Queueing Models and
Service Management
Vol. 7, No. 1, page 1–24, 2024

QMSM

Transition Probabilities of the $GI/M/c$ and $GI^X/M/c/N$ Models

Muhammad El-Taha* and Thomas Michaud

Department of Mathematics and Statistics

University of Southern Maine, Portland, ME 04104-9300, USA

(*Received January 2023; accepted November 2023*)

Abstract: Consider the batch-arrival $GI^X/M/c/N$ model with c servers, general inter-arrival batch times, finite buffer, and exponential service times. Inter-arrival batch times, batch sizes, and service times are $i.i.d.$ and independent of each other. In this article we give a simple efficient method to derive the one-step transition probabilities of the imbedded Markov chain observed at the system arrival epochs of the corresponding $G/M/c$ model. The one-step transition probabilities are computed exactly by converting a numerical integration problem into a finite sum. Another key contribution is generating the transition probabilities of the batch-arrival model by using a simple and intuitive method to extend the results of the standard $GI/M/c$ model to batch arrivals with and without a finite buffer, and in the case of finite buffer with partial and full batch rejection. We give examples to demonstrate the performance of our method.

Keywords: $GI/M/c$ model, $GI^X/M/c/N$ model, transition probabilities, batch arrivals, general arrival process, multi-server.

1. Introduction

In this article, using the $GI/M/c$ model, we develop a simple efficient algorithm to accurately compute the one-step transition probabilities of the imbedded Markov chain where the imbedding points are the arrival epochs. Moreover, we use an innovative approach to extend the results of $GI/M/c$ queueing model to the $GI^X/M/c/N$ multi-server batch arrival finite capacity model. We cover the finite and the infinite buffer cases, and for the finite buffer case we include models with partial and full batch rejection. Specifically, in this article we give a remarkably simple algorithm to derive an accurate solution for the stationary one-step probabilities and give examples to demonstrate the performance of our method. The $GI/M/c$ model and its variants are important building blocks in a wide range of queueing applications that include call centers, health-care, computer and communication, transportation, and manufacturing systems, among others. See Gontijo et al. [9] for list of references on these and other applications.

* Corresponding author
Email: el-taha@maine.edu

The well known multi-server $GI/M/c$ queue is an important basic model in queueing theory. This queueing model, with and without finite buffer, is studied by several authors. Takács [21] and [22] is perhaps the first author to study the $G/M/c$ model using the imbedded Markov chain approach. He develops a computational recursion that uses generating functions techniques. This method is described by Gross and Harris [11] and Kleinrock [15] as "extremely long" and "complex". All textbooks that deal with this model require the numerical integration of at least one transition probability expression of the imbedded Markov chain. See for example Gross and Harris [11], Kleinrock [15], Medhi [17], Ross [19], and Tijms [23]. Others use approximation methods like Cosmetatos and Godsave [3]. Moreover, using the supplemental variable approach Hokstad [12] studies the $GI/M/c$ model with finite waiting room, and provides relations between pre-arrival, post departure, and time-average probabilities. See also Yao et al. [24] who uses level crossing methods to relate similar quantities. Zhao [25] expresses the generating function of the stationary distribution in closed form. Gontijo et al. [9] estimate the inter-arrival times using the kernel method and evaluate the performance measures using algorithmic methods. Cruz et al. [4] derive estimates for certain system measures under finite sampling setting. Neuts [18] uses a matrix analytic approach to investigate this and related models. Grassmann and Tavakoli [10] review multiple numerical approaches and address stability issues. Kim and Chaudhry [14] study the finite capacity $GI/M/c/N$ model. See also Ferreira and Pacheco [8].

The batch arrival $GI^X/M/c$ model is studied by Chaudhry and Kim [2]. They start with the balance equations for the imbedded Markov chain, write the characteristic equation, solve for the characteristic equation roots using $MAPLE$, and use these roots to compute the arrival probabilities. A supplemental variable approach is used by Laxmi and Gupta [16] to solve for the finite capacity $GI^X/M/c/N$ model in order to relate pre-arrival and time-average probabilities. They then use the imbedded Markov chain approach to obtain the pre-arrival probabilities, focusing only on inter-arrival distribution functions whose Laplace transform can be analytically expressed like the Erlang and the hyper-exponential distribution functions. Matrix analytic methods pioneered by Neuts [18] are also used to study this model and its variants. The solutions approach uses algorithmic methods that utilize matrix algebra and vectors. See also Bailey and Neuts [1] who develop algorithmic methods using a modified geometric form for the study of the batch arrival $GI^X/M/c$ model. *All these methods need the one-step transition probabilities stated in Lemma 2.1, where the transition probabilities in Lemma 2.1 (iii) are approximated using numerical integration methods.*

Computing the one-step transition probabilities in the case of a transition from a state $i \geq c$ to a state $j \leq c-1$ requires tedious numerical integration. This is the key step in most approaches in the literature. In this article, the first contribution is a result that converts this expression into a finite sum. By itself, this result will make most approaches used in the literature function more efficiently. This and related results facilitate the development of a simple stable algorithm to efficiently compute the pre-arrival and time average probability distributions. Another key contribution is extending the $G/M/c$ standard multi-server model results to the $GI^X/M/c/N$ batch arrival model with and without finite capacity. We use a novel method to transform the transition probabilities of the standard multi-server model to the batch arrival model. We note that El-Taha [5] uses a convolution approach to derive a result similar to Lemma 2.2, however our current simpler direct method uses integration techniques only. Interestingly, our direct method complements the convolution approach given by El-Taha [5].

The rest of the article is organized as follows. In Section 2 we focus on the standard $GI/M/c$ model and determine the transition probabilities efficiently. In Section 3 we give an intuitive method to extend our $GI/M/c$ results to the $GI^X/M/c$ and $GI^X/M/c/N$ models. For the finite buffer model, we cover both the partial and full batch rejection. In Section 4 we give several examples. The examples focus on the $GI/M/c$ model. Moreover, we give numerical results for large buffer size problems for $\rho < 1$ and when $\rho \geq 1$. In the Appendix we describe in detail the algorithm to compute the stationary transition probabilities and the stationary distribution function for the $GI/M/c$ and its variants.

2. The $GI/M/c$ Model

In this section we focus on the $GI/M/c$ model and observe the system at pre-arrival instants to determine the one-step transition probabilities. The $GI/M/c$ model can be described as follows: We have $i.i.d.$ inter-arrival times $A_i, i \geq 1$ and $i.i.d.$ exponential service times $B_i, i \geq 1$ with common distribution functions $A(t)$ and $B(t)$ respectively. The first moments of the inter-arrival and service times are given by $E(A) = 1/\lambda$ and $E(B) = 1/\mu$ respectively. Note that for this model $\mu_n = \min(n, c)\mu$ is the state dependent service rate that shall be needed later. The system state at pre-arrival epochs is described by the Markov chain $\{X_n, n = 0, 1, \cdots\}$ with transition probabilities given by several standard textbooks in queueing theory. Because these transition probabilities play an important role in our analysis and for ease of access we reproduce them in the lemma below (see for example Gross and Harris [11]).

Lemma 2.1. *(i) for $j \leq i+1 \leq c$, the transition probabilities of $p(i,j)$ are given by*

$$p(i,j) = \int_0^\infty \binom{i+1}{i-j+1} e^{-\mu t j}(1-e^{-\mu t})^{i-j+1} dA(t)\ ;$$

(ii) For $c \leq j \leq i+1$, the transition probabilities of $p(i,j)$ are given by

$$p(i,j) = \int_0^\infty \frac{e^{-c\mu t}(c\mu t)^{i-j+1}}{(i-j+1)!} dA(t);$$

(iii) for $j+1 \leq c \leq i$

$$p(i,j) = \binom{c}{c-j} \frac{(c\mu)^{i-c+1}}{(i-c)!} \int_0^\infty \int_0^t v^{i-c} e^{-\mu(t-v)j - c\mu v}(1-e^{-\mu(t-v)})^{c-j} dv\ dA(t)\ ;$$

and $p(i,j) = 0$, otherwise.

In the next subsection we provide our first key result. Specifically, we use direct integration to show that the one-step transition probability in Lemma 2.1 (iii) can be written as a finite sum. This result complements a convolution method used by El-Taha [5].

2.1. The One-step transition probabilities

We start by defining the Markov chain imbedded at the pre-arrival instants, then describing our method for computing the transition probabilities. Let X_n be the number of customers in the system at pre-arrival time instants, and D_n be the number of customers served during the n^{th} inter-arrival time. Then, X_{n+1} and X_n are related by

$$X_{n+1} = \begin{cases} X_n + 1 - D_n & D_n \leq X_n + 1, X_n \geq 0 \\ 0 & \text{otherwise} \end{cases} .$$

It is straightforward to see that $\{X_n, n \geq 1\}$ is a Markov chain with one-step transition probabilities defined as $p(i,j) = P\{X_n = j | X_{n-1} = i\}, i = 0, \ldots, ; j = 0, \ldots,$ and given by Lemma 2.1. Let the Laplace-Stieltjes transform (LST) of the inter-arrival times distribution function $A(t)$ be given by $A^*(s) = \int_0^\infty e^{-st} dA(t)$. Moreover, let $\frac{d^n A^*(s)}{ds^n}$ be the n^{th} derivative of $A^*(s)$ and denote $A_n^*(s) = (-1)^n \frac{d^n A^*(s)}{ds^n}$. It can be easily verified that $A_n^*(s) = \int_0^\infty t^n e^{-st} dA(t)$ for all $n \geq 0$, where $A_0^*(s) = A^*(s)$. Now we present our fundamental result.

