TECHNICAL REPORT

An Algorithmic Analysis of the MMPP/G/1 Queue

by Levent Gun

ISR TR 88-40

AN ALGORITHMIC ANALYSIS OF THE $MMPP/G/1$ QUEUE

Levent Gün*

Electrical Engineering Department and Systems Research Center
University of Maryland, College Park, Maryland 20742.

ABSTRACT

A single server queue with general service time distribution is considered when the input is a *Markov modulated Poisson process (MMPP)*. An *algorithmic* solution to the transform of the stationary delay and queue length distributions is summarized, and recursive closed-form expressions are obtained for the moments of these distributions. The numerical implementation of these results is discussed in detail with particular reference to an algorithm due to Lucantoni and Ramaswami [11] and its accelerated version due to Ramaswami [19]. This algorithm is shown to be an efficient tool in the matrix-analytic solution of many stochastic models, as various steps for saving considerable amounts of unnecessary computations are identified. A special case of the model where the service time distribution is of *phase type* is discussed and the stationary queue length distribution at arbitrary times is obtained in *matrix-geometric* form. Finally, the matrix-geometric and the $M/G/1$ approaches are compared through this special case.

* Most of this research was performed under the supervision of Dr. Harry Heffes during the author's stay at AT&T Bell Laboratories, Holmdel, N.J., as a summer student during the summer of 1986. This research was also supported by the Office of Naval Research through Grant N00014-84-K-0614 and through a grant from AT&T Bell Laboratories.

1. INTRODUCTION

The basic motivation behind the work reported in this paper is to explore the computational aspects of matrix-analytic solution techniques in the study of many probabilistic models arising in applications. These methods were pioneered by Neuts and have been applied successfully to solve many stochastic models [7, 10-15]. Central to the matrix-analytic approach is the entrywise minimal nonnegative solution of non-linear matrix equations of the form

$$G = \sum_{n=0}^{\infty} A_n G^n \quad \text{or} \quad R = \sum_{n=0}^{\infty} R^n A_n. \tag{1.1}$$

Several iterative algorithms for solving these matrix equations are reviewed in the Appendix. For the $M/G/1$ type queues such as the $PH/G/1$ and the $MMPP/G/1$ queues, an efficient and numerically stable algorithm to compute these matrices without having to compute and store the matrices $\{A_n\}_0^\infty$ was given in the work of Lucantoni and Ramaswami [11]. In this paper, insights gained on the behavior of this algorithm through numerical studies are reported. In view of these observations, a simple extension based on linear extrapolation is proposed for the computation of the G matrix. This minor extension leads to a new stopping criterion and results in considerable amounts of savings in the computations for a given accuracy.

This paper illustrates the above mentioned methodology and algorithmic results by studying both the analytical and computational aspects of the stationary waiting time and the queue length distributions for $MMPP/G/1$ queueing systems. The $MMPP/G/1$ queue is a single server system with general service time distribution where the arrivals are modeled by a *Markov modulated Poisson Process (MMPP)*; the first-come-first-served (FCFS) service discipline is enforced. The *MMPP* is a doubly stochastic Poisson process whose rate is determined by the state of a continuous-time Markov chain, and it has great appeal in its applicability in modeling many problems [6, 7].

The $MMPP/G/1$ queue is a special case of the more general $N/G/1$ queue studied by Ramaswami [17]. Most of the analytical results mentioned here can also be obtained directly from the results obtained in [17] by specializing the *N-process* to a *MMPP*. However, the simplicity of the $MMPP/G/1$ queue leads to

closed-form expressions in the *recursive* calculation of higher order moments of the above-mentioned distributions.

This paper is organized as follows. In Section 2, the queueing model is described and equations describing the transition probability matrix are obtained. In Section 3, an algorithm from reference [7] is briefly outlined to compute the distributions of the *virtual* waiting time and of the waiting time seen by an *arriving* customer. In Section 4, transforms of the queue length distributions at departure epochs and at arbitrary times are obtained. In both Sections 3 and 4, *recursive* expressions are given for all the moments of these distributions. Section 5 contains various insights obtained through numerical experimentation of the Lucantoni-Ramaswami algorithm on the $MMPP/G/1$ queue. An accelerated version of this algorithm due to Ramaswami is also discussed and numerical comparisons are made. Finally, in Section 6, the stationary queue length distribution and its first two moments are obtained at an *arbitrary* time in the special case where the service time distribution is of *phase type*. Computational requirements in this special case are also discussed and compared with the algorithm available for the general service distributions.

A word on the notation used hereafter: The $r \times r$ identity matrix is denoted by I_r and the $r \times 1$ column vector of ones is denoted by e_r, while the $1 \times r$ dimensional row vector with zero entries is denoted by 0_r. The notation $\mathbf{O}$ is used for the zero matrix with appropriate dimensions. Also, the spectral radius of matrix X is denoted by $sp(X)$ and "^{T}" is used to denote the transpose operator.

2. THE MODEL

Consider an m-state irreducible continuous-time Markov chain with infinitesimal generator matrix S and the stationary probability distribution vector π. When the Markov chain is in state i, $1 \leq i \leq m$, the arrivals are Poissonian with rate λ_i and fed into a FCFS single server queue whose service times are independent and identically distributed with common distribution function $H(\cdot)$. Let λ be the $m \times 1$ column vector with i^{th} component λ_i. The k^{th} moment of the service time distribution about the origin is denoted by $\mu^{(k)}$ and the $m \times m$ diagonal matrix with elements λ_i along the diagonal is denoted by Λ. It is assumed that the system parameters are such that the queueing system is a stable one, i. e., $\pi\lambda\mu^{(1)} < 1$. Let

$\{\tau_n : n \geq 0\}$ denote the successive departure epochs, with $\tau_0 = 0$, and define $X(t)$ and $J(t)$ to be the number of customers in the system and the phase of the $MMPP$ at time t^+, respectively. The sequence $\{(X(\tau_n), J(\tau_n), \tau_{n+1} - \tau_n) : n \geq 0\}$ form a semi-Markov sequence on the state space $\{0, 1, \ldots\} \times \{1, \ldots, m\}$ with transition probability matrix $Q(\cdot)$ given by

$$Q(x) = \begin{pmatrix} B_0(x) & B_1(x) & B_2(x) & \cdot & \cdot & \cdot \\ A_0(x) & A_1(x) & A_2(x) & \cdot & \cdot & \cdot \\ \mathbf{O} & A_0(x) & A_1(x) & \cdot & \cdot & \cdot \\ \mathbf{O} & \mathbf{O} & A_0(x) & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}, \quad x \geq 0, \tag{2.1a}$$

where the $m \times m$ block entries are given by

$$A_n(x) = \int_0^x P(n,t)\, dH(t)\,, \qquad n = 0, 1, \cdots, \quad x \geq 0, \tag{2.1b}$$

$$B_n(x) = U * A_n(x), \qquad n = 0, 1, \cdots, \quad x \geq 0, \tag{2.1c}$$

with

$$P_{ij}(n,t) = P\{X(t) = n,\ J(t) = j | X(0) = 0,\ J(0) = i\}\,, \quad 1 \leq i, j \leq m\,, \tag{2.1c}$$

and

$$U(x) = \int_0^x P(0,t)\, \Lambda\, dt\,. \tag{2.1d}$$

The operator * in equation (2.1c) is the matrix convolution operator. By using standard techniques [8], the matrices $P(n,t)$ can be shown to satisfy the following Chapman-Kolmogorov equations

$$\frac{dP(n,t)}{dt} = \begin{cases} P(n,t)\,(S - \Lambda) + P(n-1,t)\,\Lambda\,, & \text{if } n = 1, 2, \cdots, \\ P(0,t)\,(S - \Lambda)\,, & \text{if } n = 0. \end{cases} \tag{2.2}$$

3. STATIONARY WAITING TIME DISTRIBUTIONS

An algorithm to compute the distributions of the *virtual* waiting time and of the waiting time seen by an *arriving* customer is given in [7]. For sake of completeness, the steps of this algorithm are briefly outlined below.

