INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

ETAQA Solutions for Infinite Markov Processes with Repetitive Structure

Alma Riska, Evgenia Smirni,

To cite this article:
Alma Riska, Evgenia Smirni, (2007) ETAQA Solutions for Infinite Markov Processes with Repetitive Structure. INFORMS Journal on Computing 19(2):215-228. https://doi.org/10.1287/ijoc.1050.0160

INFORMS Journal on Computing
Vol. 19, No. 2, Spring 2007, pp. 215–228
ISSN 1091-9856 | EISSN 1526-5528 | 07 | 1902 | 0215

DOI 10.1287/ijoc.1050.0160

ETAQA Solutions for Infinite Markov Processes with Repetitive Structure

Alma Riska
Seagate Research, 1251 Waterfront Place, Pittsburgh, Pennsylvania 15222, USA, alma.riska@seagate.com

Evgenia Smirni
Department of Computer Science, College of William and Mary, P.O. Box 8795,
Williamsburg, Virginia 23187-8795, USA, esmirni@cs.wm.edu

We describe the ETAQA (efficient technique for the solution of quasi birth-death processes) approach for the exact analysis of M/G/1 and GI/M/1-type processes, and their intersection, i.e., quasi birth-death processes. ETAQA exploits the repetitive structure of the infinite portion of the chain and derives a finite system of linear equations. In contrast to the classic techniques for solution of such systems, the solution of this finite linear system *does not* provide the entire probability distribution of the state space, but simply allows calculation of the aggregate probability of a finite set of classes of states from the state space, appropriately defined. Nonetheless, these aggregate probabilities allow for computation of a rich set of measures of interest such as the system queue length or any of its higher moments. The proposed solution approach is exact and, for the case of M/G/1-type processes, compares favorably to the classic methods as shown by detailed time and space complexity analysis. Detailed experimentation further corroborates that ETAQA provides significantly less expensive solutions when compared to the classic methods.

Key words: M/G/1-type processes; GI/M/1-type processes; quasi birth-death processes; computer system performance modeling; matrix-analytic methods; Markov chains
History: Accepted by Edward P. C. Kao, Area Editor for Computational Probability and Analysis; received December 2003; revised November 2004, accepted July 2005.

1. Introduction

Matrix-analytic techniques, pioneered by Neuts (1981, 1989) provide a framework that is widely used for exact analysis of a general and frequently encountered class of queueing models. In these models, the embedded Markov chains are two-dimensional generalizations of elementary GI/M/1 and M/G/1 queues (Kleinrock 1975) and their intersection, i.e., quasi birth-death (QBD) processes. GI/M/1 and M/G/1 queues model systems with interarrival and service times characterized by *general* distributions rather than simple exponentials and are often used as the modeling tool of choice in modern computer and communication systems (Nelson 1995; Ramaswami and Wang 1996; Squillante 1998, 2000). As a consequence, various analytic methodologies for their solution have been developed by Neuts (1989), Latouche (1993), Latouche and Stewart (1995), Meini (1998), and Grassman and Stanford (2000).

In this paper, we present ETAQA, an analytic solution technique for the exact analysis of M/G/1-type, GI/M/1-type Markov chains, and their intersection, i.e., quasi birth-death processes. Neuts (1981) defines various classes of infinite-state Markov chains with a repetitive structure. In all cases, the state space $\mathcal{S}$ is partitioned into the boundary states $\mathcal{S}^{(0)} = \{s_1^{(0)}, \ldots, s_m^{(0)}\}$ and the sets of states representing each repetitive level $\mathcal{S}^{(j)} = \{s_1^{(j)}, \ldots, s_n^{(j)}\}$, for $j \geq 1$. For the M/G/1-type Markov chains, the infinitesimal generator $\mathbf{Q}_{M/G/1}$ has upper block Hessenberg form and Neuts (1989) proposes matrix-analytic methods for their solution. The key in the matrix-analytic solution is computation of an auxiliary matrix $\mathbf{G}$. Similarly, for Markov chains of the GI/M/1-type, the infinitesimal generator has a lower block Hessenberg form, and Neuts (1981) proposes the very elegant matrix-geometric solution. QBD processes with a block tridiagonal infinitesimal generator can be solved using either methodology, but matrix geometric is the preferred one (see Latouche and Ramaswami 1999).

The traditional matrix-analytic algorithms were developed based on the concept of stochastic complementation, as explained in Riska and Smirni (2002b) and provide a recursive function for computation of the probability vector $\boldsymbol{\pi}^{(i)}$ that corresponds to $\mathcal{S}^{(j)}$, for $j \geq 1$. This recursive function is based on $\mathbf{G}$ (for the case of M/G/1-type processes) or $\mathbf{R}$ (for the case of GI/M/1-type processes). Iterative procedures are used for determining $\mathbf{G}$ or $\mathbf{R}$ (see details in Latouche 1993 and Meini 1998). For more details on stochastic

complementation and its application for the development of matrix-analytic algorithms, see Riska and Smirni (2002b).

ETAQA (which stands for the efficient technique for the analysis of QBD processes by aggregation) was first introduced in Ciardo and Smirni (1999) for solution of a *limited* class of QBD processes. This limited class allowed the return from level $\mathcal{S}^{(j+1)}$ to level $\mathcal{S}^{(j)}$, $j \geq 1$, to be directed toward a single state only. This same result was extended in Ciardo et al. (2004) for solution of M/G/1-type processes with the same restriction, i.e., returns from any higher level $\mathcal{S}^{(j+1)}$ in the Markov chain to its lower level $\mathcal{S}^{(j)}$ have to be directed to a single state only.

In this paper, we adapt the ETAQA approach for solution of *general* processes of the M/G/1-type, GI/M/1-type, as well as QBDs, i.e., we relax the above strong assumption of returns to a single state only, and provide a general solution approach that works for *any* type of returns to the lower level, i.e., transitions from any state in level $\mathcal{S}^{(j+1)}$ to any state in level $\mathcal{S}^{(j)}$, $j \geq 1$ are allowed. In contrast to the matrix-analytic techniques for solving M/G/1-type and GI/M/1-type processes that use a recursive function for computation of the probability vectors of each level, ETAQA uses a different treatment: It constructs and solves a finite linear system of $m+2n$ unknowns, where m is the number of states in the boundary portion of the process and n is the number of states in each of the repetitive "levels" of the state space, and obtain an exact solution. Instead of evaluating the stationary probability distribution of *all* states in each of the repetitive levels $\mathcal{S}^{(j)}$ of the state space $\mathcal{S}$, we calculate the *aggregate* stationary probability distribution of n classes of states $\mathcal{T}^{(i)}$, $1 \leq i \leq n$, appropriately defined (see Figure 1). This approach could be perceived as similar to lumpability since an aggregate probability distribution is computed, or perhaps also stochastic complementation. We stress that the finite system of $m+2n$ linear equations that ETAQA provides is not an infinitesimal generator, so aggregation of the infinite set $\mathcal{S}$ into a finite number of classes of states does not result in a Markov chain, so it cannot be considered similar to any lumpability of stochastic complementation techniques.

Yet, computation of the aggregate probability distribution that we compute with our method is *exact*. Furthermore, this aggregate probability distribution provides the means for calculating a variety of measures of interest, including the expected queue length and any of its higher moments. Although ETAQA does not allow for exact calculation of the queue-length distribution, it provides the means to compute the coefficient of variation (via the second moment) as well as the skewness of the distribution (via the third moment), which in turn provide further information about the queueing behavior of the system.

ETAQA results in solutions that are significantly more efficient than those from traditional methods for M/G/1-type processes. For QBD and GI/M/1-type processes, ETAQA results in solutions that are as efficient as the classic ones. We provide detailed big-O complexity analysis of ETAQA and the most efficient alternative methods. These results are further corroborated via detailed experimentation.

An additional important issue is related to the numerical stability of the method, especially for M/G/1-type processes. Riska and Smirni (2002a), a preliminary version of this paper that focused on M/G/1-type processes only, provides experimental indications that the method is numerically stable. Here, we do not focus on numerical stability, but we instead illustrate that the method generalizes to the solution of M/G/1-type, GI/M/1-type, and QBD processes of any type. Numerical stability of ETAQA and its connection to matrix-analytic methods is explored formally in Stathopoulos et al. (2005), where ETAQA's numerical stability is proved and shown to be often superior to the alternative matrix-analytic solutions.

