distribution. However, their numerical results are limited to queues with less than ten servers, and, as pointed out in a critical review by Kimura (1996), these approximations tend to be difficult to implement and evaluate computationally. Tijms *et al.* (1981) approximate the steady-state queue length distribution by considering the system as operating in two regimes: a delay without queueing under light load, and an *M/G/1* queue under heavy load. Consequently, this approach yields less accurate results under moderate loads. In another work, inspired by the existing relationship between the steady-state queue length distribution in the *M/G/1/N* and the *M/G/1* queue (described by Glasserman and Wei-Bo (1991)), Tijms (1992) and later Kimura (1996) propose new approximations for the queue length distribution in the *M/G/c/N* and the *M/G/c* queue. Because of the computational complexity of the latter approaches and/or limited accuracy in some cases, other authors have focused only on certain quantities of interest. For instance, Nozaki and Ross (1978), and Ma and Mark (1995) assume independent residual times to derive approximate values for the expected waiting time or the expected number in the system. Finally, some authors propose heuristics to approximate the loss probability. Schweitzer and Konheim (1978) estimate the overflow probability in an *M/G/c/N* queue based on the property mentioned above for *M/G/1* and *M/G/1/N* queues. Gouweleeuw and Tijms (1996), and more recently, Smith (2003), have proposed an approximation for the loss probability based on only the first two moments of the service time. By their very nature, these latter solutions fail to capture the potentially important dependence of the performance of such a queueing system on higher-order properties of the service time distribution (see Gupta *et al.* (2007), Whitt (1980), Wolff (1977)).

Therefore, when dealing with the *M/G/c/N* queue, outside simulation, a frequently-used approach is to replace the general service time distribution by a phase-type distribution, as it is known that any distribution can be approximated arbitrarily closely by a distribution of the latter type (see Johnson and Taaffe (1988)). The obvious advantage of this approach is that, in steady-state, the resulting *M/Ph/c/N* queue can be described by familiar balance equations. Generally speaking, these balance equations can be obtained using one of two possible state descriptions involving the current number of requests in the system and a vector to represent the state of the servers. The first one uses the vector of the current number of servers in each phase of the service process. In the second possible description, the vector is that of the current phases for each server (note that the servers are assumed to be homogenous but they are not synchronized). This latter state description is generally less thrifty than the first one and rarely, if ever, used. Both descriptions exhibit combinatorial growth as the number of phases and the number of servers grow.

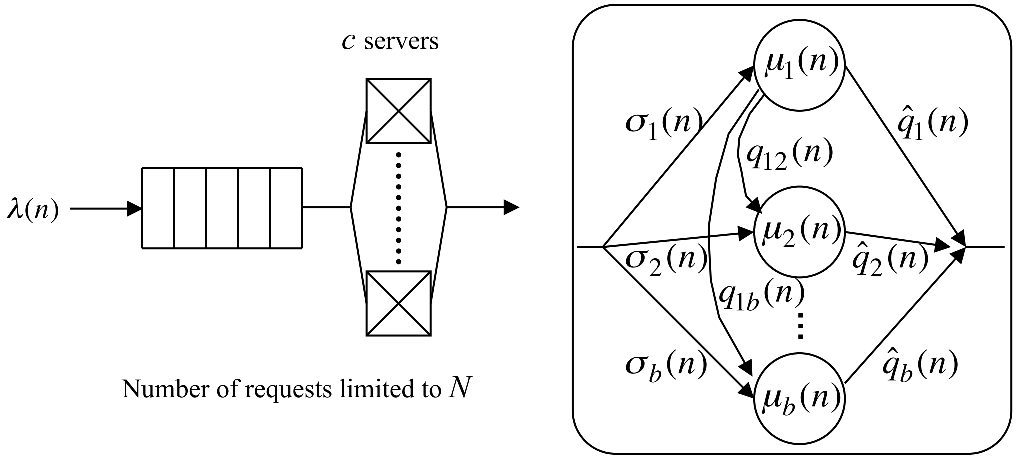
Several methods (e.g., direct iteration Seelen (1986), matrix geometric Ramaswami and Lucantoni (1985), Latouche and Ramaswami (1993), Bini, Latouche and Meini (2005)) can be used to solve these equations numerically. As long as the number of servers and service phases remains small these methods work fine. However, it is also known that the size of the system of equations to be solved suffers from what has been termed the “dimensionality curse”, in that the number of states grows combinatorially as the number of servers and phases increase. Thus, for larger numbers of servers, these methods become impractical, and there is a clear need for an approach that would handle larger numbers of servers (say, hundreds) with a reasonable number of service time phases.

In a related work, van Vuuren and Adan (2005) consider an unrestricted capacity queue with a superposition of non-Markovian arrival streams and general service times. They propose to approximate the solution of their model by aggregating the arrival streams and also the service process of multiple servers. The resulting model is then solved using matrix-geometric methods. Their numerical examples are limited to smaller numbers of servers and a range of coefficients of variation of the service time distributions not exceeding 1. It should be noted that, for this range of service time distributions, existing standard approximations, based on only the first two moments of the service time distribution, (cf. Bolch *et al.* (2005) and Kimura (1994)) produce generally excellent results for the mean number of requests in the system while being much simpler to use. However, these simple approximations fail for service time distributions with larger variability (Begin and Brandwajn (2013)).

Our goal in this paper is to propose a different approach to the approximate evaluation of the *M/Ph/c/N* queue that scales easily with the number of servers and is relatively accurate for a large range of service time distributions. Our approach is based on a reduced state description to circumvent the explosion of the number of states discussed above while taking into account the full service time distribution. Analogous ideas have been successfully applied by various authors to several other problems, e.g., networks with blocking (Altiok and Perros (1986)) or cache replacement schemes (Dan and Towsley (1990)). A similar idea of tracking a selected server has been used in the context of modeling Web servers (Gupta *et al.* (2007)).

2. MODEL, STATE DESCRIPTION AND SOLUTION

Consider the *M/Ph/c/N* queue represented in Figure 1. The times between arrivals are assumed to be memoryless (quasi-Poisson) and the service times are represented as a phase-type distribution with a total of b phases. There are c homogenous servers in our system and the buffer space is restricted to a maximum of N requests in the systems (queued and in service). We assume that $N > c$, since otherwise there would be no queue build up possible. We also assume that the rate of arrivals and the parameters of the service process may depend on the current number of requests in the system, denoted by n. This type of state dependence is useful, in particular, to represent arrivals from a finite number of exponential sources and a service process which varies with the workload. The detailed notation used in our paper is given in Table 1.



The *M/Ph/c/N* queue

The phase distribution with b phases for service times

Figure 1 – *M/Ph/c/N* queue with state dependencies.

We consider the stationary behavior of such a queue. As mentioned in the introduction, the state of our system could be fully described by the total current number of requests in the system and the numbers of requests in each phase of the service process, or, alternatively, by the current total number of requests and the current phase of each server. Instead of such a full state description, we propose to use a reduced state description in which we select one server among the c servers and describe the system by the total number of requests and the current phase of the selected server, (n,i). For $n < c$, with probability $(c-n)/c$ the selected server may be idle, in which case we use the value $i = 0$ to denote its idle state.