1. Compute the $m \times m$ irreducible stochastic matrix G and its stationary probability vector g. The G matrix is the entrywise minimal nonnegative solution of the non-linear matrix equation $G = \sum_{n=0}^{\infty} A_n G^n$, with $A_n := A_n(\infty)$. For $1 \leq i, j \leq m$, G_{ij} denote the probability that a busy period starting with the $MMPP$ in phase i ends in phase j. The matrix G is a key ingredient in obtaining most quantities of interest such as the busy period and waiting time characteristics and is studied extensively by Neuts [14]. An iterative procedure for computing the matrix G was given by Lucantoni and Ramaswami [11] and is discussed in Section 5.
2. Compute the $m \times m$ stochastic matrix $A := \sum_{n=0}^{\infty} A_n = \int_0^{\infty} e^{St}\, dH(t)$. Here, for $1 \leq i, j \leq m$, A_{ij} is the probability that a service time ends with the $MMPP$ in phase j given that the service began in phase i.
3. Compute the $m \times m$ stochastic matrix $U := U(\infty) = (\Lambda - S)^{-1}\Lambda$. For $1 \leq i, j \leq m$, U_{ij} is the probability that the first arrival to a busy period arrives with the $MMPP$ in phase j given that the last departure from the previous busy period departed with the $MMPP$ in phase i.
4. Compute the $m \times 1$ column vectors

$$\beta = \mu^{(1)}(\pi\lambda)e_m + (S + e_m\pi)^{-1}(A - I_m)\lambda$$

and

$$\mu = (I_m - G + e_m g)[I_m - A + e_m g - \beta g]^{-1} e_m \ .$$

For $1 \leq i \leq m$, the i^{th} components β_i and μ_i are the expected number of arrivals during a service that began in phase i and the expected number of departures during a busy period that began in phase i, respectively.
5. Compute the $1 \times m$ row vector $x_0 = (dU\mu)^{-1}d$, where the $m \times 1$ row vector d is such that $dUG = d$ and $de_m = 1$, with the interpretation that the i^{th} component of x_0 is the stationary probability that a *departure* leaves the system empty with the $MMPP$ in phase i.
6. Compute the $1 \times m$ row vector $y_0 = (\pi\lambda)\, x_0\, (\Lambda - S)^{-1}$ where the i^{th} component of y_0 is the stationary probability of the system being empty and that the phase of the $MMPP$ is i at an *arbitrary* time.

7. The Laplace-Stieltjes transform (LST) of the virtual delay distribution is given by $\widetilde{W}(s)e_m$ where

$$\widetilde{W}(s) = \begin{cases} sy_0\,[sI_m + S - \Lambda + \Lambda\widetilde{H}(s)]^{-1} & \text{if } s > 0, \\ \pi & \text{if } s = 0, \end{cases} \tag{3.1}$$

and $\widetilde{H}(\cdot)$ is the LST of $H(\cdot)$. The i^{th} component of $W(x)$, the inverse transform of the $1 \times m$ row vector $\widetilde{W}(s)$, is the joint probability that at an arbitrary time the $MMPP$ is in phase i and that a *virtual* customer who arrived at that time would wait less than or equal to x time units before entering service.

8. Finally, the waiting time distribution $W_a(\cdot)$ seen as by an arrival is given by

$$W_a(x) = (\pi\lambda)^{-1}\, W(x)\lambda\,. \tag{3.2}$$

The first two moments of the virtual waiting time distribution are available in [7]. Here, in view of the equations (3.1) and (3.2) *recursive* expressions are given for calculating *arbitrary* moments of the waiting time distributions. Let ρ be the traffic intensity given by $\rho = \pi\lambda\mu^{(1)}$, and pose

$$\widetilde{W}^{(n)} := \begin{cases} \widetilde{W}(0) & \text{if } n = 0, \\ \frac{d^n}{d^n s}\widetilde{W}(s)\Big|_{s=0+} & \text{if } n = 1, 2, \cdots. \end{cases}$$

Lemma 3.1 *The moments of the virtual waiting time distribution are given by*

$$\begin{aligned} E(W^n) &= (-1)^n \widetilde{W}^{(n)} e_m \\ &= \frac{(-1)^{n+1}}{(n+1)(1-\rho)}[(n+1)\mu^{(1)}d_n + c_{n+1}]\,\lambda\,, \quad n = 1, 2, \cdots \end{aligned} \tag{3.3a}$$

where the $1 \times m$ *vectors* $\{c_n\}_2^\infty$ *and* $\{d_n\}_1^\infty$ *are defined by*

$$c_n := \sum_{m=2}^{n} (-1)^m \mu^{(m)} \binom{n}{m} \widetilde{W}^{(n-m)}\,, \qquad n = 2, 3, \cdots$$

$$d_n := \begin{cases} \left[\pi(I_m - \mu^{(1)}\Lambda) - y_0\right](S + e_m\pi)^{-1}\,, & \text{if } n = 1 \\ \left[n\widetilde{W}^{(n-1)}(I_m - \mu^{(1)}\Lambda) + c_n\Lambda\right](S + e_m\pi)^{-1}, & \text{if } n = 2, 3, \cdots. \end{cases}$$

The $1 \times m$ *vectors* $\{\widetilde{W}^{(n)}\}_0^\infty$ *in these definitions are calculated by the relations*

$$\widetilde{W}^{(0)} = \pi \ , \qquad \widetilde{W}^{(n)} = (-1)^{(n)} E(W^n)\,\pi - d_n \qquad n = 1, 2, \cdots \ . \tag{3.3b}$$

Proof: Rewrite equation (3.1) as

$$\widetilde{W}(s)[sI + S - \Lambda + \Lambda \widetilde{H}(s)] = sy_0 \ , \quad s > 0 \ ,$$

and differentiate it n times. Upon letting $s \to 0^+$ in the resulting relation and using the definition of c_n

$$\widetilde{W}^{(n)} S = \begin{cases} y_0 - \pi[I_m - \mu^{(1)}\Lambda] \ , & \text{if } n = 1, \\ -n\widetilde{W}^{(n-1)}[I_m - \mu^{(1)}\Lambda] - c_n\Lambda \ , & \text{if } n = 2, 3, \cdots . \end{cases} \tag{3.4}$$

Add $\widetilde{W}^{(n)} e_m \pi$ to both sides of equation (3.4). Noting that $\pi(S + e_m\pi)^{-1} = \pi$ and using the definition of d_n give equation (3.3b) for $n = 1, 2, \cdots$. Equation (3.3b) is trivially true for $n = 0$ since $\widetilde{W}(0) = \pi$. Now replace n with $n + 1$ in (3.4) and premultiply by e_m, the desired result (3.3a) is readily obtained after some simplifications. □

The moments of the stationary waiting time distribution as seen by an arbitrary arrival now follows from equation (3.2) and are given by

$$E(W_a^n) = (-1)^n (\pi\lambda)^{-1} W^{(n)} \lambda \ , \quad n = 1, 2, \cdots . \tag{3.5}$$

4. THE STATIONARY QUEUE LENGTH DISTRIBUTIONS

In this section, results from the reference [17] are specialized to obtain equations satisfied by the transforms of the stationary queue length distributions at points of departures and at arbitrary times. These equations are then used to generate closed form *recursive* expressions to obtain the moments of these distributions.

Let the stationary queue length density at a departure point be denoted by the row vector x which is partitioned into $1 \times m$ row vectors as $x := (x_0, x_1, \ldots)$, where the i^{th}, $1 \le i \le m$, component of x_k is the joint probability that at a departure epoch there are k customers in the system and that the $MMPP$ is in phase i. The

vector x is the invariant probability vector of the irreducible stochastic matrix $Q(\infty)$ and thus satisfies the equations

$$x\,Q(\infty) = x\ , \qquad x\,e_\infty = 1\,. \tag{4.1}$$

Equation (4.1) can be rewritten in terms of the block entries as

$$x_k = x_0\,B_k + \sum_{l=1}^{k+1} x_l\,A_{k-l+1}\ , \qquad k = 0, 1, \cdots\ , \tag{4.2}$$

where $B_k := B_k(\infty)$, $k = 0, 1, \cdots$. Multiplication of equation (4.2) by the complex number z^k and summing over $k = 0, 1, \cdots$, yield the equation

$$X(z)\,[zI_m - A(z)] = x_0\,[zU - I_m]\,A(z)\ , \quad |z| \le 1, \tag{4.3}$$

where

$$X(z) := \sum_{k=0}^{\infty} x_k\,z^k \quad \text{and} \quad \mathrm{A(z)} := \sum_{\mathrm{k=0}}^{\infty} \mathrm{A_k\,z^k}\ , \qquad \mathrm{|z| \le 1}\,.$$