Lemma 2.2. *For $j > 0, j + 1 \leq c \leq i$, the one-step transition probabilities are given by*

$$p(i,j) = \sum_{k=1}^{c-j} \frac{(-1)^{c-j-k}(c-1)!}{(k-1)!(c-j-k)!j!} \left(\frac{c}{k}\right)^{i-c+2} \left[A^*((c-k)\mu) - \sum_{r=0}^{i-c+1} \frac{(k\mu)^r A_r^*(c\mu)}{r!} \right]. \tag{1}$$

Remark. The difficulty in applying Lemma 2.1 is in evaluating the double integration in part (iii) which requires tedious numerical integration. We overcome this difficulty by converting this double integration problem into a finite sum as shown in Lemma 2.2. Furthermore, in Theorem 2.3 we express the transition probabilities in a more computationally suitable form by using the derivatives of the LST of the inter-arrival times distribution function.

Proof. To prove (1) we work with a modified version of Lemma 2.1 (iii). Define ν as the time required for $i - c + 2$ service completions, and $c - j - 1$ service completions for the remaining $t - \nu$ time. This leads to the equivalent equation

$$p(i,j) = \int_0^\infty \int_0^t \binom{c-1}{c-j-1} e^{-\mu(t-\nu)j} (1 - e^{-\mu(t-\nu)})^{c-j-1} \frac{(c\mu)^{i-c+2} \nu^{i-c+1} e^{-c\mu\nu}}{(i-c+1)!} \, d\nu \, dA(t)$$

$$= \binom{c-1}{c-j-1} \frac{(c\mu)^{i-c+2}}{(i-c+1)!} \int_0^\infty e^{-\mu jt} \int_0^t e^{-\mu(c-j)\nu} \nu^{i-c+1} (1 - e^{-\mu(t-\nu)})^{c-j-1} \, d\nu \, dA(t) \, .$$

Note that

$$(1 - e^{-\mu(t-\nu)})^{c-j-1} = \sum_{k=0}^{c-j-1} \binom{c-j-1}{k} (-1)^k e^{-k\mu(t-\nu)} \, ,$$

which leads to

$$p(i,j) =$$

$$\binom{c-1}{c-j-1}\frac{(c\mu)^{i-c+2}}{(i-c+1)!}\sum_{k=0}^{c-j-1}\binom{c-j-1}{k}(-1)^k\int_0^\infty e^{-\mu(j+k)t}\int_0^t \nu^{i-c+1}e^{-\mu(c-j-k)\nu}\,d\nu\,dA(t)\,. \tag{2}$$

Let $\mathcal{K} = \frac{(\mu(c-j-k))^{i-c+2}}{(i-c+1)!}$. Now, noting that $\mathcal{K}\nu^{i-c+1}e^{-\nu\mu(c-j-k)}d\nu$ is a gamma density function function, we obtain (e.g., Tijms[23], page 442)

$$\int_0^t \nu^{i-c+1}e^{-\nu\mu(c-j-k)}d\nu = \frac{(i-c+1)!}{(\mu(c-j-k))^{i-c+2}}\left[1-\sum_{r=0}^{i-c+1}\frac{(\mu(c-j-k))^r t^r e^{-t\mu(c-j-k)}}{r!}\right].$$

Substituting into (2), we have

$$\begin{aligned} p(i,j) &= \binom{c-1}{c-j-1}\frac{(c\mu)^{i-c+2}}{(i-c+1)!}\sum_{k=0}^{c-j-1}\binom{c-j-1}{k}(-1)^k\int_0^\infty e^{-\mu(j+k)t}\frac{(i-c+1)!}{(\mu(c-j-k))^{i-c+2}} \\ &\times\left[1-\sum_{r=0}^{i-c+1}\frac{(\mu(c-j-k))^r t^r e^{-t\mu(c-j-k)}}{r!}\right]dA(t) \\ &= \binom{c-1}{c-j-1}\frac{(c\mu)^{i-c+2}}{(i-c+1)!}\sum_{k=0}^{c-j-1}\binom{c-j-1}{k}(-1)^k\frac{(i-c+1)!}{(\mu(c-j-k))^{i-c+2}} \\ &\times\left[\int_0^\infty e^{-\mu(j+k)t}\,dA(t)-\sum_{r=0}^{i-c+1}\frac{(\mu(c-j-k))^r}{r!}\int_0^\infty t^r e^{-c\mu t}\,dA(t)\right]. \end{aligned}$$

Using,

$$A^*(s) = \int_0^\infty e^{-st}dA(t) \quad \text{and} \quad A_n^*(s) = \int_0^\infty t^n e^{-st}dA(t);$$

we have

$$\begin{aligned} p(i,j) &= \binom{c-1}{c-j-1}\frac{(c\mu)^{i-c+2}}{(i-c+1)!}\sum_{k=0}^{c-j-1}\binom{c-j-1}{k}(-1)^k\frac{(i-c+1)!}{(\mu(c-j-k))^{i-c+2}} \\ &\times\left[A^*(\mu(j+k))-\sum_{r=0}^{i-c+1}\frac{(\mu(c-j-k))^r}{r!}A_r^*(c\mu)\right] \\ &= \binom{c-1}{c-j-1}\sum_{k=0}^{c-j-1}\binom{c-j-1}{k}(-1)^k\left(\frac{c}{c-j-k}\right)^{i-c+2} \\ &\times\left[A^*(\mu(j+k))-\sum_{r=0}^{i-c+1}\frac{(\mu(c-j-k))^r}{r!}A_r^*(c\mu)\right]. \end{aligned}$$

By a change of variable where $c-j-k$ is replaced by k, we obtain

$$p(i,j) = \binom{c-1}{c-j-1} \sum_{k=1}^{c-j} \binom{c-j-1}{c-j-k} (-1)^{c-j-k} \left(\frac{c}{k}\right)^{i-c+2} \times \left[A^*(\mu(c-k)) - \sum_{r=0}^{i-c+1} \frac{(k\mu)^r}{r!} A_r^*(c\mu) \right].$$

Expanding the combinations and simplifying leads to

$$p(i,j) = \frac{(c-1)!}{(c-j-1)!(c-1-(c-j-1))!} \sum_{k=1}^{c-j} \frac{(c-j-1)!}{(c-j-k)!(c-j-1-(c-j-k))!} \times (-1)^{c-j-k} \left(\frac{c}{k}\right)^{i-c+2} \left[A^*(\mu(c-k)) - \sum_{r=0}^{i-c+1} \frac{(k\mu)^r}{r!} A_r^*(c\mu) \right].$$

Rearrange to obtain

$$p(i,j) = \sum_{k=1}^{c-j} \frac{(-1)^{c-j-k}(c-1)!}{(k-1)!(c-j-k)!j!} \left(\frac{c}{k}\right)^{i-c+2} \left[A^*((c-k)\mu) - \sum_{r=0}^{i-c+1} \frac{(k\mu)^r A_r^*(c\mu)}{r!} \right].$$

which is the desired result.

Using Lemma 2.2 and Lemma 2.1 we obtain the main computationally useful result.

Theorem 2.3. *(i) For $j \le i+1 \le c$, the transition probabilities $p(i,j)$ are given by*

$$p(i,j) = \binom{i+1}{i-j+1} \sum_{r=0}^{i-j+1} (-1)^r \binom{i-j+1}{r} A^*((j+r)\mu)) .$$

(ii) For $c \le j \le i+1$, the transition probabilities $p(i,j)$ are given by

$$p(i,j) = \frac{(c\mu)^{i-j+1} A_{i-j+1}^*(c\mu)}{(i-j+1)!} .$$

(iii) for $j > 0, j+1 \le c \le i$ the transition probabilities $p(i,j)$ are given by

$$p(i,j) = \sum_{k=1}^{c-j} \frac{(-1)^{c-j-k}(c-k)C_{k,c-j}^a}{j} \left(\frac{c}{k}\right)^{i-c+2} \left[A^*((c-k)\mu) - \sum_{r=0}^{i-c+1} \frac{(k\mu)^r A_r^*(c\mu)}{r!} \right] ; \tag{3}$$

where $C_{k,c-j}^a = \prod_{m=1}^{k-1} \frac{c-m}{k-m} \times \prod_{m=k+1}^{c-j} \frac{c-m}{m-k}$, $\prod$ over empty sets is 1; and $p(i,j) = 0$, otherwise.

Note that $C^a_{k,c-j}$ is always positive, and can be written as

$$C^a_{k,c-j} = \frac{(c-1)!(c-k-1)!}{(c-k)!(k-1)!(j-1)!(c-j-k)!} \, . \tag{4}$$

Similar results are given by El-Taha [5] using a convolution approach. Our approach here is more direct and intuitive. We note that Lemma 2.2 allows for direct evaluation of $p(i,j)$ for $j=0$ in region 3, while Theorem 2.3(iii) does not. However we can always compute $p(i,0)$ as the complement of the remaining $p(i,j)$ values for $j \geq 1$.