In Section 2 we outline the matrix-analytic methods for the solution of M/G/1-type, GI/M/1-type, and QBD processes. ETAQA, along with detailed time and storage complexity analysis for the solution of M/G/1-type, GI/M/1-type, and QBD processes is presented in Sections 3, 4, and 5, respectively. We experimentally compare its efficiency with the best

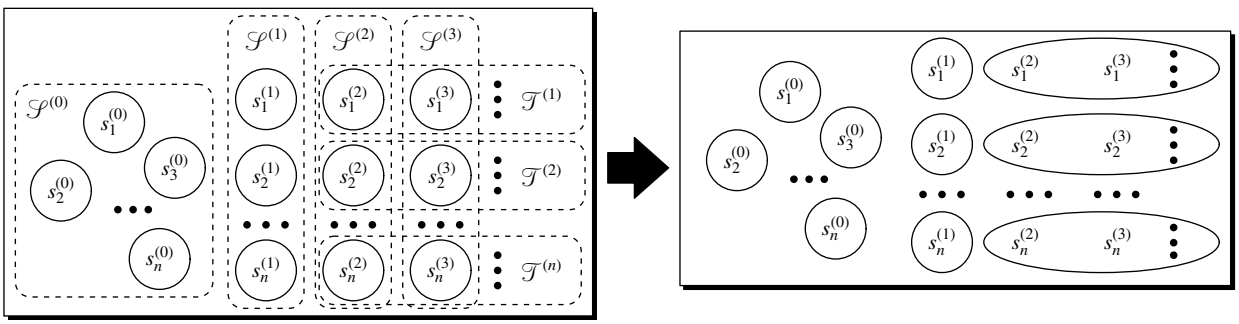

Figure 1 **Aggregation of an Infinite $\mathcal{S}$ into a Finite Number of Classes of States**

known methods in a set of realistic examples (see Section 6) for the case of M/G/1-type processes. Finally, we summarize our findings and report on ETAQA's efficiency in Section 7.

2. Background

We assume continuous time Markov chains, or CTMCs, so we refer to the infinitesimal generator $\mathbf{Q}$, but our discussion applies just as well to discrete-time Markov chains, or DTMCs. Neuts (1981) defines various classes of infinite-state Markov chains with a repetitive structure. In all cases, the state space $\mathcal{S}$ is partitioned into the boundary states $\mathcal{S}^{(0)} = \{s_1^{(0)}, \ldots, s_m^{(0)}\}$ and the sets of states $\mathcal{S}^{(j)} = \{s_1^{(j)}, \ldots, s_n^{(j)}\}$, for $j \geq 1$, while $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{\pi}^{(j)}$ are the stationary probability vectors for states in $\mathcal{S}^{(0)}$ and $\mathcal{S}^{(j)}$, for $j \geq 1$.

2.1. M/G/1-Type Processes

For the class of M/G/1-type Markov chains, the infinitesimal generator $\mathbf{Q}_{M/G/1}$ is block-partitioned as

$$\mathbf{Q}_{M/G/1} = \begin{bmatrix} \hat{\mathbf{L}} & \hat{\mathbf{F}}^{(1)} & \hat{\mathbf{F}}^{(2)} & \hat{\mathbf{F}}^{(3)} & \hat{\mathbf{F}}^{(4)} & \cdots \\ \hat{\mathbf{B}} & \mathbf{L} & \mathbf{F}^{(1)} & \mathbf{F}^{(2)} & \mathbf{F}^{(3)} & \cdots \\ \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F}^{(1)} & \mathbf{F}^{(2)} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F}^{(1)} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \quad (1)$$

"L," "F," and "B" describe "local," "forward," and "backward" transition rates, respectively, in relation to a set of states $\mathcal{S}^{(j)}$ for $j \geq 1$, and a "ˆ" indicates matrices related to $\mathcal{S}^{(0)}$.

For solution of M/G/1-type processes, several algorithms exist in Grassman and Stanford (2000), Bini et al. (2000), Meini (1998), and Neuts (1989). These algorithms first compute $\mathbf{G}$ as the solution of the matrix equation

$$\mathbf{B} + \mathbf{LG} + \sum_{j=1}^{\infty} \mathbf{F}^{(j)} \mathbf{G}^{j+1} = \mathbf{0}. \quad (2)$$

The matrix $\mathbf{G}$, which is stochastic if the process is recurrent and irreducible, has an important probabilistic interpretation: Entry (k, l) in $\mathbf{G}$ is the conditional probability of the process first entering $\mathcal{S}^{(j-1)}$ through state l, given that it starts from state k of $\mathcal{S}^{(j)}$, as defined in Neuts (1989, p. 81). The probabilistic interpretation of $\mathbf{G}$ is the same for both DTMCs and CTMCs. The interpretation in Neuts (1989, p. 81) is consistent with the discussion in Latouche and Ramaswami (1999, p. 142), where CTMCs are taken into consideration. $\mathbf{G}$ is obtained by solving (2) iteratively. However, recent advances show that computing $\mathbf{G}$ is more efficient when displacement structures are used based on representation of M/G/1-type processes by QBD processes, as discussed in Meini (1998), Bini et al. (2000), Bini and Meini (1998), and Latouche and Ramaswami (1999). The most efficient algorithm to compute $\mathbf{G}$ is the cyclic reduction in Bini et al. (2000).

Calculation of the stationary probability vector is based on Ramaswami's (1988) recursive formula, which is numerically stable because it entails only additions and multiplications. Neuts (1989) and Ramaswami (1988) suggest that subtractions on such formulas could be numerically unstable. Ramaswami's formula defines the following recursion among stationary probability vectors $\boldsymbol{\pi}^{(j)}$ for $j \geq 0$:

$$\boldsymbol{\pi}^{(j)} = -\left(\boldsymbol{\pi}^{(0)} \hat{\mathbf{S}}^{(j)} + \sum_{k=1}^{j-1} \boldsymbol{\pi}^{(k)} \mathbf{S}^{(j-k)}\right) {\mathbf{S}^{(0)}}^{-1} \quad \forall\, j \geq 1, \quad (3)$$

where $\hat{\mathbf{S}}^{(j)}$ and $\mathbf{S}^{(j)}$ are defined as

$$\hat{\mathbf{S}}^{(j)} = \sum_{l=j}^{\infty} \hat{\mathbf{F}}^{(l)} \mathbf{G}^{l-j}, \quad j \geq 1,$$

$$\mathbf{S}^{(j)} = \sum_{l=j}^{\infty} \mathbf{F}^{(l)} \mathbf{G}^{l-j}, \quad j \geq 0 \text{ (letting } \mathbf{F}^{(0)} \equiv \mathbf{L}). \quad (4)$$

Given the above definition of $\boldsymbol{\pi}^{(j)}$ and the normalization condition, a unique vector $\boldsymbol{\pi}^{(0)}$ can be obtained by solving the following system of m linear equations:

$$\boldsymbol{\pi}^{(0)} \left[\left(\hat{\mathbf{L}} - \hat{\mathbf{S}}^{(1)} {\mathbf{S}^{(0)}}^{-1} \hat{\mathbf{B}}\right)^{\diamond} \,|\, \mathbf{1}^T - \left(\sum_{j=1}^{\infty} \hat{\mathbf{S}}^{(j)}\right) \left(\sum_{j=0}^{\infty} \mathbf{S}^{(j)}\right)^{-1} \mathbf{1}^T \right] = [\mathbf{0} \,|\, 1], \quad (5)$$

where "$\diamond$" indicates that we discard one (any) column of the corresponding matrix, since we added a column representing the normalization condition. Once $\boldsymbol{\pi}^{(0)}$ is known, we can then iteratively compute $\boldsymbol{\pi}^{(j)}$ for $j \geq 1$, stopping when the accumulated probability mass is close to one. After this point, measures of interest can be computed. Since the relation between $\boldsymbol{\pi}^{(j)}$ for $j \geq 1$ is not straightforward, computing measures of interest requires generation of the whole stationary probability vector. For a limited set of measures of interest such as first and second moments of queue length, Lucantoni (1983) proposes closed-form (but complex) formulas that do not require knowledge of the entire vector $\boldsymbol{\pi}$.

Meini (1997a) gives an improved version of Ramaswami's formula. Once $\boldsymbol{\pi}^{(0)}$ is known using (5), the stationary probability vector is computed using matrix-generating functions associated with triangular Toeplitz matrices. A Toeplitz matrix has equal elements in each of its diagonals, which makes them

easier to handle than general matrices. These matrix-generating functions are computed efficiently using fast Fourier transforms (FFTs):

$$\begin{aligned}\tilde{\boldsymbol{\pi}}^{(1)} &= -\mathbf{b}\mathbf{Y}^{-1}\\ \tilde{\boldsymbol{\pi}}^{(i)} &= -\tilde{\boldsymbol{\pi}}^{(i-1)}\mathbf{Z}\mathbf{Y}^{-1} \quad i \geq 2,\end{aligned} \tag{6}$$

where $\tilde{\boldsymbol{\pi}}^{(1)} = [\boldsymbol{\pi}^{(1)}, \ldots, \boldsymbol{\pi}^{(p)}]$ and $\tilde{\boldsymbol{\pi}}^{(i)} = [\boldsymbol{\pi}^{(p(i-1)+1)}, \ldots, \boldsymbol{\pi}^{(pi)}]$ for $i \geq 2$. Matrices $\mathbf{Y}$, $\mathbf{Z}$, and $\mathbf{b}$ are

$$\mathbf{Y} = \begin{bmatrix} \mathbf{S}^{(0)} & \mathbf{S}^{(1)} & \mathbf{S}^{(2)} & \cdots & \mathbf{S}^{(p-1)} \\ \mathbf{0} & \mathbf{S}^{(0)} & \mathbf{S}^{(1)} & \cdots & \mathbf{S}^{(p-2)} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}^{(0)} & \cdots & \mathbf{S}^{(p-3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{S}^{(0)} \end{bmatrix},$$

$$\mathbf{Z} = \begin{bmatrix} \mathbf{S}^{(p)} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{S}^{(p-1)} & \mathbf{S}^{(p)} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{S}^{(2)} & \mathbf{S}^{(3)} & \cdots & \mathbf{S}^{(p)} & \mathbf{0} \\ \mathbf{S}^{(1)} & \mathbf{S}^{(2)} & \cdots & \mathbf{S}^{(p-1)} & \mathbf{S}^{(p)} \end{bmatrix},$$

$$\mathbf{b} = \boldsymbol{\pi}^{(0)} \begin{bmatrix} \hat{\mathbf{S}}^{(1)} \\ \hat{\mathbf{S}}^{(2)} \\ \hat{\mathbf{S}}^{(3)} \\ \vdots \\ \hat{\mathbf{S}}^{(p)} \end{bmatrix}^{T},$$

where p is a constant that defines how many of matrices $\mathbf{S}^{(i)}$ and $\hat{\mathbf{S}}^{(i)}$ are computed. In the above representation, the matrix $\mathbf{Y}$ is an upper block-triangular Toeplitz matrix and the matrix $\mathbf{Z}$ is a lower block-triangular Toeplitz matrix.