From equation (2.1b) it easily follows that

$$A(z) = \int_0^\infty P(z,t)\,dH(t)\ , \tag{4.4}$$

where

$$P(z,t) := \sum_{k=0.}^{\infty} P(k,t)\,z^k\ , \quad |z| \le 1\,.$$

In view of equation (2.2), $P(z,t)$ satisfies the following differential equation

$$\frac{dP(z,t)}{dt} = P(z,t)\,(S - \Lambda) + zP(z,t)\Lambda\ , \quad t \ge 0$$
$$P(z,0) = I_m\ ,$$

which is readily solved to yield

$$P(z,t) = e^{\{S-(1-z)\Lambda\}t}\ , \quad |z| \le 1 \quad \text{and} \quad \mathrm{t \ge 0.} \tag{4.5}$$

Therefore, the matrix $A(z)$ of equation (4.3) is given by

$$A(z) = \int_0^\infty e^{\{S-(1-z)\Lambda\}t}\, dH(t)\ , \qquad |z| \leq 1\ . \tag{4.6}$$

Next, an explicit expression is obtained for the quantity $X(1^-) = \sum_{k=0}^{\infty} x_k$, which is needed in the computation of the moments of the queue length distribution. Letting $z \to 1^-$ in equation (4.3) yields

$$X(1^-)\,[I_m - A] = x_0\,[U - I_m]\,A\ . \tag{4.7}$$

Note by the definition of the stochastic matrix A and the row vector π that π is also the stationary probability vector of A. Addition of $\pi = (X(1^-)e_m)\,\pi$ to both sides of the equation (4.7) readily yields

$$X(1^-) = x_0(U - I_m)A(I_m - A + e_m\pi)^{-1} + \pi\ ,$$

through the use of the fact that $\pi(I_m - A + e_m\pi)^{-1} = \pi$. Invertibility of the stochastic matrix $(I_m - A + e_m\pi)$ is shown in Kemeny and Snell, [9, p. 100].

Simplicity of the equation (4.3) allows a *recursive* computation of the moments of the queue length distribution at departure epochs. To that end, pose

$$X^{(0)} := X(1^-), \qquad A^{(0)} := A, \qquad n = 0$$

$$X^{(n)} := \tfrac{d^n X(z)}{d^n z}|_{z=1^-}, \quad A^{(n)} := \tfrac{d^n A(z)}{d^n z}|_{z=1^-}, \quad n = 1, 2, \cdots\ .$$

The results from the next lemma can be proved by using arguments similar to the ones in the proof of Lemma 3.1.

Lemma 4.1 *The moments of the queue length distribution at departure epochs are given by*

$$E(X^n) = X^{(n)}e_m \tag{4.8a}$$

$$= \frac{1}{(n+1)(1-\rho)}[(n+1)d_nA^{(1)} + x_0V_{n+1} - c_{n+1}]e_m\ , \qquad n = 1, 2, \cdots,$$

where the $m \times m$ matrices $\{V_n\}_1^\infty$ and the $1 \times m$ vectors $\{c_n\}_1^\infty$ and $\{d_n\}_1^\infty$ are

defined by

$$V_n := (U - I_m)A^{(n)} + nUA^{(n-1)}\,, \quad n = 1, 2, \cdots,$$

$$c_n := \begin{cases} 0 & \text{if } n = 1, \\ \sum_{m=2}^{n} \binom{n}{m} X^{(n-m)} A^{(m)} & \text{if } n = 2, 3, \cdots, \end{cases}$$

$$d_n := [x_0 V_n + c_n - nX^{(n-1)}(I_m - A^{(1)})](I_m - A + e_m\pi)^{-1}\,, \quad n = 1, 2, \cdots.$$

The $1 \times m$ *vectors* $\{X^{(n)}\}_0^\infty$ *in these definitions are calculated as*

$$X^{(0)} := X(1^-)\,, \qquad X^{(n)} = E(X^n)\pi + d_n\,, \quad n = 1, 2, \cdots. \tag{4.8b}$$

Let the stationary queue length density at an arbitrary time be denoted by the row vector y which is also partitioned into $1 \times m$ row vectors as $y := (y_0, y_1, \ldots)$, where

$$(y_k)_i = \lim_{t\to\infty} P[X(t) = k,\ J(t) = i | X(0) = k',\ J(0) = i'],\ 1 \le i, i' \le m\,,\ \ k, k' = 0, 1, \cdots.$$

The following result is a special case of Theorem 3.3.18 of [17, p. 246] and can be obtained using the Key Renewal Theorem [2].

Lemma 4.2 *The generating function* $Y(z) = \sum_{k=0}^{\infty} y_k z^k$ *is given by*

$$Y(z) = \begin{cases} (\pi\lambda)^{-1}(1 - z)X(z)\,[(1 - z)\Lambda - S]^{-1}\,, & \text{if } 0 \le z < 1\,, \\ \pi\,, & \text{if } z = 1\,. \end{cases} \tag{4.9}$$

Use of equation (4.9) and calculations similar to the ones made in the proof of Lemma 3.1 give the following result.

Lemma 4.3 *The moments of the stationary queue length distribution at arbitrary times are given by*

$$E(Y^n) = Y^{(n)} e_m\,, \qquad n = 1, 2, \cdots, \tag{4.10a}$$

$$= E(X^n) - n\left[X^{(n-1)} - (\pi\lambda)^{-1} Y^{(n-1)}\Lambda\right](S + e_m\pi)^{-1}\lambda\,,$$

where the $1 \times m$ *vectors* $Y^{(n)} := \frac{d^n Y(z)}{d^n z}|_{z=1^-}$, $n = 0, 1, \cdots$, *are given by*

$$Y^{(n)} = \begin{cases} \pi & \text{if } n = 0, \\ n\left[(\pi\lambda)X^{(n-1)} - Y^{(n-1)}\Lambda\right](S + e_m\pi)^{-1} + E(Y^n)\pi, & \text{if } n = 1, 2, \cdots. \end{cases} \tag{4.10b}$$

5. IMPLEMENTATION DETAILS – EXTENSIONS AND COMPARISONS

The central item in the algorithm outlined in Section 3 is the so-called G matrix introduced by Neuts [14]. An efficient and numerically stable way of computing the G matrix, *without* having to evaluate the integrals in (3.1), is given in the work of Lucantoni and Ramaswami [11]. This algorithm is based on the *randomization technique* of Grassman [3] and can be applied to the matrix-analytic solution of many stochastic models. A *sub-Newton* scheme is later incorporated to this algorithm by Ramaswami [19] to improve its rate of convergence. In this section an extrapolation method is suggested for these algorithms based on the insights gained through numerical studies. Although this proposed extension is a minor addition to the existing algorithms, it has been observed to improve their efficiency considerably especially when their convergence rates are slow.

THE RANDOMIZATION ALGORITHM

This above mentioned algorithm of Lucantoni and Ramaswami is called the *Randomization Algorithm* (RA) in the rest of this paper and uses the following theorem in Çınlar [2].

Theorem 5.1 *Let* S *be the infinitesimal generator of an* m*-state Markov process and suppose that* $-S_{ii} \leq \theta < \infty$ *for* $1 \leq i \leq m$. *Then the transition function* $P(t) \equiv exp(St)$ *may be written as*

$$P(t) = \sum_{n=0}^{\infty} e^{-\theta t} \frac{(\theta t)^n}{n!} K^n \ , \tag{5.1}$$

where K *is the stochastic matrix* $I_m + \theta^{-1} S$.

For the model described in Section 2, the RA can be given as

$$H_k^{(n+1)} = [I_m + \theta^{-1}(S - \Lambda)]\, H_k^{(n)} + \theta^{-1}\Lambda H_k^{(n)}\, G_k\,, \qquad n = 0, 1, \ldots, \tag{5.2}$$

$$G_{k+1} = \sum_{n=0}^{\infty} \gamma_n\, H_k^{(n)}\,, \qquad k = 0, 1, \ldots,$$

where $G_0 = \mathbf{O}$, $H_0^{(k)} = I_m$, $k = 0, 1, \cdots$, $\theta = \max_{1 \le i \le m}(\lambda_i - S_{ii})$ and $\gamma_n = \int_0^{\infty} e^{-\theta x} \frac{(\theta x)^n}{n!} dH(x)$, $n = 0, 1, \cdots$. It is shown in [11] that the sequence G_k converges monotonically to G. Note that γ_n is the probability that a service time has n epochs of a Poisson process with rate θ. In some useful special cases where the service distribution is either phase type or deterministic, γ_n may be computed *recursively* without numerical integration. These two cases are mentioned below.