Let $p(n,i)$ be the steady-state probability corresponding to this reduced state description. Denote by $\omega(n)$ the rate of completions for the selected server given that the current total number of requests in the system is n. Using our reduced state description, for $n \geq c$ we have

$$\omega(n) = \sum_{i=1}^{b} p(n,i)\mu_i(n)\hat{q}_i(n) / p(n) \tag{1}$$

The steady-state probability that there are n requests in the system, denoted by $p(n)$, can be expressed as

$$p(n) = \sum_{i=0}^{b} p(n,i) \tag{2}$$

Let $u(n)$ be the overall departure rate from the set of c servers given the current number of requests in the system. Since the servers are homogenous, i.e., statistically identical, we must have

$$u(n) = \min(c, n)\omega(n) \tag{3}$$

In other words, equation (3) reflects the fact that, on average, the conditional completion rate of each server is the same.

Using the proposed reduced-state description, the balance equation for $c < n < N$ (non-boundary values of n) and for $i = 1,\ldots,b$ is given by

$$\begin{aligned} &p(n,i)\left[\lambda(n) + \mu_i(n) + \nu(n,i)\right] = \\ &p(n-1,i)\lambda(n-1) + \sum_{j=1}^{b} p(n,j)\mu_j(n)q_{ji}(n) + \sum_{j=1}^{b} p(n+1,j)\mu_j(n)\hat{q}_j(n)\sigma_i(n) + p(n+1,i)\nu(n+1,i) \end{aligned} \tag{4}$$

The corresponding balance equations for other values of n are given in the Appendix.

In the above equation, $\nu(n,i)$ denotes the conditional rate of departures (request completions) by servers other than the selected server given the current state (n,i). Clearly, we need a way to determine $\nu(n,i)$ for our reduced state equations to be of use. We propose to use the following intuitive approximation

$$\nu(n,i) \approx u(n) - \omega(n) \qquad \text{for } i = 1,\ldots,b. \tag{5}$$

For $n \geq c$, the above approximation amounts to assuming $\nu(n,i) \approx (c-1)\omega(n)$. Essentially, we assume that the rate of completions for servers other than the selected server exhibits little dependence on the current service phase of the selected server. Another way of stating our approximation is that we replace $\nu(n,i)$ by its conditional expected value given n. As a result, the conditional rates of departure $\nu(n,i)$ are also expressed as a function of the reduced state description $p(n,i)$.

The details of the computation of $\nu(n,i)$ for other values of n are given in the Appendix.

Thus, using equation (4) (and the corresponding equations in Appendix for other values of n), together with equations (1), (2), (3) and (5) we get a system of equations for $p(n,i)$ which can be solved in several different ways. In our numerical examples, as a proof of concept, we use a simple fixed-point iteration to solve for the set of $p(n,i)$ and hence $u(n)$. Note that our reduced state description results in a two-dimensional Markov chain and therefore does not have a closed-form solution.

Having obtained the $u(n)$, the steady-state probability that there are n requests in the system can also be computed as

$$p(n) = \frac{1}{G}\prod_{k=1}^{n}\frac{\lambda(k-1)}{u(k)}, \qquad\qquad n = 0,1,...,N. \tag{6}$$

Here, G is a normalizing constant such that $\sum_{n=0}^{N} p(n) = 1$.

The proposed reduced-state solution is an approximation (except in the case of exponentially distributed service times) since it replaces the values of the conditional completion rates for servers other than the selected server by their marginal conditional expected values (eq. 5). At this point, we don't have theoretical bounds for the accuracy of our approximation. Numerical results presented in the next section indicate that, in practice, the accuracy is good and, moreover, tends to improve as the number of servers increases.

For our approximation to work, the essential assumption is that of homogeneous (but not synchronized) servers. The assumption of a finite queueing room (buffer size) is motivated by our desire to make the model of practical interest, and is not essential for the reduced state approximation to work. In this paper, we assume a quasi-Poisson arrival process. We believe that it should be possible to extend the idea of reduced state description to more general arrival processes such as MAP (Markovian Arrival Process). This will be the subject of future work.

Our fixed-point iteration, described in the Appendix, was intended primarily as a proof of concept. We don't have a theoretical proof of the existence or uniqueness of its solution, nor do we have a proof of convergence of the iterative scheme. In the literally thousands of cases we studied, it never failed to converge within a reasonable number of iterations (typically a few hundred to a few thousand). The computation at each iteration is quite simple so that the resulting execution speed is very fast, i.e., several orders of magnitude faster than a full state numerical solution or discrete-event simulation.

Clearly, the size of the state-space (n,i) and hence the number of equations to solve is in general far smaller than with either of the two full state descriptions mentioned earlier. Additionally and importantly, it grows only linearly with the number of servers and the number of phases, while the complexity of the full state description grows combinatorially (see Appendix).

b	Number of phases for the service time distribution
c	Number of servers
N	Buffer space, i.e., maximum of requests in the systems (queued and in service)
n	Total current number of requests in the system, $n = 0,...,N$
$\lambda(n)$	Rate of requests arrivals given the current number of requests in the system is n
$\sigma_i(n)$	Probability that service of a request starts in phase i, $i = 1,...,b$ given n
$\mu_i(n)$	Completion rate for phase i of service process given that the current number of requests is n
$q_{ji}(n)$	Probability that service process continues in phase j upon completion of phase i, $j,i = 1,...,b$ given that the current number of requests is n
$\hat{q}_i(n)$	Probability that service process ends (request departs the system) upon completion of phase $i, i = 1,...,b$ given that the current number of requests is n
m_s	Mean service time of a request
c_s	Coefficient of variation of request service time
s_s	Skewness of request service time
$p(n,i)$	Probability that there are n requests in the system and the current phase of the service process is i
$p(n)$	Marginal probability that there are n requests in the system
$u(n)$	Overall departure rate from the set of c servers given that the current number of requests in the system is n
$\omega(n)$	Departure rate from the selected server given that the current number of requests in the system is n (when the server is not idle)
$\nu(n,i)$	Departure rate from servers other than the selected server given that the current number of requests in the system is n and the current phase of the service process at the selected server is i

Table 1 – Notation used in this paper.

3. ACCURACY

In our exploration of the accuracy of the proposed approximation, we consider several sets of values for the number of servers c and the maximum number of requests in the system N, as well as offered load λ . Since performance measures such as the mean number of requests in the system in an *M/Ph/c* queue are known to depend on the shape of the service time distribution (and not only its first two moments) (see Gupta *et al.* (2007), Whitt (1980), Wolff (1977)), we also consider several sets of values for the coefficient of variation of the service time c_S and we build several distributions with different higher-order properties, viz. skewness, denoted by s_S.

To build such distributions we keep the mean service time m_S at 1, and we use the algorithm by Bobbio *et al.* (2005). We explore values of skewness s_S ranging from c_S to 100. Recall that the skewness of a random variable S is considered to be a measure of the asymmetry of the underlying distribution and is defined as $s_S = E[(S-m_s)^3]\big/Var[S]^{3/2}$ where $Var[S]$ denotes the variance of S. Generally, larger values of s_S correspond to longer-tailed distributions. The algorithm used aims to produce a phase-type distribution with the minimum number of phases to match the specified first three moments of the distribution. In our numerical examples, the number of phases varies between 2 and 13 (most frequently around 4).