2.2. GI/M/1-Type Processes

For GI/M/1-type Markov chains, the infinitesimal generator $\mathbf{Q}_{GI/M/1}$ is block-partitioned as

$$\mathbf{Q}_{GI/M/1} = \begin{bmatrix} \hat{\mathbf{L}} & \hat{\mathbf{F}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \hat{\mathbf{B}}^{(1)} & \mathbf{L} & \mathbf{F} & \mathbf{0} & \mathbf{0} & \cdots \\ \hat{\mathbf{B}}^{(2)} & \mathbf{B}^{(1)} & \mathbf{L} & \mathbf{F} & \mathbf{0} & \cdots \\ \hat{\mathbf{B}}^{(3)} & \mathbf{B}^{(2)} & \mathbf{B}^{(1)} & \mathbf{L} & \mathbf{F} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{7}$$

Key to the general solution of the generator in (7) is the fact that the following geometric relation holds among the stationary probability vectors $\boldsymbol{\pi}^{(j)}$ and $\boldsymbol{\pi}^{(1)}$ for states in $\mathcal{S}^{(j)}$:

$$\boldsymbol{\pi}^{(j)} = \boldsymbol{\pi}^{(1)}\mathbf{R}^{j-1}, \quad \forall\, j \geq 1, \tag{8}$$

where $\mathbf{R}$ is the solution of the matrix equation

$$\mathbf{F} + \mathbf{R}\mathbf{L} + \sum_{k=1}^{\infty} \mathbf{R}^{k+1}\mathbf{B}^{(k)} = \mathbf{0}, \tag{9}$$

and can be computed using iterative numerical algorithms. Matrix $\mathbf{R}$, the geometric coefficient, has an important probabilistic interpretation: Entry (k, l) of $\mathbf{R}$ is the expected time spent in state l of $\mathcal{S}^{(i)}$, before the first visit into $\mathcal{S}^{(i-1)}$, expressed in time units Δ^i, given the starting state is k in $\mathcal{S}^{(i-1)}$. Δ^i is the mean sojourn time in state k of $\mathcal{S}^{(i-1)}$ for $i \geq 2$, as defined in Neuts (1981, pp. 30–35). Latouche (1993) describes several iterative numerical algorithms for computation of $\mathbf{R}$. The normalization condition and (9) are then used to obtain $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{\pi}^{(1)}$ by solving the following system of $m+n$ equations:

$$\begin{aligned}&[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}]\\ &\cdot\left[\begin{array}{c|c|c} \hat{\mathbf{L}}^{\diamond} & \hat{\mathbf{F}} & \mathbf{1}^T \\ (\sum_{k=1}^{\infty}\mathbf{R}^{k-1}\hat{\mathbf{B}}^{(k)})^{\diamond} & \mathbf{L}+\sum_{k=1}^{\infty}\mathbf{R}^{k}\mathbf{B}^{(k)} & (\mathbf{I}-\mathbf{R})^{-1}\mathbf{1}^T \end{array}\right]\\ &= [\mathbf{0}\,|\,1].\end{aligned} \tag{10}$$

For $k \geq 1$, $\boldsymbol{\pi}^{(k)}$ can be obtained numerically from (8). More importantly, useful performance metrics are computed exactly in explicit form. For example, average queue length is computed using $\boldsymbol{\pi}^{(1)}(\mathbf{I}-\mathbf{R})^{-2}\mathbf{1}^T$.

2.3. Quasi Birth-Death Processes

Intersection of GI/M/1-type and M/G/1-type processes is the special case of the quasi birth-death (QBD) processes, whose infinitesimal generator $\mathbf{Q}_{\mathrm{QBD}}$ is of the block tri-diagonal form

$$\mathbf{Q}_{\mathrm{QBD}} = \begin{bmatrix} \hat{\mathbf{L}} & \hat{\mathbf{F}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \hat{\mathbf{B}} & \mathbf{L} & \mathbf{F} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{11}$$

While the QBD case falls under both M/G/1 and GI/M/1-type processes, it is most commonly associated with GI/M/1-type matrices because it can be solved using the well-known matrix-geometric approach introduced in Neuts (1981) (which we outlined in Section 2.2), and provides simple closed-form formulas for measures of interest such as the expected queue length. In the case of QBD processes, (9) reduces to the matrix-quadratic equation

$$\mathbf{F} + \mathbf{R}\mathbf{L} + \mathbf{R}^2\mathbf{B} = \mathbf{0}. \tag{12}$$

QBD processes have been studied extensively and several fast algorithms have been proposed for solution of (12), most notably the logarithmic-reduction algorithm, proposed by Latouche and Ramaswami (1999). Ramaswami and Latouche (1986) and Ramaswami and Wang (1996) identify several cases that allow for explicit computation of $\mathbf{R}$. Once $\mathbf{R}$ is known, $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{\pi}^{(1)}$ are obtained by solving the following system of $m+n$ equations:

$$[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}] \left[\begin{array}{c|c|c} \hat{\mathbf{L}}^{\diamond} & \hat{\mathbf{F}} & \mathbf{1}^T \\ \hat{\mathbf{B}}^{\diamond} & \mathbf{L}+\mathbf{RB} & (\mathbf{I}-\mathbf{R})^{-1}\mathbf{1}^T \end{array}\right] = [\mathbf{0} \mid 1]. \quad (13)$$

Again, the average queue length is computed as in the GI/M/1 case.

3. ETAQA Solution for M/G/1-Type Processes

In Section 2.1, we described the matrix-analytic method for solution of M/G/1-type processes. Here, we present ETAQA, an aggregated technique that computes only $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and the aggregated probability vector $\boldsymbol{\pi}^{(*)} = \sum_{i=2}^{\infty} \boldsymbol{\pi}^{(i)}$. This approach is exact and very efficient with respect to both its time and space complexity (see Section 3.2).

Block partitioning of the infinitesimal generator in (1) defines a block partitioning of the stationary probability vector $\boldsymbol{\pi}$ as $\boldsymbol{\pi} = [\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(2)}, \ldots]$ with $\boldsymbol{\pi}^{(0)} \in \mathbb{R}^m$ and $\boldsymbol{\pi}^{(i)} \in \mathbb{R}^n$, for $i \geq 1$. First, we rewrite the matrix equality $\boldsymbol{\pi}\mathbf{Q}_{M/G/1} = \mathbf{0}$ as

$$\begin{cases} \boldsymbol{\pi}^{(0)}\hat{\mathbf{L}} + \boldsymbol{\pi}^{(1)}\hat{\mathbf{B}} = \mathbf{0} \\ \boldsymbol{\pi}^{(0)}\hat{\mathbf{F}}^{(1)} + \boldsymbol{\pi}^{(1)}\mathbf{L} + \boldsymbol{\pi}^{(2)}\mathbf{B} = \mathbf{0} \\ \boldsymbol{\pi}^{(0)}\hat{\mathbf{F}}^{(2)} + \boldsymbol{\pi}^{(1)}\mathbf{F}^{(1)} + \boldsymbol{\pi}^{(2)}\mathbf{L} + \boldsymbol{\pi}^{(3)}\mathbf{B} = \mathbf{0} \\ \boldsymbol{\pi}^{(0)}\hat{\mathbf{F}}^{(3)} + \boldsymbol{\pi}^{(1)}\mathbf{F}^{(2)} + \boldsymbol{\pi}^{(2)}\mathbf{F}^{(1)} + \boldsymbol{\pi}^{(3)}\mathbf{L} + \boldsymbol{\pi}^{(4)}\mathbf{B} = \mathbf{0} \\ \qquad\qquad \vdots \end{cases}. \quad (14)$$

The first step toward solution of an M/G/1-type process is computation of $\mathbf{G}$. We assume that $\mathbf{G}$ is available, i.e., it has been computed using an efficient iterative method, e.g., the cyclic-reduction algorithm of Bini et al. (2000), or that it can be explicitly obtained if the process falls in one of the cases identified by Ramaswami and Latouche (1986) and Ramaswami and Wang (1996).