1. When $H(\cdot)$ is *deterministic* with mass a, then

$$\gamma_0 = e^{-\theta a}\,, \qquad \gamma_n = \frac{\theta a}{n}\gamma_{n-1} \qquad n = 1, 2, \cdots\,. \tag{5.3}$$

2. When $H(\cdot)$ is of *phase type* with representation (α, A), it is shown by Neuts [14, pp. 59, 60] that $\{\gamma_n\}_0^{\infty}$ has discrete phase density with representation (β, B) given by

$$\beta = \theta\alpha(\theta I_m - A)^{-1}\,, \quad B = \theta(\theta I_m - A)^{-1} \tag{5.4a}$$

and

$$\beta_{m+1} = \alpha_{m+1} + \alpha(\theta I_m - A)^{-1}A^0\,, \quad B^0 = (\theta I_m - A)^{-1}A^0\,. \tag{5.4b}$$

For the notation and the proof of this result, the reader is referred to [14]. The probabilities $\{\gamma_n\}_0^{\infty}$ can then be computed by the recursion

$$\gamma_0 = \alpha_{m+1} + \eta(1)A^0\,, \quad \gamma_n = \eta(n+1)A^0, \quad n = 1, 2, \cdots\,. \tag{5.5a}$$

where

$$\eta(0) = \theta^{-1}\,\alpha\,, \quad \eta(n+1) = \eta(n)\,B\,, \quad n = 0, 1, \cdots\,. \tag{5.5b}$$

In practice the index n in iteration (5.2) is truncated at some positive integer value N. In view of the probabilistic interpretation of γ_n the truncation index N can be chosen to satisfy $\sum_{n=N+1}^{\infty} \gamma_n < \epsilon_\gamma$.

MODIFIED NEWTON-KANTOROVICH SCHEME (MNK)

The following modification is proposed by Ramaswami [19] to improve the rate of convergence of the RA. For this scheme, called RA_MNK hereafter, the two auxiliary substochastic matrices A_i, $i = 1, 2$, are needed. These matrices are given in [11] by

$$A_i = \sum_{n=i}^{\infty} \gamma_n K_i^{(n)} \ ,$$

where the matrices $K_i^{(n)}$ are recursively defined by $K_0^{(0)} = I$, $K_i^{(0)} = \mathbf{O}$, and

$$\begin{aligned} K_0^{(n+1)} &= K_0^{(n)}[I + \theta^{-1}(S - \Lambda)] \, , \\ K_i^{(n+1)} &= K_i^{(n)}[I + \theta^{-1}(S - \Lambda)] + K_{i-1}^{(n)}\theta^{-1}\Lambda \, , \qquad n \geq 0, \ i = 1, 2. \end{aligned} \tag{5.6}$$

Having computed G_{k+1} from the RA given in equation (5.2), the MNK step is now obtained by setting

$$Z_{k+1} = (I - A_1)^{-1}(G_{k+1} - G_k) \tag{5.7a}$$

and updating G_{k+1} by

$$G_{k+1} + A_1 Z_{k+1} + A_2(G_k Z_{k+1} + Z_{k+1} G_k) \rightarrow G_{k+1} \, . \tag{5.7b}$$

The RA_MNK scheme then proceeds by using this new G_{k+1} in equation (5.2) to get the next iterate G_{k+2}. It is shown in [19] that this scheme also converges to the G matrix.

PROPOSED EXTENSION BASED ON LINEAR EXTRAPOLATIONS

The algorithms RA and RA_MNK are stopped when all the entries in the successive iterates of the G matrix are within a small number, say ϵ, of each other, i. e.,

$$\max_{1 \leq i,j \leq m} |(G_{k+1})_{ij} - (G_k)_{ij}| < \epsilon \, . \tag{5.8}$$

Computational experience has shown that when the convergence rate is slow (when ρ is large), even when the successive iterates satisfy (5.8), say at iteration K, the matrix G_K can be far away from the limiting G matrix, i. e., $G_{ij} - (G_K)_{ij}$ can be

as much as 10^2 to 10^4 times ϵ for some i and j depending on ρ. However, the rows of G_K, when considered as vectors, were observed to be almost colinear with the corresponding rows of the G matrix. Therefore, the most recent iterates, G_K and G_{K-1}, can be used to obtain an approximation G_K^* to the *stochastic* matrix G by linear extrapolation, i. e.,

$$G_K^* = G_K + L\,[G_K - G_{K-1}]\ , \tag{5.9}$$

where L is a *diagonal* matrix so that the matrix G_K^* is stochastic. Computational experience has shown that even when G_K is far away (relative to ϵ) from being a stochastic matrix, G_K^* is typically much closer to G and use of G_K^* leads to much more accurate results in the calculation of the performance measures of interest.

The above observations suggest that if G_k^* is the stochastic matrix obtained from the linear extrapolation of G_k and G_{k-1}, then for the same ϵ, the stopping criterion

$$\max_{1\le i,j\le m} |(G_{k+1}^*)_{ij} - (G_k^*)_{ij}| < \epsilon\ , \tag{5.10}$$

should be satisfied at an iteration K^* which is much smaller than K, and $G_{K^*}^*$ should be closer to G than G_K. Indeed, the stochastic matrix $G_{K^*}^*$ were observed to be much closer to G than G_K in all the numerical examples performed, and the number of iterations K^* was much smaller than K especially for large values of ρ. Note that the computation of the matrices G_k^* from G_k and G_{k-1}, $k > 1$, requires only of order m^2 additional operations, whence do not have a significant effect on the total CPU time, especially for large m, since each iteration is of order Nm^3.

NUMERICAL EXAMPLES

These claims are supported through many numerical examples, some of which are included in Table 5.1, where the RA and the RA_MNK schemes are also compared with and without the proposed extrapolations. The numerical examples in Table 5.1 correspond to the $MMPP/D/1$ queue where the service times are chosen as 4msec, 6msec, 8msec, 10msec, $11\frac{1}{3}$msec and $3\frac{1}{3}$msec for Examples 5.1.1-5.1.6, respectively. The input process is the superposition of $m-1$ *identical* two-state Markov processes with states denoted by **1** and **0** and the transition rate from state **i** to state **j** is denoted by r_{ij}, $i,j=0,1$. When the process is in state **i** it produces *Poisson* arrivals with rate λ^i, $i=0,1$. The input process is therefore an m state $MMPP$ with a tridiagonal generator matrix S and arrival rates λ_i given by

$$S_{i,i-1}=(i-1)r_{10}\,, \qquad S_{i,i}=-S_{i,i-1}-S_{i,i+1}\,,$$
$$S_{i,i+1}=(m-i)r_{01}\,, \qquad \lambda_i=\lambda^1(i-1)+\lambda^0(m-i)\,, \qquad \text{for } 1\le i\le m\,.$$

Note that the state of the $MMPP$ denotes the number of the component processes that are in state **1**, with states 1 and m of the $MMPP$ corresponding to the situation where all the Markov processes are in state **0** and in state **1**, respectively. The following numerical values are used in all there examples: $r_{01}=2.5618$, $r_{10}=1.586$, $\lambda^0=0.00349$, $\lambda^1=57.416$.

In Table 5.1, K and K^* denote the number of iterations required to satisfy the stopping criterions (5.8) and (5.10), respectively. The expected value of the virtual waiting time is displayed for comparison. The other computed performance measures have shown a typical behavior. Here, EW and EW^* correspond to the expected virtual waiting time obtained using the stopping criterions (5.8) and (5.10), respectively.

TABLE 5.1

Example 5.1.1. $m = 5$, $\rho = 0.351302$, $\epsilon_\gamma = 10^{-7}$ $(N = 9)$.

	ϵ	K	EW	CPU	K^*	EW^*	CPU
RA	10^{-3}	20	3.73625e-3	1.9	14	1.68891e-3	1.4
	10^{-5}	42	1.67364e-3	3.5	30	1.65542e-3	2.5
	10^{-7}	64	1.65464e-3	4.9	50	1.65447e-3	3.8
	10^{-9}	85	1.65446e-3	6.3	70	1.65445e-3	5.1
RA_MNK	10^{-3}	15	3.00642e-3	1.9	11	1.67773e-3	1.6
	10^{-5}	29	1.66853e-3	3.0	22	1.65499e-3	2.4
	10^{-7}	44	1.65457e-3	4.2	35	1.65446e-3	3.6
	10^{-9}	58	1.65446e-3	5.5	48	1.65445e-3	4.4

Example 5.1.2. $m = 5$, $\rho = 0.526953$, $\epsilon_\gamma = 10^{-7}$ $(N = 11)$.