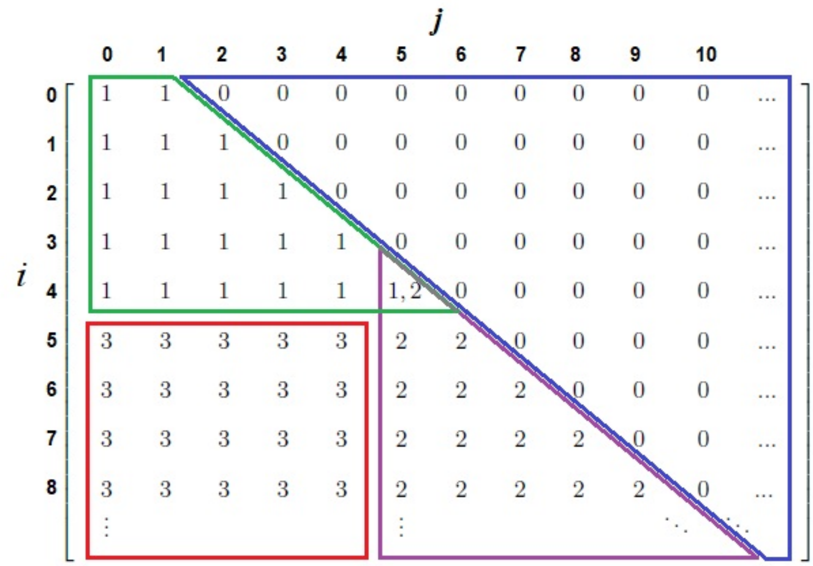

Figure 1. Transition Matrix Regions for $c=5$

Referring to Figure 1, region 1 is where transition probabilities $p(i,j)$ are given by Lemma 2.1 (i); region 2 is where $p(i,j)$ are similar to the $GI/M/1$ model with service rate $c\mu$; region 3 is where $p(i,j)$ are given by the direct method; and region 4 is where $p(i,j)$ are $0's$.

Computing the one-step transition probabilities $p(i,j)$ requires the evaluation of the derivatives of the LST of the inter-arrival time distribution functions. This can be easily done exactly for all Phase Type (PH-Type) distribution functions. Because PH-Type distributions are dense in the set of all continuous distribution functions with support on $[0,\infty)$, our approach works at least approximately for any continuous non-negative distribution function. Now, we introduce PH-Type distribution functions and give their properties, see Neuts [18] Ch. 2, including a formula for the derivatives of their LST.

PH-Type distribution functions are modeled as the time until absorption in a Markov process (MP) with a single absorption state. Consider a PH-Type distribution function defined on a MP with states $\{1,\cdots,k+1\}$, $k+1$ being the absorption state, and a transition rate matrix

$$Q = \begin{pmatrix} T & T^0 \\ \mathbf{0} & 0 \end{pmatrix},$$

where T is a $k \times k$ matrix with $T_{ij} \geq 0, i \neq j$, and $T_{ii} < 0, i = 1, \ldots, k$; $\mathbf{1}$ is a vector of ones; T^0 is chosen such that $T\mathbf{1}+T^0 = \mathbf{0}$ i.e., the row of elements of Q add up to 0; and $(\boldsymbol{\alpha}, 0)$ with $\boldsymbol{\alpha}\mathbf{1} = 1$, is the initial distribution of the MP. Let the random variable X be the time until absorption, then the distribution function of X is said to be a PH-Type distribution function with representation $(\boldsymbol{\alpha}, T)$. Now let the distribution function of X be the inter-arrival time distribution function of the $GI/M/c$ model, then the distribution function and its LST are given by

$$\begin{aligned} A(t) &= 1 - \boldsymbol{\alpha} \exp(Tt)\mathbf{1}, t \geq 0; \\ A^*(s) &= \boldsymbol{\alpha}(sI - T)^{-1}T^0 \ . \end{aligned}$$

Moreover,

$$A_n^*(s) = n!\boldsymbol{\alpha}(sI - T)^{-(n+1)}T^0 \ . \tag{5}$$

This is explored further in Section 4 where (5) is used to give closed form expressions for the derivatives of several cases of PH-Type distribution functions including the exponential, the Erlang, and the hyper-exponential.

Remarks on Complexity and Stability. We note that the method given by Theorem 2.3 (iii) involves two finite sums and computations of the order of $O((i-c)(c-j))$. In contrast the method using Lemma 2.1 (iii) would involve numerically approximating a double infinite integration. El-Taha [5] also discusses numerical stability issues that arise when calculations involve alternating between positive and negative terms. Specifically, the idea is to collect all positive (negative) terms together and use one subtraction at the end as we discuss in Subsection 4.3.

2.2. Arrival-time and time-average probability distributions.

The contribution of Lemma 2.2 and Theorem 2.3 is in computing the transition probabilities efficiently using a finite sum as compared to approximations and numerical integration that would be needed in part (iii) of Theorem 2.3. Now one can use one of several methods in the literature (see references) to compute the system size stationary distribution function. We briefly discuss one efficient method to compute the arrival-time probabilities. Then we relate the time-average probabilities to the pre-arrival ones.

Arrival-Time Probabilities.

Let $\{\pi(i), i = 0, \ldots, \}$ be the stationary distribution function of the imbedded Markov chain $\{X_n, n = 0, 1 \ldots\}$, then solve the system of equations $\pi = \pi P, \sum \pi(i) = 1$ to obtain the arrival time probabilities. A more efficient approach is to use the concept that probability flow across cuts balance to efficiently solve the pre-arrival time probabilities for the $GI/M/c/N$ model using Lemma 1.4 of Kelly [13]. For $j = 1, 2, \cdots, N$

$$\pi(j)p(j, j+1) = \sum_{k=j+1}^{N} \pi(k) \sum_{i=0}^{j} p(k, i) \ ,$$

which implies

$$\pi(j) = \sum_{k=j+1}^{N} \pi(k)a(k,j)/p(j,j+1) \ , \tag{6}$$

where $a(k,j) = \sum_{i=0}^{j} p(k,i)$.

One can compute the $\{\pi(.)\}$ recursively using (6) starting with $\pi(N)$. For the infinite capacity model we can iterate on N until we achieve a desired level of accuracy. Instead, we pre-determine N so that a prescribed level of precision is achieved. It is well-known (Gross and Harris [11]) that $\pi(n) = C\sigma^n$ for $n \geq c$ where C is a constant and $|\sigma| < 1$ is the solution of the equation

$$\sigma = \int_0^{\infty} e^{-c\mu(1-\sigma)} t dA(t) \equiv A^*(c\mu(1-\sigma)) \ . \tag{7}$$

Moreover, C is given by Gross and Harris [11] as $C = \frac{1-\sum_{j=0}^{c-1}\pi(j)}{\sigma^c(1-\sigma)^{-1}}$. The following lemma serves to guide our selection of N.

Lemma 2.4. *For all n such that*

$$n > c + \frac{\ln(\epsilon) - 2\ln(1-\sigma)}{\ln(\sigma)} \ ;$$

we have $|\pi(n+1) - \pi(n)| < \epsilon$, where ϵ is the tolerance level.

Time-Average Stationary Distribution.

One is typically interested in the system performance measures like the mean number of customers in the system and the queue L and L_q, and the mean delay in the system and the queue W and W_q respectively. To do this we need the time-average probability distribution function which is of interest in its own right. Let $\{X(t), t \geq 0\}$ be a stochastic process so that $X(t)$ is the number of customers in the system at time t, then the time-average stationary distribution is defined as

$$p(n) = \lim_{t \to \infty} P\{X(t) = n\}, n = 0, 1, \ldots, .$$

We evaluate the stationary distribution function $\{p(.)\}$ by relating those probabilities to the pre-arrival probabilities $\pi(.)$ as shown in the following result.

Lemma 2.5. *Consider the $GI/M/c/N$ queueing model and let $\rho = \lambda/c\mu$. Then the system size probabilities, $p(i), i = 0, \ldots N$, are given by*

$$p(n) = \begin{cases} (1-\rho) + \rho\pi(N) - \rho\sum_{k=0}^{c-2} \frac{c-k-1}{k+1}\pi(k), & n = 0 \ ; \\ c\rho\pi(n-1)/n \ , & 1 \leq n \leq c \ ; \\ \rho\pi(n-1) \ , & c < n \leq N \ . \end{cases}$$

A proof of Lemma 2.5 is given by El-Taha [5]. For more details on the computation of stationary probabilities the reader is referred to El-Taha and Michaud [6].

3. Extension to $GI^X/M/c/N$ Model

In this section we discuss an intuitive method to extend our $GI/M/c$ results to the $GI^X/M/c$ and $GI^X/M/c/N$ models. We note that the models have been discussed by several articles, notably by Laxmi and Gupta [16] and Chaudry and Kim [2] among others mentioned in the introduction. Our approach is different in the sense that we generate the one-step transition probabilities of the batch arrival models directly from the standard $GI/M/c$ model and the batch size distribution function.