THEOREM 1. *Given an ergodic CTMC with infinitesimal generator $\mathbf{Q}_{M/G/1}$ having the structure* (1)*, with stationary probability vector $\boldsymbol{\pi} = [\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(2)}, \ldots]$, the system of linear equations*

$$\mathbf{xX} = [1, \mathbf{0}], \quad (15)$$

where $\mathbf{X} \in \mathbb{R}^{(m+2n)\times(m+2n)}$, defined as

$$\mathbf{X} = \left[\begin{array}{c|c|c|c} \mathbf{1}^T & \hat{\mathbf{L}} & \hat{\mathbf{F}}^{(1)} - \sum_{i=3}^{\infty}\hat{\mathbf{S}}^{(i)}\mathbf{G} & (\sum_{i=2}^{\infty}\hat{\mathbf{F}}^{(i)} + \sum_{i=3}^{\infty}\hat{\mathbf{S}}^{(i)}\mathbf{G})^{\diamond} \\ \mathbf{1}^T & \hat{\mathbf{B}} & \mathbf{L} - \sum_{i=2}^{\infty}\mathbf{S}^{(i)}\mathbf{G} & (\sum_{i=1}^{\infty}\mathbf{F}^{(i)} + \sum_{i=2}^{\infty}\mathbf{S}^{(i)}\mathbf{G})^{\diamond} \\ \mathbf{1}^T & \mathbf{0} & \mathbf{B} - \sum_{i=1}^{\infty}\mathbf{S}^{(i)}\mathbf{G} & (\sum_{i=1}^{\infty}\mathbf{F}^{(i)} + \mathbf{L} + \sum_{i=1}^{\infty}\mathbf{S}^{(i)}\mathbf{G})^{\diamond} \end{array}\right], \quad (16)$$

admits a unique solution $\mathbf{x} = [\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(*)}]$, where $\boldsymbol{\pi}^{(*)} = \sum_{i=2}^{\infty}\boldsymbol{\pi}^{(i)}$.

PROOF. We first show that $[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(*)}]$ is a solution of (15) by verifying that it satisfies four matrix equations corresponding to the four sets of columns we used to define $\mathbf{X}$.

(i) The first equation is the normalization constraint:

$$\boldsymbol{\pi}^{(0)}\mathbf{1}^T + \boldsymbol{\pi}^{(1)}\mathbf{1}^T + \boldsymbol{\pi}^{(*)}\mathbf{1}^T = 1. \quad (17)$$

(ii) The second set of m equations is the first line in (14):

$$\boldsymbol{\pi}^{(0)}\hat{\mathbf{L}} + \boldsymbol{\pi}^{(1)}\hat{\mathbf{B}} = \mathbf{0}. \quad (18)$$

(iii) The third set of n equations is derived beginning from the second line in (14):

$$\boldsymbol{\pi}^{(0)}\hat{\mathbf{F}}^{(1)} + \boldsymbol{\pi}^{(1)}\mathbf{L} + \boldsymbol{\pi}^{(2)}\mathbf{B} = \mathbf{0}.$$

Because our solution does not compute $\boldsymbol{\pi}^{(2)}$ explicitly, we rewrite $\boldsymbol{\pi}^{(2)}$ such that it is expressed in terms of $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$ only. By substituting $\boldsymbol{\pi}^{(2)}$ in the above equation we obtain

$$\boldsymbol{\pi}^{(0)}\hat{\mathbf{F}}^{(1)} + \boldsymbol{\pi}^{(1)}\mathbf{L} + \boldsymbol{\pi}^{(*)}\mathbf{B} - \sum_{j=3}^{\infty}\boldsymbol{\pi}^{(j)}\mathbf{B} = \mathbf{0}. \quad (19)$$

To compute the sum $\sum_{j=3}^{\infty}\boldsymbol{\pi}^{(j)}$, we use Ramaswami's recursive formula (3) and obtain

$$\begin{aligned} \boldsymbol{\pi}^{(3)} &= -(\boldsymbol{\pi}^{(0)}\hat{\mathbf{S}}^{(3)} + \boldsymbol{\pi}^{(1)}\mathbf{S}^{(2)} + \boldsymbol{\pi}^{(2)}\mathbf{S}^{(1)})(\mathbf{S}^{(0)})^{-1} \\ \boldsymbol{\pi}^{(4)} &= -(\boldsymbol{\pi}^{(0)}\hat{\mathbf{S}}^{(4)} + \boldsymbol{\pi}^{(1)}\mathbf{S}^{(3)} + \boldsymbol{\pi}^{(2)}\mathbf{S}^{(2)} + \boldsymbol{\pi}^{(3)}\mathbf{S}^{(1)})(\mathbf{S}^{(0)})^{-1} \\ \boldsymbol{\pi}^{(5)} &= -(\boldsymbol{\pi}^{(0)}\hat{\mathbf{S}}^{(5)} + \boldsymbol{\pi}^{(1)}\mathbf{S}^{(4)} + \boldsymbol{\pi}^{(2)}\mathbf{S}^{(3)} \\ &\qquad + \boldsymbol{\pi}^{(3)}\mathbf{S}^{(2)} + \boldsymbol{\pi}^{(4)}\mathbf{S}^{(1)})(\mathbf{S}^{(0)})^{-1}, \\ &\qquad\qquad \vdots \end{aligned} \quad (20)$$

where $\hat{\mathbf{S}}^{(i)}$, for $i \geq 3$, and $\mathbf{S}^{(j)}$, for $j \geq 0$ are determined using the definitions in (4).

From the definition of $\mathbf{G}$ in (2), it follows that $\mathbf{B} = -(\mathbf{L} + \sum_{i=1}^{\infty}\mathbf{F}^{(i)}\mathbf{G}^i)\mathbf{G} = -\mathbf{S}^{(0)}\mathbf{G}$. After summing all equations in (20) by part and multiplying by $\mathbf{B}$, we obtain

$$\sum_{j=3}^{\infty}\boldsymbol{\pi}^{(j)}\mathbf{B} = \left(\boldsymbol{\pi}^{(0)}\sum_{i=3}^{\infty}\hat{\mathbf{S}}^{(i)} + \boldsymbol{\pi}^{(1)}\sum_{i=2}^{\infty}\mathbf{S}^{(i)} + \sum_{j=2}^{\infty}\boldsymbol{\pi}^{(j)}\sum_{i=1}^{\infty}\mathbf{S}^{(i)}\right) \cdot (\mathbf{S}^{(0)})^{-1}\mathbf{S}^{(0)}\mathbf{G},$$

$\mathbf{V}^{(0)}$	$\mathbf{V}^{(1)}$	$\mathbf{V}^{(2)}$	$\mathbf{V}^{(3)}$	$\cdots$
$\hat{\mathbf{L}}$	$\hat{\mathbf{F}}^{(1)}$	$\hat{\mathbf{F}}^{(2)}$	$\hat{\mathbf{F}}^{(3)}$	$\cdots$
$\hat{\mathbf{B}}$	$\mathbf{L}$	$\mathbf{F}^{(1)}$	$\mathbf{F}^{(2)}$	$\cdots$
$\mathbf{0}$	$\mathbf{B}$	$\mathbf{L}$	$\mathbf{F}^{(1)}$	$\cdots$
$\mathbf{0}$	$\mathbf{0}$	$\mathbf{B}$	$\mathbf{L}$	$\cdots$
$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{B}$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

$\mathbf{U}$
$\sum_{i=2}^{\infty} \hat{\mathbf{F}}^{(i)}$
$\sum_{i=1}^{\infty} \mathbf{F}^{(i)}$
$\mathbf{L} + \sum_{i=1}^{\infty} \mathbf{F}^{(i)}$
$\mathbf{B} + \mathbf{L} + \sum_{i=1}^{\infty} \mathbf{F}^{(i)}$
$\mathbf{B} + \mathbf{L} + \sum_{i=1}^{\infty} \mathbf{F}^{(i)}$
$\vdots$

$\mathbf{W}^{(1)}$	$\mathbf{W}^{(2)}$	$\mathbf{W}^{(3)}$	$\cdots$
$\hat{\mathbf{S}}^{(3)}\mathbf{G}$	$\hat{\mathbf{S}}^{(4)}\mathbf{G}$	$\hat{\mathbf{S}}^{(5)}\mathbf{G}$	$\cdots$
$\mathbf{S}^{(2)}\mathbf{G}$	$\mathbf{S}^{(3)}\mathbf{G}$	$\mathbf{S}^{(4)}\mathbf{G}$	$\cdots$
$\mathbf{S}^{(1)}\mathbf{G}$	$\mathbf{S}^{(2)}\mathbf{G}$	$\mathbf{S}^{(3)}\mathbf{G}$	$\cdots$
$-\mathbf{B}$	$\mathbf{S}^{(1)}\mathbf{G}$	$\mathbf{S}^{(2)}\mathbf{G}$	$\cdots$
$\mathbf{0}$	$-\mathbf{B}$	$\mathbf{S}^{(1)}\mathbf{G}$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