	ϵ	K	EW	CPU	K^*	EW^*	CPU
RA	10^{-3}	49	1.72419e-2	4.6	29	1.26234e-2	2.8
	10^{-5}	109	1.25098e-2	9.4	70	1.24573e-2	6.1
	10^{-7}	171	1.24560e-2	14.3	112	1.24557e-2	9.2
RA_MNK	10^{-3}	36	1.56773e-2	4.1	22	1.25645e-2	2.7
	10^{-5}	76	1.24871e-2	7.7	49	1.24567e-2	5.0
	10^{-7}	116	1.24560e-2	11.5	77	1.24557e-2	7.6

Example 5.1.3. $m = 5$, $\rho = 0.7026045$, $\epsilon_\gamma = 10^{-7}$ $(N = 13)$.

	ϵ	K	EW	CPU	K^*	EW^*	CPU
RA	10^{-3}	78	9.56070e-2	8.2	38	8.10129e-2	4.2
	10^{-4}	138	8.22618e-2	14.0	60	8.07974e-2	6.1
	10^{-5}	198	8.09259e-2	20.2	82	8.07759e-2	8.3
	10^{-7}	320	8.07750e-2	32.2	126	8.07735e-2	12.4
RA_ MNK	10^{-3}	59	9.04182e-2	6.7	29	8.09121e-2	3.7
	10^{-4}	99	8.17241e-2	10.8	43	8.07884e-2	4.9
	10^{-5}	139	8.08685e-2	14.8	58	8.07749e-2	6.5
	10^{-7}	219	8.07745e-2	23.0	86	8.07735e-2	9.2

Example 5.1.4. $m = 5$, $\rho = 0.8782556$, $\epsilon_\gamma = 10^{-7}$ $(N = 14)$.

	ϵ	K	EW	CPU	K^*	EW^*	CPU
RA	10^{-3}	115	0.558680	11.5	35	0.515955	4.1
	10^{-4}	261	0.520344	24.6	54	0.515631	6.0
	10^{-5}	422	0.516074	45.1	72	0.515603	7.7
RA_MNK	10^{-3}	93	0.544938	12.1	27	0.515794	3.9
	10^{-4}	194	0.518730	24.3	39	0.515619	5.1
	10^{-5}	302	0.515912	37.0	51	0.515601	6.8

Example 5.1.5. $m = 5$, $\rho = 0.995356$, $\epsilon_\gamma = 10^{-7}$ $(N = 15)$.

RA	10^{-3}	127	21.9231	14.3	32	21.8105	3.9
	10^{-4}	475	21.8414	52.5	47	21.8102	5.5
	10^{-5}	1539	21.8175	170.4	62	21.8102	7.1
RA_MNK	10^{-3}	111	21.8998	15.0	24	21.8104	3.5
	10^{-4}	396	21.8349	50.5	35	21.8102	5.0
	10^{-5}	1255	21.8157	155.8	45	21.8102	6.2

Example 5.1.6. $m = 12$, $\rho = 0.805067$, $\epsilon_\gamma = 10^{-7}$ $(N = 13)$.

RA	10^{-3}	132	7.86920e-2	118.4	49	5.91192e-2	45.6
	10^{-4}	250	6.03862e-2	220.0	93	5.85371e-2	84.9
	10^{-5}	367	5.86599e-2	327.8	138	5.84762e-2	120.5
RA_MNK	10^{-3}	102	7.06491e-2	106.9	39	5.88454e-2	45.6
	10^{-4}	178	5.96593e-2	187.0	68	5.85077e-2	74.1
	10^{-5}	254	5.85873e-2	264.9	97	5.84731e-2	102.2

The following observations immediately follow;

- For a given ϵ, K^* is about 35-95% less than K in Table 5.1 both for the RA and the RA_MNK schemes. Note that the extrapolations resulted in savings both in the number of iterations and in the CPU times for *all* values of ρ. The amount of savings increased with ρ; For the low utilization of Example 5.1.1 savings of approximately 35-40% were obtained, while savings by a factor of up to 25 were possible for the very high utilization of Example 5.1.5, both in RA and RA_MNK.

  More importantly, the performance measures obtained either from RA or RA_MNK without using the extrapolations were inaccurate for larger values of ϵ (10^{-3}, 10^{-4} and even 10^{-5}). On the other hand, the extrapolations resulted in much more accurate results in the performance measures for the same ϵ values. It is clearly seen from Table 5.1 that, generally EW^* were accurate up to n digits after the decimal, where $n = log(\epsilon)$. The same behavior can also be seen for the average queue lengths in the last column of Table 6.1. The value of ϵ used with the new stopping criterion thus provide direct information on the accuracy obtained in the calculation of the performance measures. On the other hand, the stopping criterion (5.8) gave no direct information on the accuracy desired in the calculation of the performance measures. Therefore, the use of the stopping criterion (5.10) also saves considerable amounts of CPU time by avoiding unnecessary iterations in order to obtain a desired accuracy.

- The algorithm RA_MNK required approximately 15-25% fewer number of iterations and 0-15% less CPU time than the RA, both with and without the extrapolations. Note that although the RA_MNK scheme was always faster than the RA, in some cases the corresponding CPU times were comparable due to the additional computations in the MNK step. The RA_MNK scheme also resulted in slightly more accurate performance measures.

FURTHER COMMENTS

Next, the effect of the choice of ϵ_γ is considered. In Table 5.2 ϵ_G is set to 10^{-5} and EW^* is displayed for different values of ϵ_γ for the Example 5.1.4 (resp. Example 5.1.5). The entries in the table are obtained from the RA (used with the proposed

extrapolations), however similar results are also observed for the RA_MNK. The probabilities $\gamma_n, 1 \leq n \leq N$, are normalized to sum to one. This normalization seems to improve the accuracy in the obtained performance measures especially for large ϵ_γ.

TABLE 5.2

Example 5.2.1 : $m = 5$, $\rho = 0.526953$ $\epsilon_G = 10^{-5}$.

ϵ_γ	K^*	N	EW^*
10^{-2}	69	5	0.013167
10^{-3}	70	6	0.012622
10^{-4}	70	8	0.012463
10^{-5}	70	9	0.012458
10^{-6}	70	10	0.012457
10^{-7}	70	11	0.012457

Example 5.2.2 : $m = 5$, $\rho = 0.8782556$ $\epsilon_G = 10^{-5}$.

ϵ_γ	K^*	N	EW^*
10^{-2}	73	7	0.516296
10^{-3}	72	8	0.515802
10^{-4}	72	10	0.515614
10^{-5}	72	11	0.515605
10^{-6}	72	13	0.515603
10^{-7}	72	14	0.515603

The figures in Table 5.2 indicate that, although the performance slightly degrades as ϵ_γ is increased, the "inaccuracy" in calculating the $\{\gamma_n\}_0^\infty$ seems to have a surprisingly small effect and up to 50% savings can be obtained in the computation time in cases when precision is not of primary concern. Also, comparison of the last two rows in both examples shows that savings of 20-25% can be obtained without compromising on the accuracy.

Considerable savings in computation time can also be gained in cases where the S matrix is sparse. However, the computation time per iteration will still approximately grow with m^3 since the matrices $H_k^{(n)}$ and G_k in equation (5.3) will not be sparse even when the matrix S is sparse.

Finally, the computation of the matrices $A^{(k)} := \frac{d^k A(z)}{d^k z}|_{z=1-}$, $k = 0, 1, \cdots$, used

in equation (4.8), is discussed. Equation (4.6) and Theorem 5.1 yield

$$\begin{aligned} A^{(k)} &= \int_0^\infty t^k\, e^{St}\, dH(t)\, \Lambda^k\ , \quad k = 0, 1, \cdots, \\ &= \sum_{n=0}^{\infty} \gamma_n^{(k)}\, K^n\, \Lambda^k\ , \end{aligned} \tag{5.11}$$

with

$$\gamma_n^{(k)} = \int_0^\infty t^k\ e^{-\theta t}\, \frac{(\theta t)^n}{n!}\, dH(t)\ , \quad n = 0, 1, \cdots, \quad \text{and} \quad \mathrm{k} = 0, 1, \cdots, \tag{5.12}$$

where θ and the matrix K are given in Theorem 5.1. It is plain that

$$\sum_{n=0}^{\infty} \gamma_n^{(k)} = \mu^{(k)}, \quad k = 0, 1, \cdots. \tag{5.13}$$

In view of equation (5.11), once $\gamma_n = \gamma_n^{(0)}$, $n = 0, 1, \cdots$, is calculated either by numerical integration or through the recursions given by equations (5.4) and (5.5), $\gamma_n^{(k)}$, $n = 0, 1, \cdots$, $k = 1, 2, \cdots$, can be obtained recursively by

$$\gamma_n^{(k)} = \frac{n+1}{\theta}\ \gamma_{n+1}^{(k-1)}\ , \quad n = 0, 1, \cdots \quad \text{and} \quad \mathrm{k} = 1, 2, \cdots, \tag{5.14}$$

and equation (5.13) can be used as a truncation criteria.