The section builds on Theorem 2.3 by devising a remarkably intuitive and new approach to express the transition probabilities for the batch arrival $GI/M/c$ models in terms of the transition probabilities of the $GI/M/c$ model given by Theorem 2.3. These results are expressed in Lemmas 3.1, 3.3, and 3.5. Also the results in Lemmas 3.2, 3.4, and 3.6, that follow from more general results in El-Taha and Stidham [7] allow us to compute time-average from arrival-time probabilities.

Let X be a random variable that represents the batch size, and let $b(k)$ be the pmf of X with support $[1, \infty)$ such that $b(k) = P(X = k)$. We note here that typical random variables that are used to model batch sizes have a support that starts at 0. Let $g(k)$ be a $p.m.f.$ with support $[0, \infty)$, that is, with g(k) we allow a batch of size 0 to occur like the Poisson and the geometric random variables. In this situation, we think of $b(k)$ as $b(k) = g(k)/(1 - g(0))$. Also note that in these models $A(t)$ is the distribution function of batch inter-arrival times and $1/\lambda$ is the mean time between batch arrivals, so that the overall arrival rate $\lambda_A = \lambda E[X]$. Also $\rho = \lambda E[X]/c\mu$. We require $\rho < 1$ for the infinite buffer models.

Moreover let $p^*(i, j)$ be the one-step transition probabilities just before arrival of the imbedded Markov chain of the $GI^X/M/c$ and $GI^X/M/c/N$ models. Recall that $p(i, j)$ are the corresponding one-step transition probabilities associated with the $GI/M/c$ and $GI/M/c/N$ models. Now we consider three cases.

Case 1. $GI^X/M/c$

We consider the $GI/M/c$ with batch arrivals and obtain the one-step transition probabilities.

Lemma 3.1. *Consider the $GI^X/M/c$ queueing model. Then, for all i, j*

$$p^*(i, j) = \sum_{k=1}^{\infty} p(i + k - 1, j) b(k) \ .$$

Let $\{\pi(i)\}$ and $\{p(i)\}$ be the system size pre-arrival and time-average stationary probability distribution functions respectively. Now replace $p(i, j)$ with $p^*(i, j)$ to compute $\{\pi(i)\}$ of the resulting Markov chain. In order to relate the time-average to pre-arrival probabilities we use the following result.

Lemma 3.2. *Consider the $GI^X/M/c$ queueing model with batch arrivals. Then the system size probabilities, $\{p(i), i = 0, \ldots\}$, are given by*

$$p(n) = \begin{cases} \lambda \sum_{k=0}^{n-1} \pi(k) B^c(n-k) / \min(n,c)\mu, & n = 1, \cdots ; \\ 1 - \sum_{k=1}^{\infty} p(k) , & n = 0 ; \end{cases}$$

where $B^c(n-k) = \sum_{n-k}^{\infty} b(i)$.

Proof. The proof follows from El-Taha and Stidham [7], equation (4.34), page 107 ; see also Stidham and El-Taha [20].

Case 2: $GI^X/M/c/N$ with partial rejection.

In this case a batch that brings the system state above N is partially accepted in the sense that the system will accept a part of the batch for the state to reach N and the rest of the batch is rejected. Then for all i, j the one-step transition probabilities are given by this result.

Lemma 3.3. *Consider the $GI^X/M/c/N$ queueing model with partial rejection. Then, for all i, j*

$$p^*(i,j) = \sum_{k=1}^{N-i-1} p(i+k-1, j) b(k) + p(N-1, j) B^c(N-i) ;$$

where $B^c(N-i) = \sum_{k=N-i}^{\infty} b(k)$.

Proof. It is clear that

$$\begin{aligned} p^*(i,j) &= \sum_{k=1}^{N-i-1} p(i+k-1, j) b(k) + p(N-1, j) \left(\sum_{k=N-i}^{\infty} b(k) \right) \\ &= \sum_{k=1}^{N-i-1} p(i+k-1, j) b(k) + p(N-1, j) B^c(N-i) . \end{aligned}$$

In order to relate the time-average to pre-arrival probabilities we use the following result.

Lemma 3.4. *Consider the $GI^X/M/c/N$ queueing model with partial rejection. Then the system size probabilities, $\{p(i), i = 0, \ldots N\}$, are given by*

$$p(n) = \begin{cases} \lambda \sum_{k=0}^{n-1} \pi(k) B^c(n-k) / \min(n,c)\mu, & n = 1, \cdots N ; \\ 1 - \sum_{k=1}^{N} p(k) , & n = 0 ; \end{cases}$$

where $B^c(n-k) = \sum_{n-k}^{\infty} b(i)$.

Proof. The proof is similar to Lemma 3.2.

Case 3: $GI^X/M/c/N$ with full rejection.

In this case a batch that brings the system state above N is fully rejected.

Lemma 3.5. *Consider the $GI^X/M/c/N$ with full rejection. Then, for all i, j*

$$p^*(i,j) = \sum_{k=1}^{N-i} p(i+k-1,j)b(k) + p(i,j)B^c(N-i+1) \; ;$$

where $B^c(N-i+1) = \sum_{k=N-i+1}^{\infty} b(k)$.

Proof. It is clear that

$$\begin{aligned} p^*(i,j) &= \sum_{k=1}^{N-i} p(i+k-1,j)b(k) + p(i,j)\left(\sum_{k=N-i+1}^{\infty} b(k)\right) \\ &= \sum_{k=1}^{N-i} p(i+k-1,j)b(k) + p(i,j)B^c(N-i+1) \; . \end{aligned}$$

We use the following results to relate the time-average probabilities to pre-arrival probabilities.

Lemma 3.6. *Consider the $GI^X/M/c/N$ with full rejection. Then the system size probabilities, $\{p(i), i = 0, \ldots N\}$, are given by*

$$p(n) = \begin{cases} \lambda \sum_{k=1}^{n-1} \pi(k)[B^c(n-k) - B^c(N-k)]/\min(n,c)\mu, & n = 1, \cdots N \; ; \\ \\ 1 - \sum_{k=1}^{N} p(k) \; , & n = 0 \; ; \end{cases}$$

where $B^c(n-k) = \sum_{n-k}^{\infty} b(i)$.

Proof. The proof is similar to Lemma 3.4, except here batches that result in more than N customers in the system are totally rejected.

Yao et.al.[24] and Laximi and Gupta [16] give similar relations to relate time-average and pre-arrival probabilities as in Lemma 3.4 and Lemma 3.6. Replacing $p(i,j)$ with $p^*(i,j)$, the arrival-time probabilities can be computed using $\pi = \pi P^*$, $\sum_{i \in S} \pi(i) = 1$ where $P^* = [p^*(i,j)]$ is the one-step transition matrix and S is the state space. One can also use one of the methods discussed in the literature in the introduction to compute the arrival-time distribution. The time-average probabilities are then computed using the corresponding Lemma 3.2, Lemma 3.4, or Lemma 3.6. In Appendix A, the computations for the batch arrival models are given by an appropriate adaptation of the algorithm for the $GI/M/c/N$ model.

4. Examples and Numerical Results

In this section we give the transition probabilities of small finite buffer examples of the $GI/M/c/N$ model, and provide numerical results that compare our method to other approaches in the literature. We also discuss numerical results for large scale problems. We note that in the infinite buffer case we truncate the system size by using $|\pi(n+1) - \pi(n)| < \epsilon$. Moreover, we include numerical results for the finite buffer model when $\rho \geq 1$.

4.1. Special examples

Our focus here is on inter-arrival time distributions whose LST have closed form multiple derivatives. We choose four distribution functions with coefficients of variation that vary from 0 to infinity. Specifically, we select the deterministic, Erlang, exponential, and the hyper-exponential distribution functions. Let $a(t)$ be the $p.d.f.$ of the inter-arrival times. Recall that we require that the inter-arrival time distribution function has a mean $E[A] = 1/\lambda$. Here we give explicit forms for the derivatives of the LST of the inter-arrival time distribution functions.

Deterministic. In this case we assume that $a(t) = a$ w.p. 1, so that $A_n^*(s) = a^n e^{-sa}$. Note that here $\lambda = 1/a$.

Exponential. Here $a(t) = \lambda e^{-\lambda t}, t \geq 0$, so that $A_n^*(s) = n!\lambda/(s+\lambda)^{n+1}$.

Erlang. The density function for a k phase Erlang is $a(t) = \frac{\theta(\theta t)^{k-1}}{(k-1)!}e^{-\theta t}, \theta > 0, t \geq 0$. Note that here $E[A] = k/\theta$, so that $\theta = k\lambda$. We would like the mean to stay constant, so we replace θ by $k\lambda$, and use density function for a k phase Erlang as $a(t) = \frac{k\lambda(k\lambda t)^{k-1}}{(k-1)!}e^{-k\lambda t}, \lambda > 0, t \geq 0$. Here $A_n^*(s) = n!\binom{k+n-1}{k-1}(k\lambda)^k/(s+k\lambda)^{k+n}$. Note that again $E[A] = 1/\lambda$.