$\mathbf{Y}$
$\hat{\mathbf{F}}^{(1)} - \sum_{i=3}^{\infty} \hat{\mathbf{S}}^{(i)}\mathbf{G}$
$\mathbf{L} - \sum_{i=2}^{\infty} \mathbf{S}^{(i)}\mathbf{G}^{i}$
$\mathbf{B} - \sum_{i=1}^{\infty} \mathbf{S}^{(i)}\mathbf{G}^{i}$
$\mathbf{B} - \sum_{i=1}^{\infty} \mathbf{S}^{(i)}\mathbf{G}^{i}$
$\mathbf{B} - \sum_{i=1}^{\infty} \mathbf{S}^{(i)}\mathbf{G}^{i}$
$\vdots$

$\mathbf{Z}$
$\sum_{i=2}^{\infty} \hat{\mathbf{F}}^{(i)} + \sum_{i=3}^{\infty} \hat{\mathbf{S}}^{(i)}\mathbf{G}$
$\sum_{i=1}^{\infty} \mathbf{F}^{(i)} + \sum_{i=2}^{\infty} \mathbf{S}^{(i)}\mathbf{G}^{i}$
$\mathbf{L} + \sum_{i=1}^{\infty} \mathbf{F}^{(i)} + \sum_{i=2}^{\infty} \mathbf{S}^{(i)}\mathbf{G}^{i}$
$\mathbf{L} + \sum_{i=1}^{\infty} \mathbf{F}^{(i)} + \sum_{i=2}^{\infty} \mathbf{S}^{(i)}\mathbf{G}^{i}$
$\mathbf{L} + \sum_{i=1}^{\infty} \mathbf{F}^{(i)} + \sum_{i=2}^{\infty} \mathbf{S}^{(i)}\mathbf{G}^{i}$
$\vdots$

Figure 2 The Blocks of Column Vectors Used to Prove Linear Independence

which further results in

$$\sum_{j=3}^{\infty} \boldsymbol{\pi}^{(j)}\mathbf{B} = \boldsymbol{\pi}^{(0)} \sum_{i=3}^{\infty} \hat{\mathbf{S}}^{(i)}\mathbf{G} + \boldsymbol{\pi}^{(1)} \sum_{i=2}^{\infty} \mathbf{S}^{(i)}\mathbf{G} + \boldsymbol{\pi}^{(*)} \sum_{i=1}^{\infty} \mathbf{S}^{(i)}\mathbf{G}. \tag{21}$$

Substituting (21) into (19), we obtain the third set of equations as a function of $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$ only

$$\boldsymbol{\pi}^{(0)}\left(\hat{\mathbf{F}}^{(1)} - \sum_{i=3}^{\infty} \hat{\mathbf{S}}^{(i)}\mathbf{G}\right) + \boldsymbol{\pi}^{(1)}\left(\mathbf{L} - \sum_{i=2}^{\infty} \mathbf{S}^{(i)}\mathbf{G}\right) + \boldsymbol{\pi}^{(*)}\left(\mathbf{B} - \sum_{i=1}^{\infty} \mathbf{S}^{(i)}\mathbf{G}\right) = \mathbf{0}. \tag{22}$$

(iv) Another set of n equations is obtained by summing all lines in (14) starting from the third line:

$$\boldsymbol{\pi}^{(0)} \sum_{i=2}^{\infty} \hat{\mathbf{F}}^{(i)} + \boldsymbol{\pi}^{(1)} \sum_{i=1}^{\infty} \mathbf{F}^{(i)} + \sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)}\left(\mathbf{L} + \sum_{i=1}^{\infty} \mathbf{F}^{(i)}\right) + \sum_{j=3}^{\infty} \boldsymbol{\pi}^{(j)}\mathbf{B} = \mathbf{0}.$$

Since $\sum_{j=3}^{\infty} \boldsymbol{\pi}^{(j)}\mathbf{B}$ can be expressed as a function of $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$ only as in (21), the above equation can be rewritten as

$$\boldsymbol{\pi}^{(0)}\left(\sum_{i=2}^{\infty} \hat{\mathbf{F}}^{(i)} + \sum_{i=3}^{\infty} \hat{\mathbf{S}}^{(i)}\mathbf{G}\right) + \boldsymbol{\pi}^{(1)}\left(\sum_{i=1}^{\infty} \mathbf{F}^{(i)} + \sum_{i=2}^{\infty} \mathbf{S}^{(i)}\mathbf{G}\right) + \boldsymbol{\pi}^{(*)}\left(\sum_{i=1}^{\infty} \mathbf{F}^{(i)} + \mathbf{L} + \sum_{i=1}^{\infty} \mathbf{S}^{(i)}\mathbf{G}\right) = \mathbf{0}. \tag{23}$$

In steps (i) through (iv), we showed that the vector $[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(*)}]$ satisfies (17), (18), (22), and (23), so it is a solution of (15). Now we have to show that this solution is unique. For this, it is enough to prove that the rank of $\mathbf{X}$ is $m + 2n$ by showing that its $m + 2n$ rows are linearly independent.

Since the process with infinitesimal generator $\mathbf{Q}_{M/G/1}$ is ergodic, we know that $\mathbf{1}^T$ and the set of vectors corresponding to all the columns of $\mathbf{Q}_{M/G/1}$ except any one of them, are linearly independent. We also note that by multiplying a block column of $\mathbf{Q}_{M/G/1}$ by a matrix, we get a block column that is a linear combination of the columns of the selected block column. In our proof, we use multiplication of the block columns with powers of $\mathbf{G}$.

We begin from the columns of the infinitesimal generator. In Figure 2, we show the blocks of column vectors that we use in our proof. The blocks labeled $\mathbf{V}^{(i)}$ for $i \geq 0$ are the original block columns of $\mathbf{Q}_{M/G/1}$. The block $\mathbf{U}$ is obtained by summing all $\mathbf{V}^{(i)}$ for $i \geq 2$: $\mathbf{U} = \sum_{i=2}^{\infty} \mathbf{V}^{(i)}$. Blocks $\mathbf{W}^{(i)}$ for $i \geq 1$ are obtained by multiplying the block columns $\mathbf{V}^{(j)}$ for $j \geq i + 2$ with the $(j - i + 1)$th power of $\mathbf{G}$ and summing them

$$\mathbf{W}^{(i)} = \sum_{j=i}^{\infty} \mathbf{V}^{(j+2)}\mathbf{G}^{j-i+1}, \quad i \geq 1,$$

which are used to define $\mathbf{Y} = \mathbf{V}^{(1)} - \sum_{i=1}^{\infty} \mathbf{W}^{(i)}$ and $\mathbf{Z} = \mathbf{U} + \sum_{i=1}^{\infty} \mathbf{W}^{(i)}$.

In $\mathbf{X}$ defined in (16), we use the three upper blocks of $\mathbf{V}^{(0)}$, $\mathbf{Y}$, and $\mathbf{Z}$. We argue that the rank of the matrix $[\mathbf{V}^{(0)} \mid \mathbf{Y} \mid \mathbf{Z}]$ is $m + 2n - 1$ because we obtained $\mathbf{Y}$ and $\mathbf{Z}$ respectively as linear combination of blocks $\mathbf{V}^{(1)}$

and $\mathbf{V}^{(2)}$ with the blocks $\mathbf{W}^{(i)}$ for $i \geq 1$, and none of the columns used to generate $\mathbf{W}^{(i)}$ for $i \geq 1$ is from either $\mathbf{V}^{(1)}$ or $\mathbf{V}^{(2)}$. Recall that $\mathbf{Q}_{M/G/1}$ is an infinitesimal generator, so the defect is one and the rank of $[\mathbf{V}^{(0)} \mid \mathbf{Y} \mid \mathbf{Z}]$ is exactly $m+2n-1$. Substituting anyone of these columns with a column of 1s, we obtain the rank $m+2n$. □

3.1. Computing Measures of Interest for M/G/1-Type Processes

We now consider the problem of obtaining stationary measures of interest once $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$ have been computed. Traditionally, such metrics can be calculated using moment-generating functions, as in Grassman and Stanford (2000).

Here, we consider measures that can be expressed as the expected reward rate

$$r = \sum_{j=0}^{\infty} \sum_{i \in \mathcal{S}^{(j)}} \boldsymbol{\rho}_i^{(j)} \boldsymbol{\pi}_i^{(j)},$$

where $\boldsymbol{\rho}_i^{(j)}$ is the *reward rate* of state $s_i^{(j)}$. For example, to compute the expected queue length in steady state, where $\mathcal{S}^{(j)}$ represents the system states with j customers in the queue, we let $\boldsymbol{\rho}_i^{(j)} = j$. To compute the second moment of the queue length, we let $\boldsymbol{\rho}_i^{(j)} = j^2$.