The performance metrics considered include the mean number of requests in the system (relative error), the loss probability (relative error), the delay probability (relative error), the server utilization (relative error) and the steady-state probability distribution $p(n)$ (mean relative error). We define the relative error (expressed as a percentage) of our reduced state description versus the actual values as the ratio $100 \times (approximate - actual) / actual$. The mean relative error (used for $p(n)$) is defined as the weighted average of the absolute values of the corresponding relative errors. Since we use the state probabilities as weights, this amounts to $100 \times \sum_{n=0}^{N} \left| p(n)_{approximate} - p(n)_{actual} \right|$. The actual values are obtained from a numerical solution of the full-description balance equations for the number of servers $c \leq 64$, and by discrete-event simulation for larger values of c. In the simulation runs we use 7 independent replications with 50,000,000 completions per replication. We only consider relative errors for probabilities exceeding 0.001 when the actual values are obtained from a numerical solution, and 0.01 when the values come from simulation. There are two main reasons for this limitation. First, in most real-life problems system and workload parameters are rarely known to a very high degree of accuracy. Second, for small quantities, it is easy to have large relative errors which do not seem particularly meaningful as a measure of the practical accuracy of an approximation.

We start by a number of examples with state-independent service and simple Poisson arrival process with rate λ. In describing our results, we use the notion of offered load per server defined as the ratio λ / c.

In an attempt to give a comprehensive view of the accuracy of the proposed approximation, each figure corresponds to hundreds of data points explored, and the surfaces shown are obtained using an interpolation from sets of scattered data points. Additionally, we present overall error distribution tables over close to 10,000 experiments in which the number of servers c spans the range of 4 to 256, the buffer size N varies between 8 and 1024, the coefficient of variation of the service time distribution c_s spans the range of 0.4 to 7, and the offered load per server λ / c is between 0.2 and 2.

We start by noting that the reduced-state approximation produces virtually flawless results when the coefficient of variation of the service time distribution our approximation does not exceed 1. Errors tend to appear for larger values of the coefficient of variation of the service time and their extent may depend on the skewness of the service time distribution. We illustrate this dependence in Figure 2 for the relative error in the mean number in the system with $c = 32$ servers, the buffer size $N = 64$ and the offered load per server kept at 0.8 ($\lambda = 0.8 \times c$). We observe that the relative error tends to increase as the coefficient of variation of the service time c_s increases, especially for smaller values of the skewness s_s of the service time distribution. Overall, the relative error for the mean number in the system for the example considered remains below 5% even when the coefficient of variation of the service time distribution reaches 7. Note that in a non-negative distribution with a given coefficient of variation c_s the skewness s_s must be greater than a certain value (see Appendix). This is the reason behind the white “impossible area” band in Figure 2. Table 2 gives the average relative error in this example for a selected set of values for the coefficient of variation of the service time c_s.

Because the accuracy of the approximation tends to be uniformly excellent for smaller coefficients of variation of the service time distribution, we devote most of our attention to coefficients of variation of the service time above 1.

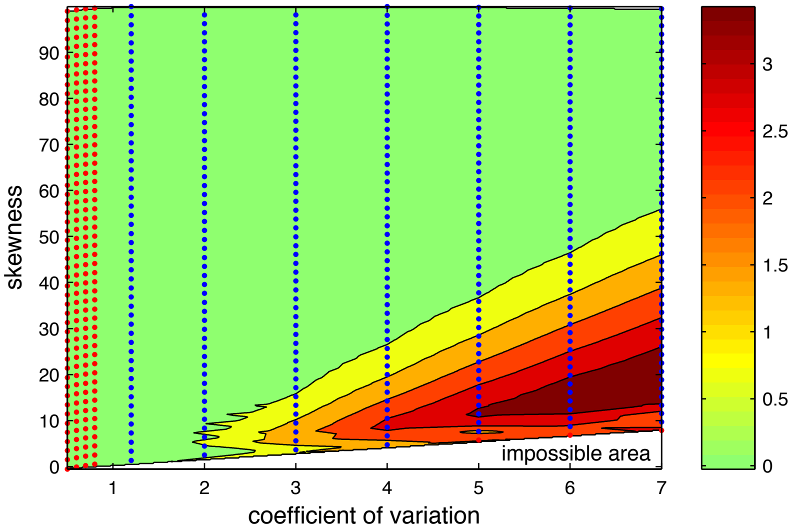


Fig. 2 – Percentage relative errors of the approximate solution for the mean number of requests in *M/Ph/c/N* queue with $c = 32$, $N = 64$ and $\lambda = 0.8 \times c$.

Coefficient of variation c_S	0.5	1	2	3	4	5	6	7
Average error (in percent)	0.10%	0%	0.04%	0.20%	0.46%	0.75%	1.08%	1.40%

Table 2 – Accuracy of the approximate solution for the mean number of requests in *M/Ph/c/N* queue with $c = 32$, $N = 64$ and $\lambda = 0.8 \times c$ as a function of the coefficient of variation c_S.

In Figure 3, we show an example of the relative error for the mean number in the system for an *M/Ph/c/N* queue with a buffer space $N = 4 \times c$ and a large coefficient of variation $c_s = 4$ as a function of the number of servers and of the offered load per server. We note that the relative error in this example appears to be the highest when the offered load per server is around 0.9 and it decreases relatively rapidly as the number of servers increases. Table 3 shows the overall relative errors for the mean number in the system for a large set of experiments. We observe that in almost 95% of the cases considered the relative error is below 5% and exceeds 10% in less than 2% of cases. The mean relative error is 0.9%.

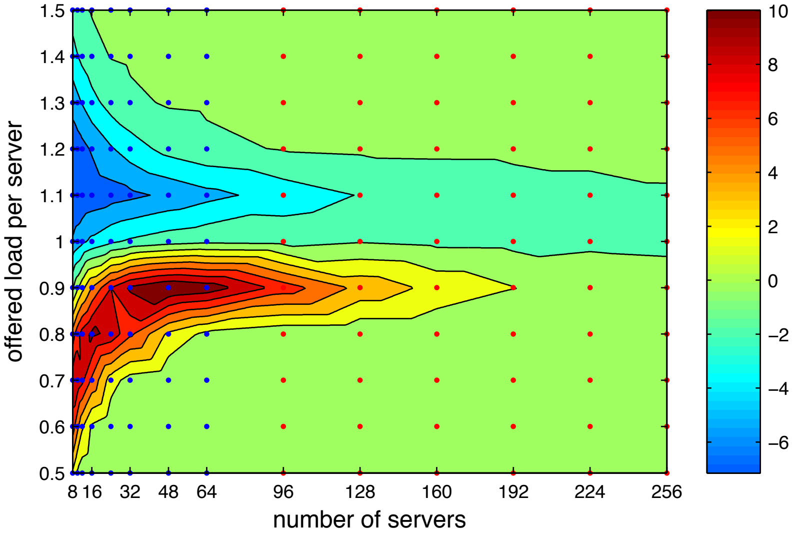


Fig. 3 – Percentage relative errors of the approximate solution for the mean number of requests in *M/Ph/c/N* queue with $c_S = 4,\ s_s = 15$ and $N = 4 \times c$

Average	Median	<5%	5-10%	10-15%	>15%
0.87%	0.03%	94.98%	3.05%	0.97%	1.00%

Table 3 – Overall accuracy of the approximate solution for the mean number of requests in *M/Ph/c/N* queue.