6. SPECIAL CASE: THE *MMPP/PH/1* QUEUE

In this section the service time distribution is specialized to be of *phase type*. The stationary queue length distribution at arbitrary times is obtained in *matrix-geometric* form, and simple expressions are given for the first two moments of this distribution. The stationary waiting time distribution for $GI/PH/1$ queue was derived in the work of Ramaswami and Lucantoni, [18, Thm. 1], and will not be repeated here.

Let the service distribution have *irreducible* phase representation (α, Q) of order l, where α is the $1 \times l$ row vector of initialization probabilities and Q is the $l \times l$ matrix of transition rates among the *transient* phases of the phase distribution.

Also, denote the $l \times 1$ column vector of absorption rates to the *absorbing* phase by Q^0.

A natural state space E for this system is given by

$$E = \begin{cases} (0,i) & \text{if } k = 0,\ 1 \le i \le m, \\ (k,i,j) & \text{if } k = 1,2,\cdots,\ 1 \le i \le m,\ 1 \le j \le l, \end{cases}$$

where k indicates the buffer size, while i and j represent the state of the $MMPP$ and the phase of the server, respectively. The service phase is not defined when the queue is empty. Since two events cannot occur simultaneously (with positive probability), it is an easy exercise to show that the underlying continuous-time Markov chain is *irreducible.*

The stationary queue length density at an arbitrary time is again denoted by the row vector y which is partitioned as $y := (y_0, y_1, \ldots)$, where y_0 is $1 \times m$ and y_k, $k = 1,2,\cdots$, are $1 \times lm$ vectors and correspond to the stationary probabilities of the states in lexicographical order. The corresponding infinitesimal generator matrix P then takes the form

$$P = \begin{pmatrix} S-\Lambda & \Lambda\otimes\alpha & 0 & 0 & 0 & 0 & \cdot \\ I_m\otimes Q^0 & A_1 & A_0 & 0 & 0 & 0 & \cdot \\ 0 & A_2 & A_1 & A_0 & 0 & 0 & \cdot \\ 0 & 0 & A_2 & A_1 & A_0 & 0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}, \tag{6.1}$$

where the $lm \times lm$ matrices A_k, $0 \le k \le 2$, are given by

$$A_0 = \Lambda \otimes I_l, \quad A_1 = (S-\Lambda) \oplus Q, \quad A_2 = I_m \otimes Q^0\alpha\ .$$

Here $\otimes$ and $\oplus$ denote the Kronecker product and the Kronecker sum, respectively, [1].

In order to obtain the stationary queue length distribution in matrix-geometric form, let the $lm \times lm$ matrix R be the minimal nonnegative solution to the matrix-quadratic equation

$$R^2 A_2 + R A_1 + A_0 = \mathbf{O}\ , \tag{6.2}$$

First, for a stable system, all the eigenvalues of R are shown to be inside the *open* unit disc.

Lemma 6.1 *The MMPP/PH/1 queueing system is stable if and only if* $sp(R) < 1$.

Proof: Since $\widetilde{P} := A_0 + A_1 + A_2 = S \oplus (Q + Q^0\alpha)$, the matrix $\widetilde{P}$ is irreducible owing to the irreducibility of the matrices S and $Q + Q^0\alpha$. Therefore $sp(R) < 1$ if and only if $pA_0e_{lm} < pA_2e_{lm}$, where the $1 \times lm$ row vector p is the stationary vector of $\widetilde{P}$ [14, p. 83]. From the structure of $\widetilde{P}$ it is easy to see that $p = \pi \otimes q$ where the $1 \times l$ row vector q is the stationary probability vector of the matrix $(Q + Q^0\alpha)$. Therefore, the statement of the Lemma follows since

$$p\,A_2\,e_{lm} = q\,Q^0 = effective\ service\ rate$$

and similarly

$$p\,A_0\,e_{lm} = \pi\,\lambda = effective\ arrival\ rate\,.$$

□

Lemma 6.2 *The matrix* $Z := \begin{pmatrix} S - \Lambda & \Lambda \otimes \alpha \\ I_m \otimes Q^0 & A_1 + RA_2 \end{pmatrix}$ *is an irreducible generator matrix.*

Proof: Post multiplication of Z by $e_{(lm+m)}$ after some simplifications yield

$$Z\,e_{(lm+m)} = \begin{pmatrix} 0_m^T \\ R(e_m \otimes Q^0) - \lambda \otimes e_l \end{pmatrix}\,.$$

But since

$$0_{lm}^T = (R^2A_2 + RA_1 + A_0)e_{lm} = (I_{lm} - R)[\lambda \otimes e_l - R(e_m \otimes Q^0)]$$

and $sp(R) < 1$, the equation $[\lambda \otimes e_l - R(e_m \otimes Q^0)] = 0_{lm}^T$ holds. Therefore, each row of the matrix Z sum to zero. On the other hand, by the definitions of the block entries of the matrix Z, its off-diagonal entries are all non-negative and the matrix Z is a generator matrix. Irreducibility of the Z matrix follows from the irreducibility

of the phase representation (α, Q) and of the matrix S, and from the non-negativity of the matrix R. □

The following *matrix-geometric* form of the stationary probability vector y now follows from Theorem 1.5.1. of [14, p. 25] and the normalization condition $\sum_{k=1}^{\infty} y_k e_{lm} + y_0 e_m = 1$.

Theorem 6.3 *Let z_0 and z_1 be $1 \times m$ and $1 \times lm$ row vectors, respectively, such that the vector $z := (z_0, z_1)$ is the unique positive solution to the equations*

$$z\,Z = 0_{(m+lm)} \ , \qquad z e_{(lm+m)} = 1. \tag{6.3}$$

Then, the stationary probability vector $y = (y_0, y_1, \ldots)$ has the following **matrix-geometric** *form*

$$\begin{aligned} y_0 &= c^{-1}\, z_0 \ , \\ y_1 &= c^{-1}\ z_1 \ , \\ y_k &= y_1\, R^{k-1} \ , \quad k = 1, 2, \cdots \ , \end{aligned} \tag{6.4}$$

with

$$c = z_0\, e_m + z_1\, (I_{lm} - R)^{-1}\, e_{lm} \ . \tag{6.5}$$

In this special case, the matrix-geometric nature of the solution yields the moments of the queue length distribution in a much simpler form. Although in principle, higher order moments can also be obtained, only the first two moments are given here.

Corollary 6.4 *The first two moments of the stationary queue length distribution at arbitrary times are given by*

$$E(Y) = y_1 (I_{lm} - R)^{-2}\, e_{lm} \ , \tag{6.6a}$$

$$E(Y^2) = y_1 (I_{lm} + R)\, (I_{lm} - R)^{-3}\, e_{lm} \ . \tag{6.6b}$$

Proof: The following equalities can be shown by direct computation;

$$\sum_{k=1}^{\infty} k R^{k-1}\, (I_{lm} - R)^2 = I_{lm} \tag{6.7a}$$

$$\sum_{k=1}^{\infty} k^2 R^{k-1}\, (I_{lm} - R)^3 = I_{lm} + R \tag{6.7b}$$

Equation (6.6a) now follows from equations (6.4), (6.7a) and Lemma 6.1 since

$$E(Y) = \sum_{k=1}^{\infty} k y_k e_{lm} = y_1 \sum_{k=1}^{\infty} k R^{k-1} e_{lm}$$

$$= y_1 (I_{lm} - R)^{-2} e_{lm} \ .$$

Similarly, the second moment follows from equations (6.4), (6.7b) and Lemma 6.1 . □

Note for the $M/M/1$ case that $R = \rho$ and (6.6) reduces to the well-known expressions

$$E(Y) = \frac{\rho}{1-\rho},$$

$$E(Y^2) = \frac{\rho}{(1-\rho)^2} + \frac{\rho^2}{(1-\rho)^2} \ .$$

NUMERICAL IMPLEMENTATION

The matrix R of equation (6.2) can be computed by using any of the algorithms summarized in the Appendix by replacing the upper limit in the summation for n by 2. The comments following equation (5.6) for the computation of the G matrix also apply to these algorithms. However, since the matrix R is *not stochastic*, the extrapolation (5.9) and the stopping criterion (5.10) cannot be employed to the R matrix calculations *in general.* Therefore, in order to obtain reasonable accuracy ϵ must be chosen sufficiently small and this may require huge number of iterations, especially for large utilizations.