Since our solution approach computes $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)}$, we rewrite r as

$$r = \boldsymbol{\pi}^{(0)} \boldsymbol{\rho}^{(0)T} + \boldsymbol{\pi}^{(1)} \boldsymbol{\rho}^{(1)T} + \sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)} \boldsymbol{\rho}^{(j)T},$$

where $\boldsymbol{\rho}^{(0)} = [\boldsymbol{\rho}_1^{(0)}, \ldots, \boldsymbol{\rho}_m^{(0)}]$ and $\boldsymbol{\rho}^{(j)} = [\boldsymbol{\rho}_1^{(j)}, \ldots, \boldsymbol{\rho}_n^{(j)}]$, for $j \geq 1$. Then, we must show how to compute the above summation without explicitly using the values of $\boldsymbol{\pi}^{(j)}$ for $j \geq 2$. We can do so if the reward rate of state $s_i^{(j)}$, for $j \geq 2$ and $i = 1, \ldots, n$, is a polynomial of degree k in j with arbitrary coefficients $\mathbf{a}_i^{[0]}, \mathbf{a}_i^{[1]}, \ldots, \mathbf{a}_i^{[k]}$:

$$\forall j \geq 2,\ \forall i \in \{1, 2, \ldots, n\},$$
$$\boldsymbol{\rho}_i^{(j)} = \mathbf{a}_i^{[0]} + \mathbf{a}_i^{[1]} j + \cdots + \mathbf{a}_i^{[k]} j^k. \tag{24}$$

The definition of $\boldsymbol{\rho}_i^{(j)}$ illustrates that the set of measures of interest that we can compute includes any moment of the probability vector $\boldsymbol{\pi}$. The only metrics of interest that we cannot compute using our aggregate approach are those whose reward rates $\boldsymbol{\rho}_i^{(j)}$ for states $s_i^{(j)}$ have different coefficients in their polynomial representation, for different inter-level index $j \geq 2$. The set of measures of interest that cannot be computed by the following methodology seldom arises in practice, since we expect that within each inter-level of the repeating portion of the process the states have similar probabilistic interpretation.

We compute $\sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)} \boldsymbol{\rho}^{(j)T}$ as

$$\begin{aligned}\sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)} \boldsymbol{\rho}^{(j)T} &= \sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)} \left(\mathbf{a}^{[0]} + \mathbf{a}^{[1]} j + \cdots + \mathbf{a}^{[k]} j^k\right)^T \\ &= \sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)} \mathbf{a}^{[0]T} + \sum_{j=2}^{\infty} j \boldsymbol{\pi}^{(j)} \mathbf{a}^{[1]T} \\ &\quad + \cdots + \sum_{j=2}^{\infty} j^k \boldsymbol{\pi}^{(j)} \mathbf{a}^{[k]T} \\ &= \mathbf{r}^{[0]} \mathbf{a}^{[0]T} + \mathbf{r}^{[1]} \mathbf{a}^{[1]T} + \cdots + \mathbf{r}^{[k]} \mathbf{a}^{[k]T},\end{aligned}$$

and the problem is reduced to computation of $\mathbf{r}^{[l]} = \sum_{j=2}^{\infty} j^l \boldsymbol{\pi}^{(j)}$, for $l = 0, \ldots, k$.

We show how $\mathbf{r}^{[k]}$, $k > 0$, can be computed recursively, starting from $\mathbf{r}^{[0]}$, which is simply $\boldsymbol{\pi}^{(*)}$. Multiplying (14) from the second line on by the appropriate factor j^k results in

$$\begin{cases} 2^k \boldsymbol{\pi}^{(0)} \hat{\mathbf{F}}^{(1)} + 2^k \boldsymbol{\pi}^{(1)} \mathbf{L} + 2^k \boldsymbol{\pi}^{(2)} \mathbf{B} = \mathbf{0} \\ 3^k \boldsymbol{\pi}^{(0)} \hat{\mathbf{F}}^{(2)} + 3^k \boldsymbol{\pi}^{(1)} \mathbf{F}^{(1)} + 3^k \boldsymbol{\pi}^{(2)} \mathbf{L} + 3^k \boldsymbol{\pi}^{(3)} \mathbf{B} = \mathbf{0} \\ \vdots \end{cases}.$$

Summing these equations by parts, we obtain

$$\begin{aligned}&\underbrace{\boldsymbol{\pi}^{(0)} \sum_{j=1}^{\infty} (j+1)^k \hat{\mathbf{F}}^{(j)}}_{\stackrel{\text{def}}{=} \hat{\mathbf{f}}} + \underbrace{\boldsymbol{\pi}^{(1)} \left(2^k \mathbf{L} + \sum_{j=1}^{\infty} (j+2)^k \mathbf{F}^{(j)}\right)}_{\stackrel{\text{def}}{=} \mathbf{f}} \\ &\quad + \sum_{h=2}^{\infty} \boldsymbol{\pi}^{(h)} \left(\sum_{j=1}^{\infty} (j+h+1)^k \mathbf{F}^{(j)} + (h+1)^k \mathbf{L}\right) \\ &\quad + \underbrace{\sum_{h=2}^{\infty} h^k \boldsymbol{\pi}^{(h)}}_{= \mathbf{r}^{[k]}} \mathbf{B} = \mathbf{0},\end{aligned}$$

which can then be rewritten as

$$\begin{aligned}&\sum_{h=2}^{\infty} \boldsymbol{\pi}^{(h)} \left[\left(\sum_{j=1}^{\infty} \sum_{l=0}^{k} \binom{k}{l} (j+1)^l h^{k-l} \mathbf{F}^{(j)}\right)\right. \\ &\qquad \left. + \left(\sum_{l=0}^{k} \binom{k}{l} 1^l h^{k-l} \mathbf{L}\right)\right] + \mathbf{r}^{[k]} \mathbf{B} = -\hat{\mathbf{f}} - \mathbf{f}.\end{aligned}$$

Exchanging the order of summations, we obtain

$$\sum_{l=0}^{k} \binom{k}{l} \underbrace{\sum_{h=2}^{\infty} \boldsymbol{\pi}^{(h)} h^{k-l}}_{= \mathbf{r}^{[k-l]}} \left(\mathbf{L} + \sum_{j=1}^{\infty} (j+1)^l \mathbf{F}^{(j)}\right) + \mathbf{r}^{[k]} \mathbf{B} = -\hat{\mathbf{f}} - \mathbf{f}.$$

Finally, isolating the case $l = 0$ in the outermost summation we obtain

$$\begin{aligned}&\mathbf{r}^{[k]} \left(\mathbf{B} + \mathbf{L} + \sum_{j=1}^{\infty} \mathbf{F}^{(j)}\right) \\ &\quad = -\hat{\mathbf{f}} - \mathbf{f} - \sum_{l=1}^{k} \binom{k}{l} \mathbf{r}^{[k-l]} \left(\mathbf{L} + \sum_{j=1}^{\infty} (j+1)^l \mathbf{F}^{(j)}\right),\end{aligned}$$

which is a linear system of the form $\mathbf{r}^{[k]}(\mathbf{B}+\mathbf{L}+\sum_{i=1}^{\infty}\mathbf{F}^{(j)})=\mathbf{b}^{[k]}$, where the right-hand side $\mathbf{b}^{[k]}$ is an expression that can be effectively computed from $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and the vectors $\mathbf{r}^{[0]}$ through $\mathbf{r}^{[k-1]}$. However, the rank of $\mathbf{B}+\mathbf{L}+\sum_{i=1}^{\infty}\mathbf{F}^{(j)}$ is $n-1$. This is true because $\mathbf{B}+\mathbf{L}+\sum_{i=1}^{\infty}\mathbf{F}^{(j)}$ is an infinitesimal generator with rank $n-1$, so the above system is under-determined. We drop any of the columns of $\mathbf{B}+\mathbf{L}+\sum_{i=1}^{\infty}\mathbf{F}^{(j)}$, resulting in

$$\mathbf{r}^{[k]}\left(\mathbf{B}+\mathbf{L}+\sum_{i=1}^{\infty}\mathbf{F}^{(j)}\right)^{\diamond}=(\mathbf{b}^{[k]})^{\diamond}, \tag{25}$$

and obtain one additional equation for $\mathbf{r}^{[k]}$ by using the flow-balance equations between $\bigcup_{l=0}^{j}\mathscr{S}^{(l)}$ and $\bigcup_{l=j+1}^{\infty}\mathscr{S}^{(l)}$ for each $j\geq 1$ and multiplying them by the appropriate factor j^k,

$$\begin{cases} 2^k\boldsymbol{\pi}^{(0)}\sum_{l=2}^{\infty}\hat{\mathbf{F}}^{(l)}\mathbf{1}^T+2^k\boldsymbol{\pi}^{(1)}\sum_{l=1}^{\infty}\mathbf{F}^{(l)}\mathbf{1}^T=2^k\boldsymbol{\pi}^{(2)}\mathbf{B}\mathbf{1}^T \\ 3^k\boldsymbol{\pi}^{(0)}\sum_{l=3}^{\infty}\hat{\mathbf{F}}^{(l)}\mathbf{1}^T+3^k\boldsymbol{\pi}^{(1)}\sum_{l=2}^{\infty}\mathbf{F}^{(l)}\mathbf{1}^T \\ \qquad\qquad +3^k\boldsymbol{\pi}^{(2)}\sum_{l=1}^{\infty}\mathbf{F}^{(l)}\mathbf{1}^T=3^k\boldsymbol{\pi}^{(3)}\mathbf{B}\mathbf{1}^T \\ \qquad\qquad \vdots \end{cases}. \tag{26}$$