Figure 4 and Table 4 are devoted to the relative error in the loss probability (i.e., the probability that an arriving request finds the buffer full). Figure 4 shows the relative error in the loss probability as a function of the number of servers and of the offered load per server for a set of parameters with a relatively small buffer space $N = c + 10$ and a large coefficient of variation $c_s = 4$. In this example, the relative error in the loss probability rarely exceeds 2%. Table 4 shows the overall distribution of relative errors in the loss probability for a large set of experiments. We observe in Table 4 that overall the mean error is below 9%, in some 87% of cases the error remains below 5% and it exceeds 10% in about 11% of the cases considered. Typically, the infrequent larger relative errors tend to happen when the loss probability is close to or below 0.01. The influence of these larger relative errors on the mean is clear from the value of the median error which is below 0.25%.

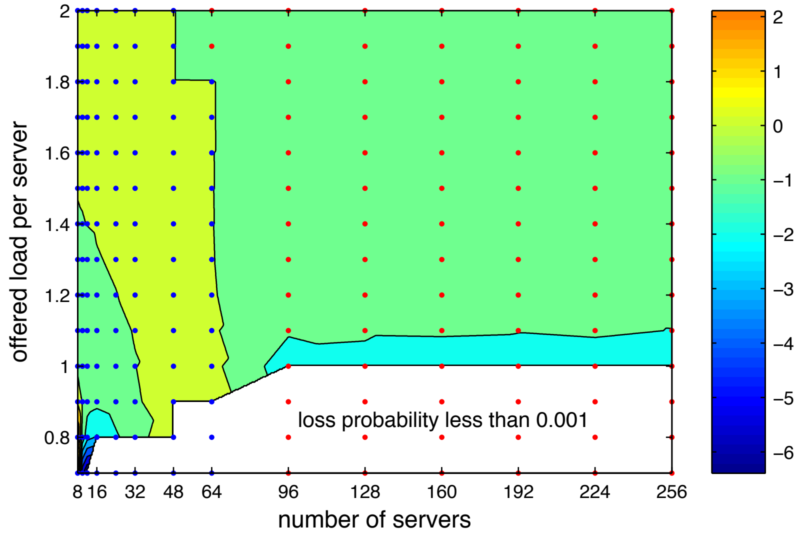


Fig. 4 – Percentage relative errors of the approximate solution for the loss probability in *M/Ph/c/N* queue with $c_S = 4$, $s_S = 60$ and $N = c + 10$.

Average	Median	<5%	5-10%	10-15%	>15%
8.6%	0.24%	87.19%	2.32%	1.66%	8.83%

Table 4 – Overall accuracy of the approximate solution for the loss probability in *M/Ph/c/N* queue.

In Figure 5 and Table 5, we examine the relative error in the delay probability (i.e., the probability that arriving requests find all servers busy). Figure 5 illustrates the relative error in the delay probability as a function of the number of servers c and of the size of the buffer space N for a coefficient of variation of the service time $c_s = 3$ and offered load per server of 1 (i.e., $\lambda = c$). As mentioned above, this value of the offered load corresponds to largest errors. Nonetheless, we note that the relative error for the delay probability, in this example, remains well under 5%. It is interesting to note that there seems to be no clear trend in the relative error as a function of the buffer space N. Table 5 shows the overall distribution of relative errors in the delay probability for a large number of experiments. We observe that in 93% of the cases considered the relative error remains below 5% and it exceeds 10% in less than 4% of cases. The mean of the relative error for the delay probability is less than 2%.

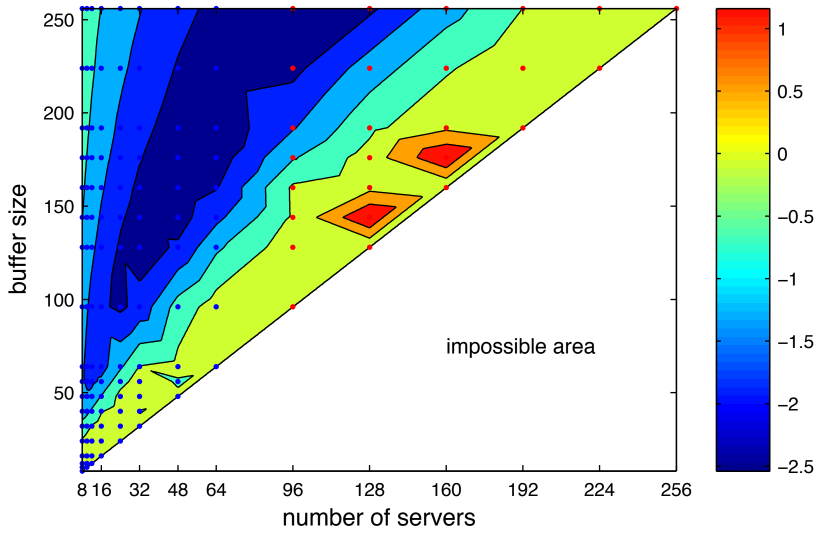


Fig. 5 – Percentage relative errors of the approximate solution for the delay probability in *M/Ph/c/N* queue with $c_S = 3,\ s_s = 60$ and $\lambda = 1 \times c$.

Average	Median	<5%	5-10%	10-15%	>15%
1.89%	0.20	92.95%	3.60%	1.22%	2.23%

Table 5 – Overall accuracy of the approximate solution for the delay probability in *M/Ph/c/N* queue.

Figure 6 and Table 6 show the relative error in the per server utilization. Figure 6 illustrates the behavior of the reduced-state approximation for the set of parameters with a large coefficient of variation $c_s = 4$ and a buffer space $N = c + 10$ used in Figure 4. Table 6 gives the overall distribution of relative errors in the per server utilization. We observe that the relative errors in the per server utilization tend to be low, below 5% in over 98% of cases considered. They exceed 10% in less than 0.5% of cases, and the mean relative error is below 0.5%.