However, in this quadratic case, the extension proposed in Section 5 can also be used in the computation of the R matrix, in view of the results obtained by Latouche [10]. In summary, Latouche's paper establishes several useful relationships between the matrices R and G for the quadratic case. In particular, Latouche shows that, if the matrices R and G are the minimal nonnegative solutions to the matrix quadratic equations

$$R = A_0 + RA_1 + R^2 A_2, \qquad \text{and} \qquad G = A_2 + A_1 G + A_0 G^2, \tag{6.8}$$

then

$$R = A_0(I - A_1 - A_0G)^{-1}, \quad \text{and} \quad G = (I - A_1 - RA_2)^{-1}A_2 \ . \qquad (6.9)$$

In view of this result, the matrix R can also be computed first by computing the G matrix with any of the algorithms (A.i)-(A.iv) of the Appendix together with the proposed extrapolation technique, and then obtaining R from the first equation in (6.9). Note for the $MMPP/PH/1$ queue that, the matrix R is of dimension ml.

An alternative approach is *not* to use the special structure of the service time distribution and solve an equation of the form (1.1) for a G matrix using the randomization algorithm given by equation (2.5). Although, the equation is of order N after truncation, the matrix G is only of dimension m. However, the analysis of the stationary probabilities for the $M/G/1$ type queues is much less transparent and the results cannot be put in as concise and explicit a form as in the matrix-geometric method [14]. The numerical discussion below aims to shed some light in providing guidelines in choosing the appropriate methodology in cases where both are applicable. Numerical results for the $M/G/1$ approach corresponds to the RA when used with the stopping criterion (5.10), called RA_{extr} hereafter.

The first-order computational requirements of the algorithms (see Appendix), *per iteration*, for the $MMPP/PH/1$ model are

$$\text{SS and MSS} : 3(lm)^3, \qquad \text{MNK} : 8(lm)^3, \qquad \text{RA} : 2Nm^3.$$

One would compare the computational complexities of the matrix-geometric and the $M/G/1$ approaches by comparing N to l^3. However, the results of Table 6.1 shows that even when l is small, the $M/G/1$ approach yields far less number of iterations and gives much more accurate results compared to any of the algorithms (A.i)-(A.iv). Furthermore, as before, the number of iterations in RA did not increase with the utilization ρ (although N slightly did), making the $M/G/1$ approach especially superior at higher utilizations. In Table 6.1, the notation X and X_{extr} denote the algorithm X when used with the stopping criterion (5.8) and (5.10), respectively.

The numbers in Table 6.1 for algorithms (A.i)-(A.iv) also compares these algorithms with each other as well as the effect of the proposed extrapolation method on these algorithms. For further examples and comparisons the reader is refered to

[5]. It is concluded in [5] that for small values of N, the MSS_{extr} is to be prefered over the MNK_{extr} due to the relative effect of the overhead in the MNK scheme. The comparison of the first order operation count per iteration given above and the number of iterations required for convergence in each scheme in Table 6.1 also supports this claim. Therefore, the $M/G/1$ approach (RA) is compared only to MSS_{extr} in the following discussion; Note that for $N \approx \frac{3}{2}l^3$, the RA_{extr} and the MSS_{extr} have approximately the same operation count per iteration. In Examples 6.1.2-6.1.4, even when $l = 2$, MSS_{extr} and RA_{extr} required about the same CPU time per iteration. However, the number of iterations in RA_{extr} were far less than the number of iterations in MSS_{extr}. RA_{extr} gave the same accuracy in the average queue length, $E(QL)$, with savings by a factor of approximately 25 (resp. 5) in Examples 6.1.1-6.1.3 (resp. in Example 6.1.4). In Example 6.1.5, although RA_{extr} required about 3.5 times more CPU time per iteration than MSS_{extr}, it resulted in savings in the number of iterations by a factor of 20, thus yielding savings by a factor of approximately 6 in CPU time. In Examples 6.1.7 and 6.1.8 since $l = 1$, the MSS_{extr} required much less CPU time (approximately by a factor of 20), but even this is offset by the huge difference in the number of iterations; a factor of 20 and 140 in Example 6.1.7 and Example 6.1.8, respectively. On the other hand, in Example 6.1.6 where $l = 4$, the MSS_{extr} required about 3 times more CPU time per iteration. Note also in this example that one gets the same accuracy in RA_{extr} in about 20 times fewer number of iterations, i. e., savings by a factor of 60 in CPU time. Potential savings that can be obtained by choosing RA_{extr} over MSS_{extr} for larger values of l is no further elaborated but left to the readers imagination!

TABLE 6.1

(i) SS_{extr}, (ii) MSS, (iii) MSS_{extr}, (iv) MNK, (v) MNK_{extr}, (vi) RA_{extr}.

		# of iter.						$E(QL)$					
	$-\log_{10}(\epsilon)$	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(i)	(ii)	(iii)	(iv)	(v)	(vi)
Example 6.1.1.													
m=2, l=2,	3	49	86	15	77	13	10	40.096	22.972	29.028	23.583	29.162	26.797
N=18, ρ=0.768.	5	161	315	120	263	112	17	26.905	26.746	26.842	26.756	26.835	26.797
	6	423	437	229	362	203	20	26.815	26.792	26.803	26.793	26.802	26.797
	7	853	559	349	460	300	20	26.799	26.797	26.798	26.797	26.798	26.797
Example 6.2.2.													
m=2, l=2,	5	163	420	77	378	74	26	93.458	80.952	91.723	82.060	91.585	89.395
N=13, ρ=0.987.	7	655	1220	371	1059	349	40	89.582	89.293	89.475	89.308	89.466	89.395
	8	1431	1636	709	1413	649	47	89.430	89.385	89.405	89.386	89.403	89.395
Example 6.1.3.													
m=2, l=2,	5	100	85	50	51	47	9	6.5371	6.4897	6.5099	6.4915	6.5085	6.5000
N=13, ρ=0.875.	7	395	149	113	131	102	11	6.5006	6.4999	6.5001	6.4999	6.5001	6.5000
	9	789	213	177	186	157	14	6.5000	6.5000	6.5000	6.5000	6.5000	6.5000
Example 6.1.4.													
m=2, l=2,	3	50	24	14	22	13	9	3.2676	3.1497	3.2611	3.1584	3.2620	3.2126
N=11, ρ=0.767.	5	132	57	44	51	40	18	3.2122	3.2110	3.2122	3.2111	3.2122	3.2116
	7	321	91	77	80	69	28	3.2117	3.2116	3.2116	3.2116	3.2117	3.2116
Example 6.1.5.													
m=2, l=2,	5	114	93	54	83	51	9	7.2496	7.1974	7.2187	7.1988	7.2172	7.2083
N=41, ρ=0.880.	7	466	163	122	144	111	15	7.2091	7.2082	7.2084	7.2082	7.2084	7.2083
	9	961	232	192	204	171	22	7.2083	7.2083	7.2083	7.2083	7.2083	7.2083
Example 6.1.6.													
m=2, l=4,	5	110	93	53	83	50	9	6.9253	6.8748	6.8963	6.8764	6.8948	6.8856
N=35, ρ=0.888.	7	440	163	122	143	110	16	6.8855	6.8855	6.8857	6.8855	6.8857	6.8856
	9	668	232	192	173	140	22	6.8857	6.8856	6.8856	6.8856	6.8856	6.8856
Example 6.1.7.													
m=2, l=1,	5	73	71	47	53	39	5	5.0092	4.9962	5.0038	4.9972	5.0027	5.0000
N=28, ρ=0.833.	7	186	119	95	88	73	6	5.0001	5.0000	5.0000	5.0000	5.0000	5.0000
Example 6.1.8.													
m=2, l=1,	5	78	259	59	207	54	5	51.742	46.433	51.224	47.354	51.079	50.000
N=32, ρ=0.980.	7	381	707	271	530	234	6	50.068	49.959	50.036	49.991	50.028	50.000
	9	1313	1169	707	860	554	8	50.001	50.000	50.004	50.000	50.000	50.000

The above discussion of the results in Table 6.1 indicates that even when the number of service phases is small there is computational advantage in ignoring the special structure of the service time distribution and using the $M/G/1$ approach. It should be reminded however that, as already indicated above, both the waiting time and the queue length distribution calculations are much simpler in the matrix-geometric case once the R matrix is obtained. In the $M/G/1$ approach, as given here in Section 3 for the $MMPP/G/1$ queue, these distributions are typically given in terms of the transform equations, whence Laplace and Z transform inversions are needed. Since the transform inversion methods require considerable CPU times, especially when accuracy is of primary concern, the decisions must be based on the number of distribution points and the degree of accuracy desired in these distributions. However, for the $MMPP/G/1$ queue, Lemmas 3.1 and 4.1 gives the moments of the queue length and the waiting time distributions *recursively* in a fraction of the time required to compute the G matrix without having to compute the inverse transforms and the matrices $\{A_n\}_0^\infty$. Also, an *alternative* approach in which an R matrix that satisfies equation (6.4) but not (6.2) is obtained explicitly in terms of the system parameters is presented in [4] for the $MMPP/PH/1/K$ queue.