Hyper-exponential. Let $a_i(t)$ be an exponential $p.d.f$ with parameter λ_i. Then, the k phase hyper-exponential is given by $a(t) = \sum_{i=1}^{k} p_i a_i(t)$; so that $A^*(s) = \sum_{i=1}^{k} \frac{p_i\lambda_i}{s+\lambda_i}$. We shall use the two phase hyper-exponential which is a mixture of two exponential distribution functions. The density function can be written as $a(t) = p\lambda_1 e^{-\lambda_1 t} + (1-p)\lambda_2 e^{-\lambda_2 t}, \lambda_1 > 0, \lambda_2 > 0, t \geq 0, (0 \leq p \leq 1)$. Therefore

$$A_n^*(s) = n!\left[\frac{p\lambda_1}{(s+\lambda_1)^{n+1}} + \frac{(1-p)\lambda_2}{(s+\lambda_2)^{n+1}}\right] .$$

Now, we show how to use these transforms to write the one-step transition matrix in explicit form. We present small-scale, finite-buffer examples for a system with $c = 3$ servers and a total capacity of $K = 6$. Now referring to the regions described in Figure 1 and using Theorem 2.3, we write the one-step transition probabilities in the more computationally convenient form as follows.

(i) For Region 1, where $i \leq c-1$ and $j \leq i+1$, (i.e.: $i = 0, 1, 2$ and $j = 1, ..., i+1$), we use

$$p(i,j) = \frac{(i+1)!}{j!}\sum_{k=0}^{i-j+1}\frac{(-1)^k A^*((j+k)\mu)}{(i-j-k+1)!k!} .$$

(ii) For Region 2, where $i = c, c+1, ..., K-1, \quad j = c, c+1, ..., i+1, \quad i+1 \leq K$ (i.e.: $i = 3, 4, 5$ and $j = 3, 4, 5$), we use

$$p(i,j) = \frac{(c\mu)^{i-j+1}}{(i-j+1)!}A_{i-j+1}^*(c\mu) = \frac{(3\mu)^{i-j+1}A_{i-j+1}^*(3\mu)}{(i-j+1)!} .$$

(iii) For Region 3, where $1 \leq j \leq c-1 < i$ (i.e.: $i = 3, 4, 5$ and $j = 1, 2$), we use the direct result

$$p(i,j) = \sum_{k=1}^{3-j} \frac{C_{k,3-j}(3-k)}{j} \left(\frac{3}{k}\right)^{i-1} \left[A^*((3-k)\mu) - \sum_{r=0}^{i-2} \frac{(k\mu)^r A_r^*(3\mu)}{r!}\right] ;$$

where

$$C_{k,3-j} = \prod_{m=1}^{k-1} \frac{3-m}{k-m} \prod_{m=k+1}^{3-j} \frac{3-m}{k-m} \quad , \; C_{3-j,3-j} = \prod_{m=1}^{2-j} \frac{3-m}{3-j-m} \, , \quad \text{and}$$

$$C_{v,v} = \prod_{m=1}^{v-1} \frac{3-m}{v-m} \; ; \text{ so that } C_{1,1} = 1, C_{1,2} = -1, C_{2,2} = 2 \, .$$

Also, for $i \geq c$ and $j = 0$, we simply use $p(i,j) = 1 - \sum_{n=1}^{K} p(i,n)$. Moreover, for Region 4, $p(i,j) = 0$ where $j > i+1$. When $i = K$, use $p(K,j) = p(K-1,j)$ for all $j = 0, \cdots, K$. Note that, if i is the number in the system immediately prior to an arrival, then the transition probabilities when $i = K-1$ must be the same as the probabilities when $i = K$, because in the first case, the system becomes full, and in the second case, the system is already full and the new arrival is lost. Given the Markovian property, the transition probabilities are unaffected by additional arrivals while the system is full. Therefore the one-step transition matrix is given by

$$P = \begin{bmatrix}
1 - A^*(\mu) & A^*(\mu) & 0 & 0 & 0 & 0 & 0 \\
1 - 2A^*(\mu) + A^*(2\mu) & 2A^*(\mu) - 2A^*(2\mu) & A^*(2\mu) & 0 & 0 & 0 & 0 \\
\begin{array}{l} 1 - 3A^*(\mu) + 3A^*(2\mu) \\ \; -A^*(3\mu) \end{array} & \begin{array}{l} 3A^*(\mu) - 6A^*(2\mu) \\ \; +3A^*(3\mu) \end{array} & 3A^*(2\mu) - 3A^*(3\mu) & A^*(3\mu) & 0 & 0 & 0 \\
1 - \sum_{n=1}^{4} p(3,n) & p(3,1) & p(3,2) & 3\mu A_1^*(3\mu) & A^*(3\mu) & 0 & 0 \\
1 - \sum_{n=1}^{5} p(4,n) & p(4,1) & p(4,2) & \frac{9}{2}\mu^2 A_2^*(3\mu) & 3\mu A_1^*(3\mu) & A^*(3\mu) & 0 \\
1 - \sum_{n=1}^{6} p(5,n) & p(5,1) & p(5,2) & \frac{9}{2}\mu^3 A_3^*(3\mu) & \frac{9}{2}\mu^2 A_2^*(3\mu) & 3\mu A_1^*(3\mu) & A^*(3\mu) \\
1 - \sum_{n=1}^{6} p(6,n) & p(6,1) & p(6,2) & \frac{9}{2}\mu^3 A_3^*(3\mu) & \frac{9}{2}\mu^2 A_2^*(3\mu) & 3\mu A_1^*(3\mu) & A^*(3\mu)
\end{bmatrix}$$

where

$$p(3,1) = \frac{9}{2}A^*(\mu) - 18A^*(2\mu) + \frac{27}{2}A^*(3\mu) + 9\mu A_1^*(3\mu) \; ;$$
$$p(3,2) = 9A^*(2\mu) - 9A^*(3\mu) - 9\mu A_1^*(3\mu) \; ;$$
$$p(4,1) = \frac{27}{4}A^*(\mu) - 54A^*(2\mu) + \frac{189}{4}A^*(3\mu) + \frac{81}{2}\mu A_1^*(3\mu) + \frac{27}{2}\mu^2 A_2^*(3\mu) \; ;$$
$$p(4,2) = 27A^*(2\mu) - 27A^*(3\mu) - 27\mu A_1^*(3\mu) - \frac{27}{2}\mu^2 A_2^*(3\mu) \; ;$$

$$
\begin{aligned}
p(5,1) &= \frac{81}{8}A^*(\mu) - 162A^*(2\mu) + \frac{1215}{8}A^*(3\mu) + \frac{567}{4}\mu A_1^*(3\mu) + \frac{243}{4}\mu^2 A_2^*(3\mu) \\
&\quad + \frac{27}{2}\mu^3 A_3^*(3\mu) \; ; \\
p(5,2) &= 81A^*(2\mu) - 81A^*(3\mu) - 81\mu A_1^*(3\mu) - \frac{81}{2}\mu^2 A_2^*(3\mu) - \frac{27}{2}\mu^3 A_3^*(3\mu) \; ; \\
p(6,1) &= p(5,1) \text{ ; and } p(6,2) = p(5,2) \; .
\end{aligned}
$$

Deterministic Arrivals. For deterministic inter-arrivals with probability density function $a(t) = a$, and 0 otherwise (i.e. $\lambda^{-1} = a$), thus we have

$$A_n^*(s) = a^n e^{-sa} \; ;$$

therefore

$$A^*(\mu(j+m)) = e^{-\mu(j+m)a} \; ;$$

and

$$A_n^*(c\mu) = a^n e^{-c\mu a} \; .$$

This gives the following: $A^*(0) = 1$, $A^*(\mu) = e^{-a\mu}$, $A_1^*(3\mu) = ae^{-3a\mu}$, $A^*(2\mu) = e^{-2a\mu}$, $A_2^*(3\mu) = a^2e^{-3a\mu}$, $A^*(3\mu) = e^{-3a\mu}$, and $A_3^*(3\mu) = a^3e^{-3a\mu}$. Substitute in the general arrivals matrix to get the corresponding one-step transition matrix for deterministic inter-arrival times. Thus our transition matrix is as follows:

$$
P = \begin{bmatrix}
1 - e^{-a\mu} & e^{-a\mu} & 0 & 0 & 0 & 0 & 0 \\
1 - 2e^{-a\mu} + e^{-2a\mu} & 2e^{-a\mu} - 2e^{-2a\mu} & e^{-2a\mu} & 0 & 0 & 0 & 0 \\
\begin{matrix} 1 - 3e^{-a\mu} + 3e^{-2a\mu} \\ - e^{-3a\mu} \end{matrix} & \begin{matrix} 3e^{-a\mu} - 6e^{-2a\mu} \\ + 3e^{-3a\mu} \end{matrix} & 3e^{-2a\mu} - 3e^{-3a\mu} & e^{-3a\mu} & 0 & 0 & 0 \\
1 - \sum_{n=1}^{4} p(3,n) & p(3,1) & p(3,2) & 3a\mu e^{-3a\mu} & e^{-3a\mu} & 0 & 0 \\
1 - \sum_{n=1}^{5} p(4,n) & p(4,1) & p(4,2) & \frac{9}{2}a^2\mu^2 e^{-3a\mu} & 3a\mu e^{-3a\mu} & e^{-3a\mu} & 0 \\
1 - \sum_{n=1}^{6} p(5,n) & p(5,1) & p(5,2) & \frac{9}{2}a^3\mu^3 e^{-3a\mu} & \frac{9}{2}a^2\mu^2 e^{-3a\mu} & 3a\mu e^{-3a\mu} & e^{-3a\mu} \\
1 - \sum_{n=1}^{6} p(6,n) & p(6,1) & p(6,2) & \frac{9}{2}a^3\mu^3 e^{-3a\mu} & \frac{9}{2}a^2\mu^2 e^{-3a\mu} & 3a\mu e^{-3a\mu} & e^{-3a\mu}
\end{bmatrix}
$$