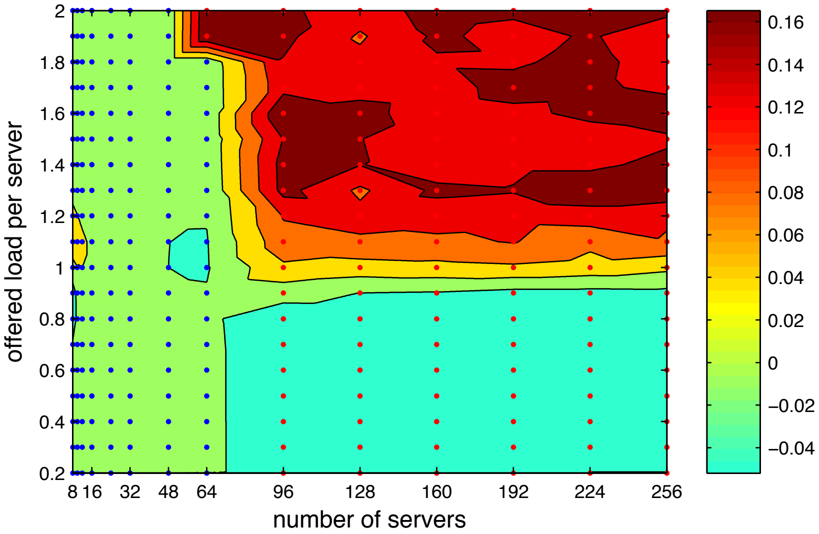


Fig. 6 – Percentage relative errors of the approximate solution for per server utilization in *M/Ph/c/N* queue with $c_S = 4,\ s_s = 60$ and $N = c + 10$.

Average	Median	<5%	5-10%	10-15%	>15%
0.30%	0.01%	98.67%	1.19%	0.14%	0%

Table 6 – Overall accuracy of the approximate solution for per server utilization in *M/Ph/c/N* queue.

Figure 7 and Table 7 illustrate the ability of the reduced-state approximation to reproduce the shape of steady-state probability distribution $p(n)$. For each set of parameters (i.e., for each steady-state distribution $p(n)$), we consider the absolute values of the relative errors in the $p(n)$ for all $n = 0,\ldots,N$, and we use the weighted average of these values, where the state probabilities $p(n)$ are the weights, as the relevant metric. We refer to this metric of deviation between the exact and approximate values of $p(n)$ as the mean relative error for $p(n)$ since it is the expected value of (the absolute values of) the relative errors for each $n = 0,\ldots,N$. As mentioned at the beginning of this section, this amounts to using $100 \times \sum_{n=0}^{N} \left| p(n)_{approximate} - p(n)_{actual} \right|$ as the mean relative error. Since it is difficult to estimate very small values in discrete-event simulation, we limit the number of servers to 64, value for which we are still able to obtain exact numerical results for the actual state probabilities.

Figure 7 shows an example of the mean relative errors as a function of the number of servers and of the offered load per server for a set of model parameters with the coefficient of variation of the service time $c_s = 3$. We observe that the mean relative error for the steady-state distribution $p(n)$ varies from close to 0 to less than 10% depending on the offered load. We also observe that the largest relative errors tend to occur when the offered load per server is close to 1 and they tend to decrease as the number of servers increases. Table 7 shows the overall mean relative errors over a large number of experiments. It is worthwhile noting that, in general, our approximation correctly reproduces the shape of the steady-state distribution $p(n)$. The average error is below 4%, and the errors remain below 5% in some 80% of the cases considered.

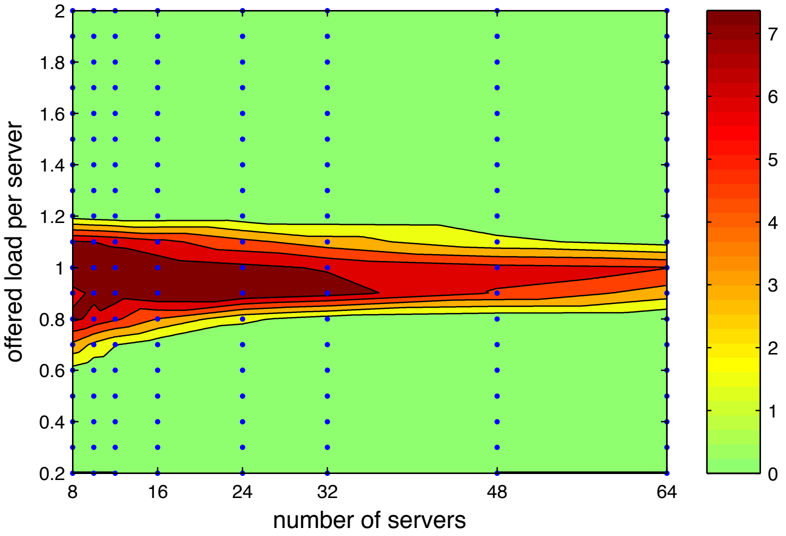


Fig. 7 – Percentage mean relative errors of the approximate solution for $p(n)$ in *M/Ph/c/N* queue with $c_S = 3,\ s_s = 60$ and $\lambda = 1 \times c$.

Average	Median	<5%	5-10%	10-15%	>15%
3.76%	0.43%	80.42%	7.64%	3.17%	8.77%

Table 7 – Overall accuracy of the approximate solution for $p(n)$ in *M/Ph/c/N* queue.

Our results illustrate the overall good accuracy of the proposed reduced-state approximation. As noted at the beginning of this section, we have shown few results for coefficients of variation of the service time below 1 because our approximation produces excellent results in those cases. Errors tend to be larger for large values of the coefficient of variation, say, over 7, in particular when the skewness of the service time distribution is small. Even there, errors decrease as the number of servers increases. The size of the queueing room does not appear to have a uniform easy-to-characterize effect on the accuracy of the results.

The proposed approximation method works well also when the rate of arrivals and the service phases depend on the current number of requests in the system n. This is illustrated in Figure 8 where we compare the results obtained for the mean number of requests in the system using our approximation with those of an exact numerical solution as a function of the offered load. These results correspond to a system with 32 servers ($c = 32$) and a queueing room of 128 ($N = 128$). The service process has a coefficient of variation of 3 ($c_s = 3$) with a skewness of 30. The completion rate of each service phase depends on the current number of requests in the system through a dependency of the form $s(n) = an + b$ where the coefficients a and b are chosen so that $s(1) = 1$ and $s(N) = 0.7$, i.e., the service rate degrades linearly from full speed for a single user down to 70% of its nominal speed as the current number of requests in the system increases. The rate of arrivals of the quasi-Poisson arrival process is given by $\lambda(n) = \phi \times (1 - n/(2N))$. This workload dependency can be viewed as representing a set of $2N$ identical exponential request sources. Thus the model is an instance of the machine repairman model with state-dependent repair rates. We observe in Figure 8, which shows the mean number in the system as a function of the maximum offered load ϕ, the very good accuracy of our approximate results for this example.