ACKNOWLEDGMENTS

I wish to express my gratitude to Dr. Harry Heffes for his valuable assistance and support during this research. I also thank Dr. Bharat Doshi for his involvement and interest especially on the early stages of this research. Special thanks are also due to Dr. David M. Lucantoni for many helpful comments and suggestions. Finally, I wish to express my heartfelt appreciation of the efforts of my advisor Prof. Armand M. Makowski who carefully read through several drafts of this manuscript and made significant improvements through his comments.

REFERENCES

[1] J.W. Brewer, " Kronecker products and matrix calculus in systems theory," *IEEE Trans. on Circuits and Systems*, **25**, pp. 772-781, 1978.

[2] E. Çınlar, *Introduction to Stochastic Processes.* Prentice Hall, 1975.

[3] W. Grassman, "Transient solutions in Markovian queues," *Europian Journal of Operations Research,* **1**, pp. 396-402, 1977.

[4] L. Gün, "Closed-form matrix-geometric solution for a class of quasi-birth-and-death processes", Proceedings of the 21^{st} it Annual Conference on Information Sciences and Systems, John Hopkins University, Baltimore, MD., March 1987.

[5] L. Gün, " Experimental results on matrix-analytical solution techniques – extensions and comparisons," submitted to *Stochastic Models*, 1988.

[6] H. Heffes, " A class of data traffic processes - Covariance function characterization and related queueing results," *B.S.T.J.*, **59**, pp. 897-929, 1980.

[7] H. Heffes, and D. M. Lucantoni, " A Markov modulated characterization of packetized voice and related statistical multiplexer performance," *IEEE J. on Selected Areas in Communication - Special issue on Network Performance Evaluation,* SAC-4, **6**, pp. 856-868, 1986.

[8] S. Karlin, and H. M. Taylor, *A First Course in Stochastic Processes.* Academic Press, 1975.

[9] J. Kemeny, and J. L. Snell, *Finite Markov Chains.* Van Nostrand Publishing Co., 1960.

[10] G. Latouche, " A note on two matrices occurring in the solution of quasi-birth-and-death processes," *Stochastic Models*, **3**, pp. 125-136, 1987.

[11] D. M. Lucantoni and V. Ramaswami, " Efficient algorithms for solving the non-linear matrix equations arising in phase type queues," *Stochastic Models,* **1**, pp. 29-51, 1985.

[12] M. F. Neuts, "The Markov renewal branching process," *Proc. Conf. on Math. Methods in the Theory of Queues*, pp. 1-21, Kalamazo, MI, 1974.

[13] M. F. Neuts, " Moment formulas for the Markov renewal branching process," *Adv. Appl. Prob.*, **8**, pp. 690-711, 1976.

[14] M. F. Neuts, *Matrix-geometric Solutions in Stochastic Models.* The John Hopkins University Press, 1981.

[15] M. F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and Their Applications,* in preparation.

[16] J. Ortega and W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[17] V. Ramaswami, " The $N/G/1$ queue and its detailed analysis," *Adv. Appl. Prob.,* **12**, pp. 222-261, 1980.

[18] V. Ramaswami and D. M. Lucantoni, " Stationary waiting time distribution in queues with phase type service and in quasi-birth-death processes," *Stochastic Models*, **1**, pp. 125-136, 1985.

[19] V. Ramaswami, " Nonlinear matrix equations in applied probability - solution techniques and open problems," to appear in *SIAM Review*, 1988.

APPENDIX
REVIEW OF THE ITERATIVE ALGORITHMS

In this Appendix, iterative algorithms available for finding the matrices G and R in equation (1.1) are briefly surveyed. Although the discussion is given here for the G matrix, these algorithms apply *mutatis mutandis* to the calculation of the R matrix. It is assumed that $\rho < 1$, or equivalently, that the matrix G is stochastic.

(A.i.) Successive Substitutions (SS): It is shown in [12] that the sequence of matrices $\{G_k\}_0^\infty$ obtained by

$$G_0 = \mathbf{O} \ , \qquad G_{k+1} = \sum_{n=0}^{\infty} A_n G_k^n \ , \qquad k \geq 0, \tag{A.1}$$

is non-decreasing and converges to the G matrix. The convergence rate of this direct iterative scheme is *R-linear*, and can be extremely slow when ρ is close to one.

(A.ii.) Modified Successive Substitutions (MSS): In cases where the matrix $(I - A_1)$ is well conditioned, there is computational advantage in using the iteration

$$G_0 = \mathbf{O} \ , \qquad G_{k+1} = (I - A_1)^{-1} \left(A_0 + \sum_{n=2}^{\infty} A_n G_k^n \right) \ , \qquad k \geq 0. \tag{A.2}$$

Note that, since the matrix A_1 is substochastic, $(I - A_1)^{-1}$ exists. It can easily be shown that the sequence of matrices $\{G_k\}_0^\infty$ obtained from MSS is componentwise larger then the corresponding matrices obtained from SS. Therefore, although MSS requires *a* matrix inversion it converges in fewer number of iterations. It is reported in [19] that MSS results in about 20% savings compared to SS.

(A.iii.) Newton-Kantorovich Method (NK): The Gateaux derivative [16] of the mapping $F(X) = X - \sum_{n=0}^{\infty} A_n X^n$ at X is given by the linear map

$$[F'(X)] : U \longmapsto U - \sum_{n=1}^{\infty} \sum_{l=0}^{n-1} A_n X^l U X^{n-1-l} \ .$$

It is shown in [13] that $F'(X)$ is a non-negative matrix. Furthermore, for $\mathbf{O} \leq X \leq G$ (componentwise), $F'(X)$ is an *isotone* so that the operator F is *order-convex*. It then follows by the Monotone Convergence Theorem [16, p. 45] that the Newton-Kantorovich scheme for (1.1) given by

$$G_0 = \mathbf{O} \ , \qquad G_{k+1} = G_k - [F'(G_k)]^{-1} F(G_k), \quad k \geq 0,$$

converges monotonically to G. The iterates $\{G_k\}_0^\infty$ can equivalently be obtained by the iteration [13, 19]

$$G_0 = \mathbf{O} \ , \qquad G_{k+1} = G_k + Y_k, \quad k \geq 0, \tag{A.3.a}$$

where Y_k is the unique solution to the linear system

$$Y_k = -F(G_k) + \sum_{n=0}^{\infty} \sum_{l=0}^{n-1} A_n G_k^l Y_k G_k^{n-1-l} \quad k \geq 0\,. \tag{A.3.b}$$

The numerical results in Neuts [13] indicate that despite the faster rate of convergence, the NK scheme is more time consuming than the first two schemes, as the system in (A.3.b) needs to be solved at every iteration. Therefore, this scheme is not considered in this paper.

(A.iv.) Modified Newton-Kantorovich Scheme (MNK): The following modification is proposed by Ramaswami [19] for the NK scheme in order to avoid the solution of the linear system in (A.3.b).

$$G_0 = \mathbf{O}\,, \qquad G_{k+1} = G_k + Y_k, \tag{A.4.a}$$

where

$$Y_k = -F(G_k) + A_1 Z_k + A_2(Z_k G_k + G_k Z_k)\,, \tag{A.4.b}$$

with

$$Z_k = -(I - A_1)^{-1} F(G_k) \qquad k \geq 0\,. \tag{A.4.c}$$

The algorithm (A.4) corresponds to truncating the summation in (A.3.b) at $n = 2$ and using the *estimate* Z_k of Y_k instead of solving the linear system. It is shown in [19] that the MNK scheme also converges monotonically to G.