where

$$
\begin{aligned}
p(3,1) &= \frac{9}{2}e^{-a\mu} - 18e^{-2a\mu} + \frac{27}{2}e^{-3a\mu} + 9\mu a e^{-3a\mu} \; ; \\
p(3,2) &= 9e^{-2a\mu} - 9e^{-3a\mu} - 9\mu a e^{-3a\mu} \; ; \\
p(4,1) &= \frac{27}{4}e^{-a\mu} - 54e^{-2a\mu} + \frac{189}{4}e^{-3a\mu} + \frac{81}{2}\mu a e^{-3a\mu} + \frac{27}{2}\mu^2 a^2 e^{-3a\mu} \; ; \\
p(4,2) &= 27e^{-2a\mu} - 27e^{-3a\mu} - 27\mu a e^{-3a\mu} - \frac{27}{2}\mu^2 a^2 e^{-3a\mu} \; ; \\
p(5,1) &= \frac{81}{8}e^{-a\mu} - 162e^{-2a\mu} + \frac{1215}{8}e^{-3a\mu} + \frac{567}{4}\mu a e^{-3a\mu} + \frac{243}{4}\mu^2 a^2 e^{-3a\mu}
\end{aligned}
$$

$$+ \frac{27}{2}\mu^3 a^3 e^{-3a\mu} \; ;$$

$$p(5,2) = 81e^{-2a\mu} - 81e^{-3a\mu} - 81\mu a e^{-3a\mu} - \frac{81}{2}\mu^2 a^2 e^{-3a\mu} - \frac{27}{2}\mu^3 a^3 e^{-3a\mu} \; ;$$

$$p(6,1) = p(5,1), \text{ and } \; p(6,2) = p(5,2) \; .$$

Similar to the deterministic distribution function, one can easily generate one-step transition matrices for the other distributions, namely the exponential, the Erlang, and the hyper-exponential.

4.2. Numerical results for small buffer size

Here we provide numerical results for the distributions given above, for a system with $c = 3$ servers, arrival rate $\lambda = 5$, service rate $\mu = 2$, capacity of $N = 6$, and utilization factor $\rho = .83\overline{3}$. For the hyper-exponential we use $p = .8, \lambda_1 = 8$, and $\lambda_2 = 2$, so that the overall $\lambda = 5$. Numerical results are reported in Tables 1-3. In Table 1, we use the exponential distribution function and compare our method with the traditional method, i.e. the $M/M/3/6$ model. The results match perfectly. This helps to verify our approach numerically. Table 2 reports the time-average probabilities for the other three distribution functions, and Table 3 reports performance measures.

Table 1. Numerical Results: Finite Buffer Model With Exponential Arrivals

p(n)	Exponential-Direct	Exponential-Traditional
0	0.067958810	0.067958810
1	0.169897026	0.169897026
2	0.212371283	0.212371283
3	0.176976069	0.176976069
4	0.147480057	0.147480057
5	0.122900048	0.122900048
6	0.102416707	0.102416707

Table 2. Numerical Results: inite Buffer Model With Deterministic, Erlang, or Hyper-Exponential Arrivals

p(n)	Deterministic	Erlang	Hyper-Exponential
0	0.047234853	0.060394802	0.095667547
1	0.127764093	0.153533580	0.170777140
2	0.254669333	0.228194616	0.182173364
3	0.232414603	0.196990341	0.153721590
4	0.156638062	0.150739248	0.148862427
5	0.107500964	0.117912665	0.131909935
6	0.073778091	0.092234748	0.116887997

Table 3. Performance Measures, Finite Buffer Models

	L	W
Exponential - Direct	2.944488506	0.656092538
Exponential - Traditional	2.944488506	0.656092538
Deterministic - Direct	2.941072182	0.635068584
Erlang - Direct	2.946822639	0.649247728
Hyper-exponential - Direct	2.952616004	0.668684379

4.3. Numerical results for large buffer size

In this subsection we deal with large scale applications and report on computational methodology and numerical results. We cover the $GI/M/c$ model with heavy traffic $\rho < 1$. This will result in large buffer sizes determined by Lemma 2.4. The approach for $GI/M/c/N$ with $\rho < 1$ work similarly except that N is predetermined. We also give numerical results for the $GI/M/c/N$ when $\rho \geq 1$.

GI/M/c with $\rho < 1$.

For the four distributions, deterministic, Erlang, exponential, and hyper-exponential, we recursively compute the stationary distributions $\{\pi(.)\}$ and $\{p(.)\}$ using the direct method as in Theorem 2.3 and the algorithm in the Appendix. For the large scale examples, reported in Table 4 and Figures 2 and 3, we use $\lambda = 5.8$, $\mu = .2$, $c = 30$, and $\rho = .96\overline{6}$. In the case of the hyper-exponential we use $p = .873563218, \lambda_1 = 8$, and $\lambda_2 = 2$ which gives a coefficient of variation ≈ 1.430035 . For each of the four distribution functions, we use $\epsilon = 10^{-125}$ in step 3 of the algorithm to recursively solve for σ, and use Lemma 2.4 (step 4 of the algorithm) with $\epsilon = 10^{-16}$ to identify the truncation value for the infinite capacity. The values for N are 490, 705, 918 and 1349 for the deterministic, Erlang, exponential, and hyper-exponential respectively.

For large buffer size N, numerical stability becomes an issue that needs to be dealt with. The term $(-1)^j$ in Theorem 2.3 (i) and (iii) which causes subtraction in every other step is one source of numerical instability. To address this issue consider a sum of the form $S(.) = \sum_{k=1}^{J}(-1)^{J-k}h_k(.)$, and rewrite as

$$S(.) = |\sum_{u=1}^{[J/2]} h_{2u}(.) - \sum_{u=1}^{[(J+1)/2]} h_{2u-1}(.)| \tag{8}$$

Note that how this form converts $J/2$ subtractions into one subtraction at the end. We use (8) to rewrite (3) as

$$p(i,j) = \left| \sum_{u=1}^{[(c-j+1)/2]} \frac{C^a_{2u-1,c-j}(c-2u+1)}{j} \times \left(\frac{c}{2u-1}\right)^{i-c+2} \Big[A^*((c-2u+1)\mu) - \sum_{r=0}^{i-c+1} \frac{((2u-1)\mu)^r A^*_r(c\mu)}{r!}\Big] - \sum_{u=1}^{[(c-j)/2]} \frac{C^a_{2u,c-j}(c-2u)}{j} \left(\frac{c}{2u}\right)^{i-c+2}\right.$$

$$\times \quad \left[A^*((c-2u)\mu) - \sum_{r=0}^{i-c+1} \frac{(2u\mu)^r A_r^*(c\mu)}{r!}\right]\Bigg| \tag{9}$$

Additionally, part (i) of Theorem 2.3 can be written in a similar stable form.

We use the Python programming language to compute the stationary probabilities for large-scale examples. Use of the Decimal package, a fixed-decimal package capable of arbitrarily long mantissas, is notable in addressing the overflow errors associated with large factorials and the underflow issues created by the LST values with large c and N.

The results for exponential inter-arrivals are compared with traditional methods for computing stationary probabilities using $M/M/c$ queues such as described in Gross & Harris[11] and Kleinrock[15]. This serves to verify, numerically, our method computations. For deterministic, Erlang, and hyper-exponential inter-arrivals, our results are compared with Takács method as generated by the QTS software provided by Gross & Harris [11].

As can be seen from the QTS (Takács) results for the Erlang, deterministic, and hyper-exponential distributions, difficulties with floating-point overflow/underflow exist with this number of servers, and can persist as low as $c = 10$. These problems expand with increasing c, limiting usable results from that software.

With the exception of $p(0)$, which compounds the error present in all other values of $p(n)$, our method is more numerically stable than Takács as implemented by QTS software provided by Gross & Harris [11] even when using floating-point levels of precision. Our use of the Decimal package provides accuracy for substantially higher values of c while also reducing the error of $p(0)$ below that of floating-point implementations.

Table 4 provides performance measures comparing our direct algorithm and Takács method. There is a good match between the two methods. Moreover, Figure 2 and Figure 3 provide the cumulative and density distribution functions for the four distributions, however we truncated the distribution functions in Figures 2 and 3 at 100 and 80 respectively.