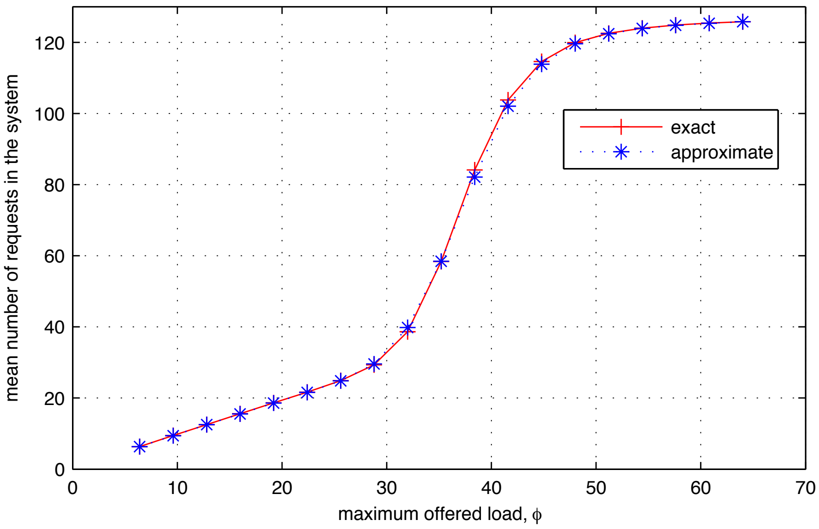


Fig. 8 – Mean number of requests in *M/Ph/c/N* queue with $c = 32$, $N = 4 \times c$, $c_S = 3$, $s_s = 30$ and state dependencies for the service process and the arrival process for various levels of workload.

Examining the reasons for deviations between the exact values and those produced by our approach, we note that the only approximation in our reduced state solution is in the computation of the conditional completion rates of "other" servers given the number of requests in the system and the current phase of the selected server $\nu(n,i) \approx u(n) - \omega(n)$ (our formula (5)) and $\nu(n,0) \approx u(n)$ (see Appendix). It is not surprising then that the deviations tend to decrease as the number of servers increases. Indeed, the knowledge of the current phase of just one out of many servers does not convey much information about the state of the other servers.

As mentioned before, it is clear that the number of states in our reduced state description increases linearly with the number of servers and phases (see Appendix).

4. CONCLUSION

This paper presents an approach, believed to be novel in the context of its application, to the solution of a queueing system with multiple homogenous servers, quasi-general service times (phase-type distributions), quasi-Poisson arrivals and a limited buffer space (queueing room). The proposed approach uses a reduced state description in which the state of only one server is represented explicitly while the other servers are accounted for through their rate of completions.

We solve the resulting system of balance equations using a fixed-point iteration. We do not have a theoretical proof of existence and uniqueness of the solution or convergence of the fixed-point scheme. Our reduced state approach involves an approximation, and, unfortunately, we have not been able to obtain theoretical bounds on its accuracy.

In practice, in literally tens of thousands of varied examples, the fixed-point iteration never failed to converge with a reasonable speed. The accuracy of our approximation is generally good and, importantly, tends to improve as the number servers in the system increases. This conclusion is supported by a large number of data points. Additionally, the proposed reduced state approximation is intuitive and quite simple to implement in a standard programming language. It is also thrifty in terms of memory requirements and its execution speed is fast.

In the classical state description used for this type of queueing system, the number of states grows combinatorially, making the problem intractable for larger numbers of servers and/or phases. By contrast, the computational complexity in terms of the number of states in our reduced state description grows only linearly in the number of servers and phases. This puts problems with hundreds of servers and several phases within easy reach of a fast numerical solution.

The proposed approach appears applicable to other problems with multiple servers and quasi-general service times such as systems with quasi-general arrival times, systems with priorities, etc.

REFERENCES

Altiok, T., and Perros, H.G. 1986. Open Networks of Queues with Blocking: Split and Merge Configurations*, AIIE Trans*,. Vol 18, pp. 251-261.

Begin, T., and Brandwajn, A. 2013. A note on the accuracy of several existing approximations for *M/Ph/m* queues, *HSNCE*, Kyoto, Japan.

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

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

Dan, A., and Towsley, D. 1990. An approximate analysis of the LRU and FIFO buffer replacement schemes, *ACM SIGMETRICS Performance Evaluation Review*, Vol. 18(1), pp.143-152.

Glasserman, P., and Wei-Bo G. 1991. Time-changing and truncating K-capacity queues from one K to another, *Journal of Applied Probability,* Vol. 28 (3), pp. 647-655.

Gouweleeuw, F.N., and Tijms, H. 1996.A simple heuristic for buffer design in finite-capacity queues, *European Journal of Operational Research*,Vol. 88 (3), pp. 592-598.

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

Gupta, V., Harchol-Balter,M., Sigman,K., and Whitt,W. 2007. Analysis of Join-the-Shortest-Queue Routing for Web server Farms, *Performance Evaluation*, Vol. 64 (9-12), pp. 1062-1081.

Hokstad, P. 1978. Approximations for the *M/G/m* Queue, *Operations Research*, Vol. 26 (3), pp. 510-523.

Johnson, M.A., and Taaffe, M.R. 1988. The denseness of phase distributions. *School of Industrial Engineering*, Purdue University.

Kimura, T. 1994. Approximations for multi-server queues: system interpolations, *Queueing Systems*, Vol. 17, 1994, pp. 347-382.

Kimura, T. 1996. A transform-free approximation for the finite capacity *M/G/s* queue, *Operations Research*,Vol. 44 (6), pp. 984-988.

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

Bini, D., Latouche, G., and Meini, B. 2005. *Numerical methods for structured Markov chains*. Oxford: Oxford University Press.

Ma, B.N.W., and Mark, J.W. 1995. Approximation of the Mean Queue Length of an *M/G/c* Queueing System, *Operations Research*, Vol. 43 (1), *Special Issue on Telecommunications Systems: Modeling,* *Analysis and Design*, pp. 158-165.

Miyazawa, M. 1986. Approximation of the Queue-Length Distribution of an*M/GI/s* Queue by the Basic Equations, *Journal of Applied Probability*, Vol. 23 (2), pp. 443-458.

Nozaki, S.A., and Ross, S.M. 1978. Approximations in Finite-Capacity Multi-Server Queues with Poisson Arrivals, *Journal of Applied Probability*, Vol. 15 (4), pp. 826-834.

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

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

Schweitzer, P., and Konheim, A. 1978. Buffer overflow calculations using an infinite-capacity model, *Stochastic Processes and Their Applications*, Vol. 6, pp. 267–276.

Seelen, L.P. 1986. An Algorithm for *Ph/Ph/c* Queues, *European Journal of the Operations Research Society*, Vol. 23, pp. 118-127.

Smith, J.M. 2003. *M/ G/c/K* blocking probability models and system performance, *Performance Evaluation*, Vol. 52 (4), pp. 237–267.

Tijms, H., Van Hoorn, M.H., and Federgruen, A. 1981. Approximations for the Steady-State Probabilities in the *M/G/c* Queue, *Advances in Applied Probability*, Vol. 13 (1), pp. 186-206.

Tijms, H. 1992. Heuristics for finite-buffer queues, *Probability in the Engineering and Informational Sciences,* Vol. 6 (3), pp. 277-285.

van Vuuren, M., and Adan, I. 2005. Approximating the *Σ GI/G/s* queue by using aggregation and matrix analytic methods, *Stochastic Models*, Vol. 21 (2-3), pp. 767-784.

Whitt, W. 1980. The effect of variability in the *GI/G/s* queue, *Journal of Applied Probability*, Vol. 17 (4), pp. 1062-1071.

Wolff, R.W. 1977. The Effect of Service Time Regularity on System Performance, *Computer Performance*, North Holland, pp. 297-304.