Let

$$\hat{\mathbf{F}}_{[k,j]}=\sum_{l=j}^{\infty}l^k\hat{\mathbf{F}}^{(l)} \quad\text{and}\quad \mathbf{F}_{[k,j]}=\sum_{l=j}^{\infty}l^k\mathbf{F}^{(l)}, \quad j\geq 1. \tag{27}$$

We then sum all lines in (26) and obtain

$$\underbrace{\boldsymbol{\pi}^{(0)}\sum_{j=2}^{\infty}j^k\hat{\mathbf{F}}_{[0,j]}\mathbf{1}^T}_{\stackrel{\text{def}}{=}\hat{f}_c}+\underbrace{\boldsymbol{\pi}^{(1)}\sum_{j=1}^{\infty}(j+1)^k\mathbf{F}_{[0,j]}\mathbf{1}^T}_{\stackrel{\text{def}}{=}f_c}+\sum_{h=2}^{\infty}\boldsymbol{\pi}^{(h)}\sum_{j=1}^{\infty}(j+h)^k\mathbf{F}_{[0,j]}\mathbf{1}^T=\underbrace{\sum_{j=2}^{\infty}j^k\boldsymbol{\pi}^{(j)}}_{=\mathbf{r}^{[k]}}\mathbf{B}\mathbf{1}^T,$$

which, with steps analogous to those just performed to obtain (25), can be written as

$$\mathbf{r}^{[k]}(\mathbf{F}_{[1,1]}-\mathbf{B})\mathbf{1}^T=c^{[k]}, \tag{28}$$

where

$$c^{[k]}=-\left(\hat{f}_c+f_c+\sum_{l=1}^{k}\binom{k}{l}\mathbf{r}^{[k-l]}\sum_{j=1}^{\infty}j^l\mathbf{F}_{[0,j]}\cdot\mathbf{1}^T\right). \tag{29}$$

Note that the $n\times n$ matrix

$$(\mathbf{B}+\mathbf{L}+\mathbf{F}_{[0,1]})^{\diamond}\mid(\mathbf{F}_{[1,1]}-\mathbf{B})\mathbf{1}^T \tag{30}$$

has full rank. This is true because $\mathbf{B}+\mathbf{L}+\mathbf{F}_{[0,1]}$ is an infinitesimal generator with rank $n-1$, so has a unique stationary probability vector $\boldsymbol{\gamma}$ satisfying $\boldsymbol{\gamma}(\mathbf{B}+\mathbf{L}+\mathbf{F}_{[0,1]})=\mathbf{0}$. However, this same vector must satisfy $\boldsymbol{\gamma}\mathbf{B}\mathbf{1}^T>\boldsymbol{\gamma}\mathbf{F}_{[1,1]}\mathbf{1}^T$ to ensure that the process has positive drift toward $\mathscr{S}^{(0)}$, so is ergodic, and so $\boldsymbol{\gamma}(\mathbf{F}_{[1,1]}-\mathbf{B})\mathbf{1}^T<0$, which shows that $(\mathbf{F}_{[1,1]}-\mathbf{B})\mathbf{1}^T$ cannot be possibly obtained as linear combination of columns in $\mathbf{B}+\mathbf{L}+\mathbf{F}_{[0,1]}$; therefore the $n\times n$ matrix defined in (30) has full rank.

So, we can compute $\mathbf{r}^{[k]}$ using (25) and (28), i.e., solving a linear system in n unknowns (of course, we must do so first for $l=1,\ldots,k-1$). As an example, consider $\mathbf{r}^{[1]}$, which is used to compute measures such as the first moment of the queue length. In this case,

$$\mathbf{b}^{[1]}=-\left(\boldsymbol{\pi}^{(0)}\sum_{j=1}^{\infty}(j+1)\hat{\mathbf{F}}^{(j)}+\boldsymbol{\pi}^{(1)}\left(2\mathbf{L}+\sum_{j=1}^{\infty}(j+2)\mathbf{F}^{(j)}\right)+\boldsymbol{\pi}^{(*)}\left(\mathbf{L}+\sum_{j=1}^{\infty}(j+1)\mathbf{F}^{(j)}\right)\right),$$

and

$$c^{[1]}=-\left(\boldsymbol{\pi}^{(0)}\sum_{j=2}^{\infty}j\hat{\mathbf{F}}_{[0,j]}+\boldsymbol{\pi}^{(1)}\sum_{j=1}^{\infty}(j+1)\mathbf{F}_{[0,j]}+\boldsymbol{\pi}^{(*)}\sum_{j=1}^{\infty}j\mathbf{F}_{[0,j]}\right)\mathbf{1}^T.$$

In the general case that was considered here, some measures might be infinite. For example, if the sequences are summable but decrease only like $1/j^h$ for some $h>1$, then the moments of order $h-1$ or higher for the queue length do not exist (are infinite). From a practical point of view, we always store a finite set of matrices from the sequences $\{\hat{\mathbf{F}}^{(j)}:\ j\geq 1\}$ and $\{\mathbf{F}^{(j)}:\ j\geq 1\}$, so the sums of type $\hat{\mathbf{F}}_{[k,j]}$ and $\mathbf{F}_{[k,j]}$ for $j\geq 1$, $k\geq 0$ are always finite. When $\{\hat{\mathbf{F}}^{(j)}:\ j\geq 1\}$ and $\{\mathbf{F}^{(j)}:\ j\geq 1\}$ have a nicer relation, like a geometric one, the treatment in this section can be modified to simplify the different sums introduced here, and give closed-form formulas.

3.2. Time and Storage Complexity

In this section, we present a detailed comparison of ETAQA for M/G/1-type processes with the Matrix-analytic method using the FFT implementation of Ramaswami's recursive formula as in Section 2.1. The complexity analysis is within the accuracy of O-notation. $O^L(x)$ denotes the time complexity of solving a linear system described by x nonzero entries, and $\eta\{\mathbf{A}\}$ denotes the number of nonzero entries in $\mathbf{A}$. In the general case, $\eta\{\hat{\mathbf{F}}\}$ and $\eta\{\mathbf{F}\}$ mean $\eta\{\bigcup_{j=1}^{p}\hat{\mathbf{F}}^{(j)}\}$ and $\eta\{\bigcup_{j=1}^{p}\mathbf{F}^{(j)}\}$, respectively.

Since practically, we cannot store an infinite number of matrices, we store up to p matrices of type $\hat{\mathbf{F}}^{(j)}$ and $\mathbf{F}^{(j)}$, $j\geq 1$. Furthermore, for the matrix-analytic method to reach the necessary accuracy, it is necessary

to compute up to s block vectors $\boldsymbol{\pi}^{(i)}$ of the stationary probability vector $\boldsymbol{\pi}$.

We outline the required steps for each method and analyze the computation and storage complexity of each step up to the computation of the expected queue length of the process. We do not include the cost to compute the matrix $\mathbf{G}$ since *both* methodologies require the computation of $\mathbf{G}$ as a first step. $\mathbf{G}$ should be computed with an efficient method like the cyclic-reduction algorithm of Bini et al. (2000). Furthermore, we do not consider the cost of computing $\hat{\mathbf{S}}^{(i)}$ and $\mathbf{S}^{(i)}$ for $i \geq 0$ since they are required in both methodologies.

Analysis of ETAQA for M/G/1 processes is as follows

- Computation of the aggregate stationary probability vector $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$
  - —$O(p(m\eta\{\hat{\mathbf{F}}, \mathbf{G}\} + n\eta\{\mathbf{F}, \mathbf{G}\}))$ to compute $\sum_{i=k}^{\infty} \hat{\mathbf{S}}^{(i)}\mathbf{G}$ and $\sum_{i=k}^{\infty} \hat{\mathbf{S}}^{(i)}\mathbf{G}$ for $i \geq 1$, and $k = 1, 2, 3$, whose sparsity depends directly on the sparsity of $\mathbf{G}$, $\hat{\mathbf{F}}^{(i)}$, and $\mathbf{F}^{(i)}$ for $i \geq 1$.
  - —$O(p(\eta\{\hat{\mathbf{F}}\} + \eta\{\mathbf{F}\}))$ to compute $\sum_{j=1}^{\infty} \mathbf{F}^{(j)}$ and $\sum_{j=2}^{\infty} \hat{\mathbf{F}}^{(j)}$.
  - —$O^L(\eta\{\hat{\mathbf{B}}, \hat{\mathbf{L}}, \mathbf{L}, \hat{\mathbf{F}}, \mathbf{F}, \mathbf{G}\})$ for the solution of the system of $m + 2n$ linear equations.
- Storage requirements for computation of $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$
  - —$O(mn + n^2)$ to store $\sum_{i=1}^{\infty} \hat{\mathbf{S}}^{(i)}$ and $\sum_{i=1}^{\infty} \mathbf{S}^{(i)}$.
  - —$m + 2n$ to store $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$.
- Computation of the expected queue length
  - —$O(p(\eta\{\hat{\mathbf{F}}\} + \eta\{\mathbf{F}\}))$ to compute $\sum_{j=1}^{\infty} j^k \mathbf{F}^{(j)}$ and $\sum_{j=2}^{\infty} j^k \hat{\mathbf{F}}^{(j)}$, where k is constant.
  - —$O^L(\eta\{\mathbf{F}, \mathbf{L}, \mathbf{B}\})$ for solution of the sparse system of n linear equations.