Table 4. Performance Measures when $\rho < 1$

Distribution/Method	L	W
Deterministic - Direct	39.357	6.786
Deterministic - Takács	39.357	6.786
Erlang - Direct	45.600	7.870
Erlang - Takács	45.637	7.869
Exponential - Direct	52.083	8.980
Exponential - Traditional	52.083	8.980
Hyper-exponential - Direct	65.500	11.300
Hyper-exponential - Takács	65.480	11.290

Our approach in dealing with the batch arrival model is to build on the corresponding standard $G/M/c$ model as indicated in the adaptation algorithm given in the appendix.

GI/M/c/N with $\rho \geq 1$.

In this case we need to make a few adjustments to the algorithm in the appendix. We do not need to root solve for σ but need to pre-specify N. Also, we initialize $\pi(N) = 1$

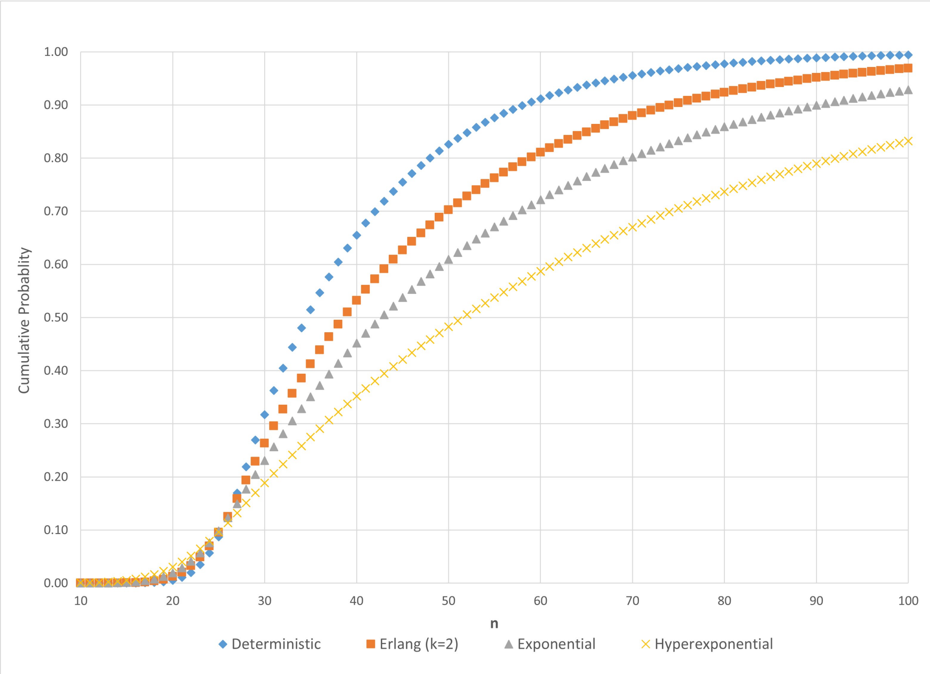

Figure 2. Cumulative Distribution Functions for $c = 30, \rho = 0.96\overline{6}$

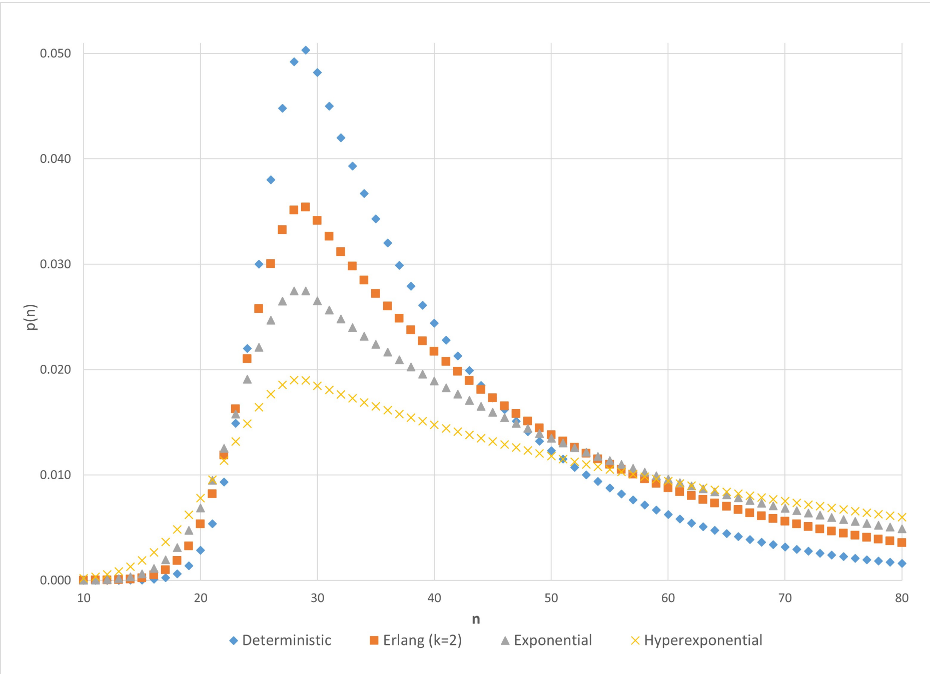

Figure 3. Probability Mass Functions for $c = 30, \rho = 0.96\overline{6}$

and iterate on $\pi(j)$, $j = N-1, \ldots, 0$, then normalize. The adaption algorithm to this case, in the appendix, specifies the necessary modifications. We note that, for $\rho \geq 1$ the distribution function becomes centered close to N and as $\rho \to \infty$, $P(N) \to 1$. Table 5 reports L and W and $P(N)$ for this case using the four distribution functions used for other numerical results. For the numerical results reported reported in Table 5, we use $N = 100$, $\lambda = 6.8$, $\mu = 1/3$, $c = 20$, resulting in a $\rho = 1.13$. In the case of the hyper-exponential we use $p = .941176470588$, $\lambda_1 = 8$, and $\lambda_2 = 2$ which gives a $\rho = 1.13$ similar to other distributions.

Table 5. Performance Measures when $\rho > 1$

Distribution	L	W	$P(N)$
Deterministic	96.2500	17.4504	0.18888
Erlang	94.3750	16.2058	0.14360
Exponential	92.5019	15.4170	0.11765
Hyper-exponential	89.8611	14.8160	0.10807

5. Concluding Remarks

A great majority of textbooks that deal with the $G/M/c$ model and its extensions introduce the stationary one-step transition probabilities in the form of Lemma 2.1 where the set of transition probabilities in Lemma 2.1(iii) are approximated using truncation and numerical integration. In this article it is shown that this indefinite integral can be converted into a finite sum using simple direct integration techniques. Thus we give a computationally efficient and exact finite sum expression for the transition probabilities. Moreover, we use a novel method to generate the one-step transition probabilities of the related batch arrival models in terms of the one-step transition probabilities of the $G/M/c$ model. For the finite buffer case we include models with partial and full batch rejection. To demonstrate the applicability of this method, we give examples and numerical results for several cases, including when $\rho \geq 1$. Numerical experience indicates that one needs to pay special attention to numerical stability of the algorithms.

References

[1] Baily, D. E., & Neuts, M. F. (1981). Algorithmic methods for multi-server queues with group arrivals and exponential services. *European Journal of Operational Research*, 8(2), 184–196.

[2] Chaudhry, M.L., & Kim, J.J. (2016). Analytically elegant and computationally efficient results in terms of roots for the $GI^X/M/c$ queueing system. *Queueing Systems*, 82, 237–257.

[3] Cosmetatos, G., & Godsave, S. (1980). Approximations in the multi-server queue with hyper-exponential inter-arrival times and exponential service times. *Journal of the Operational Research Society*, 31(1), 57–62.

[4] Cruz, F.R.B., Santos, M.A.C., Oliveira, F.L.P. & Quinino, R.C. (2021). Estimation in a general bulk-arrival Markovian multi-server finite queue. *Operational Research*, 21(1), 73–89, .

[5] El-Taha, M. (2021). An efficient convolution method to compute the stationary transition probabilities of the G/M/c model and its variants. In *10th IFIP International Conference on Performance Evaluation and Modeling in Wireless and Wired Networks (PEMWN), IEEE, 2021*, pages 1–6.

[6] El-Taha, M., & Michaud, T. (2022). An efficient method to compute the stationary probabilities of the $GI^X/M/c/N$ model. *https://doi.org/10.48550/arXiv.2212.09527*, pages 1–27.

[7] El-Taha, M., & Stidham, Jr., S. (1999). *Sample-Path Analysis of Queueing Systems*. Kluwer Academic Publishing, Boston, 1999.

[8] Ferreira, F., & Pacheco, A. (2006). Analysis of GI/M/s/c queues using uniformisation. *Computers and Mathematics with Applications*, 51, 291–304.

[9] Gontijo, G.M., Atuncar, G.S., Cruz, F.R.B., & Kerbache, L. (2011). Performance evaluation and dimensioning of GI^X/M/c/N systems through kernel estimation. *Mathematical Problems in Engineering*.

[10] Grassmann, W., & Tavakoli, J. (2014). Efficient methods to find the equilibrium distribution of the number of customers in GI/M/c queues. *IFOR*, 52, 197–205.