ACKNOWLEDGEMENTS

The authors wish to thank the anonymous referees for their thorough and constructive review of an earlier version of this paper.

APPENDIX

A.1 Balance equations for the reduced-state description

For completeness, we give here the reduced state description balance equations for the cases not explicitly treated in the body of the paper, starting with the case $0 < n < c.$

$$p(n,i)\left[\lambda(n)+\mu_i(n)+v(n,i)\right]=p(n-1,0)\lambda(n-1)\sigma_i(n)/(c-n+1)+p(n-1,i)\lambda(n-1)+$$
$$\sum_{j=1}^{b}p(n,j)\mu_j(n)q_{ji}(n)+p(n+1,i)\nu(n+1,i), \qquad i=1,\ldots,b$$

$$p(n,0)\left[\lambda(n)+v(n,0)\right]=\sum_{i=1}^{b}p(n+1,i)\mu_i(n)\hat{q}_i(n)+p(n+1,0)\nu(n+1,0)+p(n-1,0)\lambda(n-1)(c-n)/(c-n+1).$$

Note that we use the notation $p(n,0)$ to denote the case when the selected server is idle, which, in our model, can only happen if $n<c.$ Note also that we have $p(n=0,i\neq 0)=0$ and $\nu(n=1,i\neq 0)=0.$

In the case $n=0$, we can only have

$$p(0,0)[\lambda(0)]=\sum_{i=1}^{b}p(1,i)\mu_i(0)\hat{q}_i(0)+p(1,0)\nu(1,0).$$

For $n=c$ we have

$$p(c,i)\left[\lambda(c)+\mu_i(c)+v(c,i)\right]=p(c-1,0)\lambda(c-1)\sigma_i(c)+p(c-1,i)\lambda(c-1)+\sum_{j=1}^{b}p(c,j)\mu_j(c)q_{ji}(c)+p(c+1,i)v(c+1,i)$$
$$+\sum_{j=1}^{b}p(c+1,j)\mu_j(c)\hat{q}_j(c)\sigma_i(c), \qquad i=1,\ldots,b.$$

Finally, for $n=N$ we obtain

$$p(N,i)\left[\mu_i(N)+\nu(N,i)\right]=p(N-1,i)\lambda(N-1)+\sum_{j=1}^{b}p(N,j)\mu_j(N)q_{ji}(N), \qquad i=1,\ldots,b.$$

The conditional rate of completions for the selected server when it is not idle and there are n requests in the system is given by $\omega(n)=\sum_{i=1}^{b}p(n,i)\mu_i(n)\hat{q}_i(n)/\sum_{j=1}^{b}p(n,j)$ and the conditional rate of request completion given n can be expressed as $u(n)=\min(n,c)\omega(n).$

As before, we approximate the conditional rates of departure $\nu(n,i)$ for $i=1,,,b$ by $\nu(n,i)\approx u(n)-\omega(n).$ For $0<n<c,$ when the selected server is idle, we use $\nu(n,0)\approx u(n)$.

A.2 Constraints on skewness

Let Z be a non-negative random variable, and denote by m_i its i-th moment $E[Z^i]$ and by n_i its i-th normalized moment. We have $n_2 = m_2 / m_i^2$ and $n_3 = m_3 / (m_1 m_2)$, and it is has been shown that (see Osogami and Harchol-Balter (2006)) $n_3 \geq n_2$. It follows that the skewness of Z, denoted by s_Z, must satisfy the relationship

$$s_z \geq c_Z - 1 / c_Z \ ,$$

where $c_Z = (m_2 / m_1^2 - 1)^{1/2}$ and $s_Z = (m_3 - 3m_1 m_2 + 2m_1^3) / (m_2 - m_1^2)^{3/2}$. c_Z is the coefficient of variation of the random variable Z.

A.3 Fixed-point iteration

The system of equations for $p(n,i)$ can be solved in a number of ways. Using a superscript to denote the iteration number (k), one possible straightforward approach is to use a single array to store the values of the state probabilities $p^k(n,i)$ in which any newly computed value immediately replaces the value from the previous iteration, and a single array for the request completion rates $u^k(n)$. The steps of our approach are:

Step 1. Select an initial distribution $p^0(n,i)$ and compute the corresponding values for the conditional request completion rates $u^0(n)$.

Step 2. At iteration $k, k = 1,2,...$, enumerate states in the order of increasing values of n and, for each n, in the order $i = 0,1,...,b$ to compute non-normalized values $\tilde{p}^k(n,i)$ directly from the corresponding balance equations as

$$\tilde{p}^k(0,0) = \frac{1}{\lambda(0)} \left[\sum_{i=1}^{b} p^{k-1}(1,i) \mu_i(0) \hat{q}_i(0) + p^{k-1}(1,0) \nu(1,0) \right]$$

For $n = 1,...,c-1$

$$\tilde{p}^k(n,0) = \frac{1}{\lambda(n) + u^{k-1}(n)} \left[\sum_{i=1}^{b} p^{k-1}(n+1,i) \mu_i(n) \hat{q}_i(n) + p^{k-1}(n+1,0) u^{k-1}(n+1) + \tilde{p}^k(n-1,0) \lambda(n-1)(c-n)/(c-n+1) \right]$$

$$\tilde{p}^k(n,i) = \frac{1}{\lambda(n)+\mu_i(n)+(n-1)\omega^{k-1}(n)}\left[\begin{array}{l}\tilde{p}^k(n-1,0)\lambda(n-1)\sigma_i(n)/(c-n+1)+\tilde{p}^k(n-1,i)\lambda(n-1) \\ +\sum\limits_{j=1}^{i-1}\tilde{p}^k(n,j)\mu_j(n)q_{ji}(n)+\sum\limits_{j=i}^{b}p^{k-1}(n,j)\mu_j(n)q_{ji}(n)+p^{k-1}(n+1,i)n\omega^{k-1}(n+1)\end{array}\right],$$

$i = 1,...,b.$

For $n = c$

$$\tilde{p}^k(c,i) = \frac{1}{\lambda(c)+\mu_i(c)+(c-1)\omega^{k-1}(c)}\left[\begin{array}{l}\tilde{p}^k(c-1,0)\lambda(c-1)\sigma_i(c)+\tilde{p}^k(c-1,i)\lambda(c-1)+\sum\limits_{j=1}^{i-1}\tilde{p}^k(c,j)\mu_j(c)q_{ji}(c) \\ +\sum\limits_{j=i}^{b}p^{k-1}(c,j)\mu_j(c)q_{ji}(c)+p^{k-1}(c+1,i)(c-1)\omega^{k-1}(c+1) \\ +\sum\limits_{j=1}^{b}p^{k-1}(c+1,j)\mu_j(c)\hat{q}_j(c)\sigma_i(c)\end{array}\right]$$

$i = 1,...,b.$

For $n = c+1,...,N-1$

$$\tilde{p}^k(n,i) = \frac{1}{\lambda(n)+\mu_i(n)+(c-1)\omega^{k-1}(n)}\left[\begin{array}{l}p^k(n-1,i)\lambda(n-1)+\sum\limits_{j=1}^{i-1}\tilde{p}^k(n,j)\mu_j(n)q_{ji}(n)+\sum\limits_{j=i}^{b}p^{k-1}(n,j)\mu_j(n)q_{ji}(n) \\ +\sum\limits_{j=1}^{b}p^{k-1}(n+1,j)\mu_j(n)\hat{q}_j(n)\sigma_i(n)+p^{k-1}(n+1,i)(c-1)\omega^{k-1}(n+1)\end{array}\right]$$

$i = 1,...,b.$

For $n = N$

$$\tilde{p}^k(N,i) = \frac{1}{\mu_i(N)+(c-1)\omega^{k-1}(N)}\left[\tilde{p}^k(N-1,i)\lambda(N-1)+\sum_{j=1}^{i-1}\tilde{p}^k(N,j)\mu_j(N)q_{ji}(N)+\sum_{j=i}^{b}p^{k-1}(N,j)\mu_j(N)q_{ji}(N)\right],$$

$i = 1,...,b.$

Step 3. Compute normalized values $p^k(n,i)$

$$p^k(n,i) = \tilde{p}^k(n,i)/\sum_{n=0}^{N}\sum_{i=0}^{b}\tilde{p}^k(n,i)$$

Step 4. Compute the corresponding values of the conditional request completions rates given n

$$\omega^k(n) = \sum_{i=1}^{b}p^k(n,i)\mu_i(n)\hat{q}_i(n)/\sum_{i=1}^{b}p^k(n,i) \text{ and } u^k(n) = \min(n,c)\omega^k(n).$$

If $\max_n \left|1 - u^{k-1}(n) / u^k(n)\right| < \varepsilon$ where ε is the desired convergence stringency, continue with Step 5 else go back to Step 2.

In our numerical studies, we used $\varepsilon = 10^{-8}$.

Step 5. Use the values for $u(n)$ obtained from the iteration to compute the steady-state distribution $p(n)$ from formula (6), as well as any derived performance metrics.

This simple-minded iteration was only intended as a proof of concept. Because it uses directly the balance equations for the reduced state together with the standard normalization condition $\sum_{n=0}^{N}\sum_{i=0}^{b} p(n,i) = 1$, it seems clear that, if it converges, it must converge to the unique solution of the balance equations for $p(n,i)$. We do not have a theoretical proof of its convergence.

The number of iterations needed to attain convergence varies from a few hundred to several thousand depending on the service time distribution and the number of servers. It does not seem particularly sensitive to the offered load λ or the buffer size N. As illustrated in Figure 9 for a variant of the fixed-point scheme described above, the number of iterations clearly increases as the number of servers c or the coefficient of variation of the service time c_s increases. Since the computational effort at each iteration is quite limited, the overall execution speed remains very fast even for hundreds of servers.

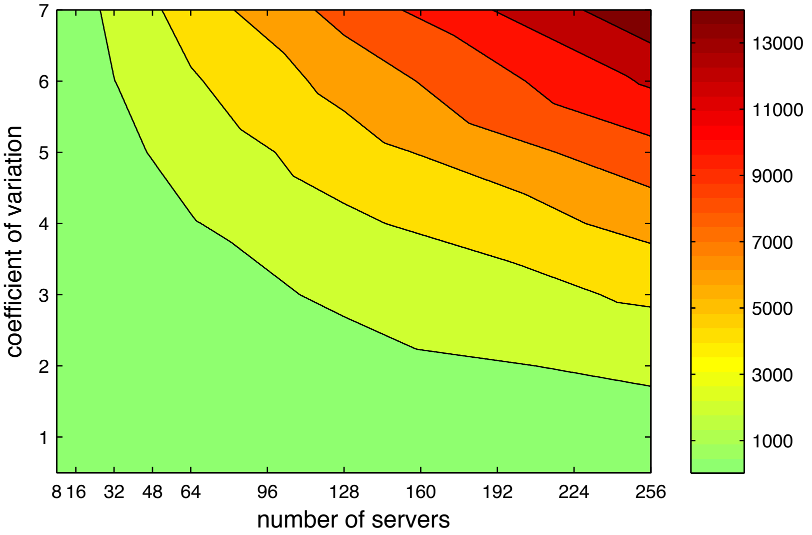


Fig. 9 – Number of iterations for the approximate solution in *M/Ph/c/N* queue with $s_S = 1.5 \times c_S$, $\lambda = 0.8 \times c$ and $N = 2 \times c$.

A.4 Complexity

It is clear that the number of states in the proposed reduced state approximation grows linearly with the number of servers and phases in the service distribution, as opposed to combinatorially in the standard full state description. Figures 10a and 10b illustrate this point.

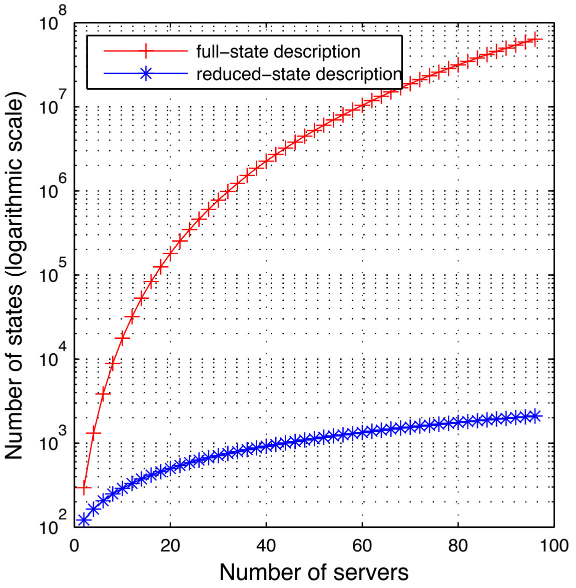


Fig.10a – Comparison between the number of states in the full and the reduced state description for *M/Ph/c/N* queue with $b = 4$ and $N = 5 \times c + 20$.

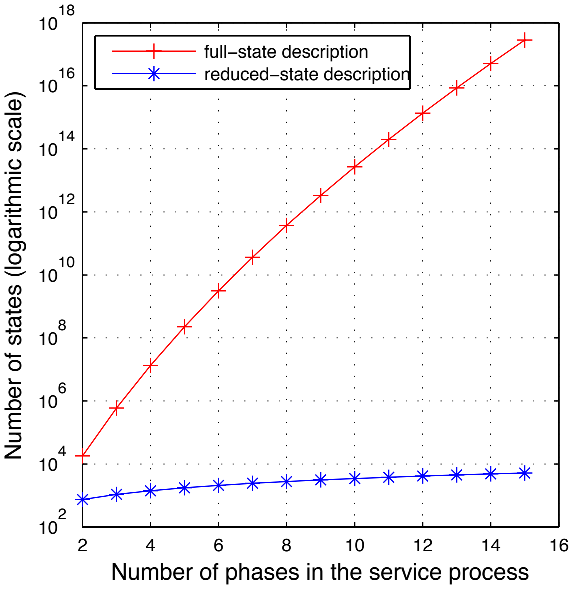


Fig.10b – Comparison between the number of states in the full and the reduced-state description for *M/Ph/c/N* queue with $c = 64$ and $N = 5 \times c + 20$.