[11] Gross, D., Shortle, J.F., Thompson, J.M., & Harris, C. (2008). *Fundamentals of Queueing Theory*. John Wiley, New Jersey, 4th edition.

[12] Hokstad, P. (1975). The G/M/m queue with finite waiting room. *Journal of Applied Probability*, 12(4), 779–792.

[13] Kelly, F. (1979). *Reversibility and Stochastic Networks*. Wiley, New York.

[14] Kim, J.J., & Chaudhry, M.L. (2017). A novel way of treating the finite-buffer queue $GI/M/c/N$ using roots. *International Journal of Mathematical Models and Methods in Applied Sciences*, 11.

[15] Kleinrock, L. (1975). *Queueing Systems, vol. I*. Wiley Intersciences, New York.

[16] Laxmi, P. V. & Gupta, U.C. (2000). Analysis of finite-buffer multi-server queues with group arrivals: $GI^X/M/c/N$. *Queueing Systems*, 36(1-3), 125–140.

[17] Medhi, J. (2003). *Stochastic Models in Queueing Theory*. Academic Press, New York, 2nd edition.

[18] Neuts, M.F. (1981). *Matrix-Geometric Solutions in Stochastic Models-An Algorithmic Approach*. Johns Hopkins University Press.

[19] Ross, S.M. (2007). *Introduction to Probability Models*. Academic Press, San Diego, 9th edition.

[20] Stidham, Jr., S., & El-Taha, M. (1989). Sample-path analysis of processes with imbedded point processes. *Queueing Systems*, 5, 131–165.

[21] Takács, L. (1957). On a queueing problem concerning telephone traffic. *Acta Math.Acad.Sci.Hungar.*, 8, 325–335.

[22] Takács, L. (1962). *Introduction to the theory of queues*. Oxford University Press, New York.

[23] Tijms, H.C. (2003). *A First Course in Stochastic Models*. Wiley, New York.

[24] Yao, D.D., Chaudhry, M.L., & Templeton, J.G.C. (1984). A note on some relations in the queue $GI^X/M/c$. *Operations Research Letters*, 3(1), 53–56 .

[25] Zhao, Y. (1994). Analysis of the GI^X/M/c model. *Queueing Systems*, 15, 347–364.

A. Appendix

The direct algorithm presented here is used to compute the stationary probabilities. Our computational methodology follows Sections 2 and 3, wherein the transition probabilities are prepared following the examples described in the first part of Section 4. Moreover, $|\sigma| < 1$ is determined from $A^*(c\mu(1-\sigma)) = \sigma$. Below we give an algorithm for the multi-server model, followed by an adaptation to the finite buffer batch-arrival multi-server cases.

Algorithm for the $GI/M/c$ Model

Initialization. Let ϵ be the maximum allowable error, c be the number of servers each with mean rate μ, and λ be the mean arrival rate. We also input the LST of the inter-arrival times distribution and number of phases and/or weights if applicable. Note that for large scale examples we determine N using ϵ.

1. Compute $\rho = \lambda/c\mu$ and check that $\rho < 1$ (i.e.: a long-run solution exists).
2. For each specified inter-arrival distribution, compute $A^*(s)$ for $s = k\mu$ where $k = 1, 2, ..., c$ and $A_n^*(c\mu)$ for $n = 1, 2, ..., N-c+1$

   Deterministic: $A^*(s) = e^{-s/\lambda}$, $A_n^*(c\mu) = \lambda^{-n} e^{-c\mu/\lambda}$

   Exponential: $A^*(s) = \frac{\lambda}{s+\lambda}$, $A_n^*(c\mu) = \frac{n!\lambda}{(c\mu+\lambda)^{n+1}}$

   Erlang (two-phase): $A^*(s) = \frac{4\lambda^2}{(s+2\lambda)^2}$, $A_n^*(c\mu) = n!(n+1)\frac{4\lambda^2}{(c\mu+2\lambda)^{n+2}}$

Hyper-exponential (two phase): $A_n^*(s) = n!\left[\frac{p\lambda_1}{(s+\lambda_1)^{n+1}} + \frac{(1-p)\lambda_2}{(s+\lambda_2)^{n+1}}\right]$, and $A_n^*(c\mu) = n!\left[\frac{p\lambda_1}{(c\mu+\lambda_1)^{n+1}} + \frac{(1-p)\lambda_2}{(c\mu+\lambda_2)^{n+1}}\right]$.

3. Root-solve by iterating over the following until $|\sigma_{n+1} - \sigma_n| < \epsilon$

$$\sigma_{n+1} = A^*[c\mu(1-\sigma_n)]$$

4. For a given ϵ, determine N (for large examples) using

$$N = \min\left\{n \in \mathcal{N} \mid n \geq c + \frac{\ln(\epsilon) - 2\ln(1-\sigma)}{\ln(\sigma)}\right\}$$

5. Define $p(i,j) = 0$ for all $i = 0, 1, ..., N-2, \;\; j = i+2, i+3, ..., N.$
6. Compute $p(i,j)$ for $i = 0, 1, 2, ..., c-1, \;\; j = 1, 2, ..., i+1$ using the computationally friendly form:

$$p(i,j) = \frac{(i+1)!}{j!}\sum_{k=0}^{i-j+1}\frac{(-1)^k A^*((j+k)\mu)}{(i-j-k+1)!k!} \ .$$

7. Compute $p(i,j)$ for $i = c, c+1, ..., N, \;\; j = c, c+1, ..., i+1, \quad i+1 \leq N$ using

$$p(i,j) = \frac{(c\mu)^{i-j+1}A^*_{i-j+1}(c\mu)}{(i-j+1)!}$$

8. Compute $p(i,j)$ for $i = c, c+1, ..., N, \;\; j = 1, 2, ..., c-1$ using the computationally friendly form:

$$p(i,j) = \frac{c!}{j!}\sum_{k=1}^{c-j}\frac{(-1)^{(c-j-k)}}{k!(c-j-k)!}\left(\frac{c}{k}\right)^{i-c+1}\left[A^*((c-k)\mu) - \sum_{r=0}^{i-c+1}\frac{(k\mu)^r A_r^*(c\mu)}{r!}\right]$$

9. Compute $p(i,j)$ for $i = c, c+1, ..., N, \;\; j = 0$ using

$$p(i,j) = 1 - \sum_{n=1}^{N} p_{i,n}$$

10. Define $a(k,j) = 0$ for $k = 0, 1, ..., N, \;\; j = k, k+1, ..., N$
11. Compute $a(k,j)$ for $k = j+1, j+2, ..., N, \;\; j = 0, 1, ..., c-1$ using

$$a(k,j) = \sum_{i=0}^{j} p(k,i)$$

12. Compute $\pi'(j) = \sigma^j$ for $j = c, c+1, ..., N$
13. Compute $\pi'(j)$ for $j = c-1, c-2, ..., 0$ recursively using

$$\pi'(j) = \frac{\sum_{k=j+1}^{N}\pi'(k)a(k,j)}{p(j,j+1)}$$

14. Compute $\pi(j)$ for $j = 0, 1, ..., N$ by normalizing π'_j using

$$\pi(j) = \frac{\pi'(j)}{\Phi}$$

where

$$\Phi = \sum_{k=0}^{N} \pi'(k) = \sum_{k=0}^{c-1} \pi'(k) + \sum_{k=c}^{N} \sigma^k = \sum_{k=0}^{c-1} \pi'(k) + \frac{\sigma^c(1-\sigma^{N-c+1})}{(1-\sigma)}$$

15. Compute $p(0)$ using

$$p(0) = (1-\rho) + \rho\pi(N) - \rho\sum_{k=0}^{c-2} \frac{c-k-1}{k+1}\pi(k)$$

16. Compute $p(n)$ for $n = 1, 2, ..., c-1$ using

$$p(n) = \frac{c\rho\pi(n-1)}{n}$$

17. Compute $p(n)$ for $n = c, c+1, ..., N$ using

$$p(n) = \rho\pi(n-1)$$

18. Compute performance measures using $E[L] = \sum_{i=1}^{N} ip(i)$ and $E[W] = E[L]/\lambda$.

Below we provide adaptations of this algorithm to related models by specifying the steps that need to be modified.

Adaptation to the $GI/M/c/N$ model with $\rho < 1$

In this case we replace step 4 by inputing N as a predetermined value, and adjust the last row of the transition matrix so that $p(N, i) = p(N-1, i), i = 0, \ldots, N$.

Adaptation to the $GI/M/c/N$ model with $\rho \geq 1$

In this case we modify the following steps.

1. Modify step 1 to check for $\rho \geq 1$
2. Remove step 3.
3. Insure that $p(N, i) = p(N-1, i), i = 0, \ldots, N$
4. Modify step 4 to input N as a predetermined value
5. Modify step 11 to iterate on $j = 0, 1, \ldots, N-1$
6. Modify step 12 to input $\pi'(N) = 1$
7. Modify step 13 so that $j = N-1, \ldots, 0$.

Adaptation to the $GI^X/M/c/N$ model

We refer to cases 2 and 3 in Section 3.

1. Repeat steps 1 and 2 only in the above algorithm.
2. Compute $p^*(i, j)$ as in Section 3 for case 2 and case 3