Analysis of the M/G/1 matrix-analytic methodology:

- Computation of the stationary probability vector $\boldsymbol{\pi}$
  - —$O(p(m\eta\{\hat{\mathbf{F}}, \mathbf{G}\} + n\eta\{\mathbf{F}, \mathbf{G}\}))$ to compute $\hat{\mathbf{S}}^{(i)}$ for $i \geq 1$ and $\mathbf{S}^{(i)}$ for $i \geq 0$.
  - —$O(n^3 + m\eta\{\hat{\mathbf{F}}, \mathbf{G}\} + n\eta\{\hat{\mathbf{B}}\})$ to compute the inverses of $\mathbf{S}^{(0)}$ and $\sum_{j=0}^{\infty} \mathbf{S}^{(j)}$ and additional multiplications of full matrices.
  - —$O^L(m^2)$ to solve the system of m linear equations.
  - —$O(pn^3 + sn^2 + p \log p)$ (Meini 1997a) since the FFT-based version of Ramaswami's recursive formula is used to compute the s vectors of the stationary probability vector.
- Storage requirements for computation of $\boldsymbol{\pi}$
  - —$O(p(mn + n^2))$ to store all sums $\hat{\mathbf{S}}^{(i)}$ for $i \geq 1$ and $\mathbf{S}^{(i)}$ for $i \geq 0$.
  - —m to store $\boldsymbol{\pi}^{(0)}$.
  - —sn to store $\boldsymbol{\pi}^{(i)}$ for $1 \leq i < leqs$.
- Computation of the expected queue length: $O(sn)$.

The ETAQA solution is more efficient in both computation and storage. In comparison to the matrix-analytic solution, it entails only a few steps and is thus much easier to implement. Because we do not need to generate the whole stationary probability vector, s does not appear for ETAQA-M/G/1. Usually s is several times higher than p or n.

Furthermore, since the ETAQA solution does not introduce matrix inversion or multiplication, the sparsity of the original process is preserved, resulting in significant savings in both computation and storage. Sparsity of $\mathbf{G}$ is key for preserving sparsity of the original process, in both methods. There are special cases where $\mathbf{G}$ is very sparse (e.g., $\mathbf{G}$ is a single-column matrix if $\mathbf{B}$ is single-column). In these cases, the sums $\hat{\mathbf{S}}^{(i)}$ for $i \geq 1$, and $\mathbf{S}^{(i)}$ for $i \geq 0$ almost preserve the sparsity of the original process and reduce computation and storage.

4. ETAQA Solution for GI/M/1-Type Processes

We apply the same aggregation technique from Section 3 to obtain the exact aggregate solution of GI/M/1-type processes. Using the same block partitioning of $\boldsymbol{\pi}$ allows us to rewrite the matrix equality $\boldsymbol{\pi}\mathbf{Q}_{GI/M/1} = \mathbf{0}$ as

$$\begin{cases} \boldsymbol{\pi}^{(0)}\hat{\mathbf{L}} + \sum_{i=1}^{\infty} \boldsymbol{\pi}^{(i)}\hat{\mathbf{B}}^{(i)} = \mathbf{0} \\ \boldsymbol{\pi}^{(0)}\hat{\mathbf{F}} + \boldsymbol{\pi}^{(1)}\mathbf{L} + \sum_{i=2}^{\infty} \boldsymbol{\pi}^{(i)}\mathbf{B}^{(i-1)} = \mathbf{0} \\ \boldsymbol{\pi}^{(1)}\mathbf{F} + \boldsymbol{\pi}^{(2)}\mathbf{L} + \sum_{i=3}^{\infty} \boldsymbol{\pi}^{(i)}\mathbf{B}^{(i-2)} = \mathbf{0} \\ \boldsymbol{\pi}^{(2)}\mathbf{F} + \boldsymbol{\pi}^{(3)}\mathbf{L} + \sum_{i=4}^{\infty} \boldsymbol{\pi}^{(i)}\mathbf{B}^{(i-3)} = \mathbf{0} \\ \vdots \end{cases} \quad (31)$$

Assuming that $\mathbf{R}$ is available, we apply the same steps as for the case of M/G/1-type processes and formulate the following theorem:

THEOREM 2. *Given an ergodic CTMC with infinitesimal generator $\mathbf{Q}_{GI/M/1}$ having the structure* (7)*, with stationary probability vector $\boldsymbol{\pi} = [\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(2)}, \ldots]$, the system of linear equations*

$$\mathbf{x}\mathbf{X} = [1, \mathbf{0}], \quad (32)$$

where $\mathbf{X} \in \mathbb{R}^{(m+2n)\times(m+2n)}$, defined as

$$\mathbf{X} = \left[\begin{array}{c|c|c|c} \mathbf{1}^T & \hat{\mathbf{L}} & \hat{\mathbf{F}} & \mathbf{0}^{\circ} \\ \mathbf{1}^T & \hat{\mathbf{B}}^{(1)} & \mathbf{L} & \mathbf{F}^{\circ} \\ \mathbf{1}^T & \sum_{i=2}^{\infty} \mathbf{R}^{i-2}(\mathbf{I}-\mathbf{R})\hat{\mathbf{B}}^{(i)} & \sum_{i=1}^{\infty} \mathbf{R}^{i-1}(\mathbf{I}-\mathbf{R})\mathbf{B}^{(i)} & \left(\mathbf{F}+\mathbf{L}+\sum_{i=1}^{\infty} \mathbf{R}^{i}\mathbf{B}^{(i)}\right)^{\circ} \end{array}\right], \quad (33)$$

admits a unique solution $\mathbf{x} = [\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(*)}]$, *where* $\boldsymbol{\pi}^{(*)} = \sum_{i=2}^{\infty} \boldsymbol{\pi}^{(i)}$.

Proof. (i) The first equation is the normalization constraint:

$$\boldsymbol{\pi}^{(0)}\mathbf{1}^T + \boldsymbol{\pi}^{(1)}\mathbf{1}^T + \boldsymbol{\pi}^{(*)}\mathbf{1}^T = 1. \tag{34}$$

(ii) From the first line in (31), $\boldsymbol{\pi}^{(0)}\hat{\mathbf{L}} + \boldsymbol{\pi}^{(1)}\widehat{\mathbf{B}}^{(1)} + \sum_{i=2}^{\infty} \boldsymbol{\pi}^{(i)}\widehat{\mathbf{B}}^{(i)} = \mathbf{0}$. Now $\sum_{i=2}^{\infty} \boldsymbol{\pi}^{(i)}\widehat{\mathbf{B}}^{(i)} = \sum_{i=2}^{\infty}(\sum_{j=i}^{\infty} \boldsymbol{\pi}^{(j)} - \sum_{j=i+1}^{\infty} \boldsymbol{\pi}^{(j)})\widehat{\mathbf{B}}^{(i)}$ and after simple derivations that exploit the geometric relation of the stationary probability vectors $\boldsymbol{\pi}^{(j)}$, for $j \geq 2$, we obtain $\boldsymbol{\pi}^{(0)}\hat{\mathbf{L}} + \boldsymbol{\pi}^{(1)}\widehat{\mathbf{B}}^{(1)} + \boldsymbol{\pi}^{(*)} \sum_{i=2}^{\infty} \mathbf{R}^{i-2}(\mathbf{I} - \mathbf{R})\widehat{\mathbf{B}}^{(i)} = \mathbf{0}$.

(iii) From the second line of (31) and using similar derivations as in step (ii), we get $\boldsymbol{\pi}^{(0)}\hat{\mathbf{F}} + \boldsymbol{\pi}^{(1)}\mathbf{L} + \boldsymbol{\pi}^{(*)} \sum_{i=1}^{\infty} \mathbf{R}^{i-1}(\mathbf{I} - \mathbf{R})\mathbf{B}^{(i)} = \mathbf{0}$.

(iv) By summing the remaining lines in (31), $\boldsymbol{\pi}^{(1)}\mathbf{F} + \boldsymbol{\pi}^{(*)}(\mathbf{L} + \mathbf{F}) + \sum_{i=3}^{\infty}\sum_{j=i}^{\infty} \boldsymbol{\pi}^{(j)}\mathbf{B}^{(i-2)} = \mathbf{0}$. Expressing $\sum_{i=3}^{\infty}\sum_{j=i}^{\infty} \boldsymbol{\pi}^{(j)}\mathbf{B}^{(i-2)}$ as a function of $\boldsymbol{\pi}^{(*)}$, we get $\boldsymbol{\pi}^{(1)}\mathbf{F} + \boldsymbol{\pi}^{(*)}(\mathbf{L} + \mathbf{F} + \sum_{i=1}^{\infty} \mathbf{R}^i \cdot \mathbf{B}^{(i)}) = \mathbf{0}$.

$\mathbf{X}$ has full rank because $\mathbf{Q}_{GI/M/1}$ has a defect of one. We obtained the second and third block columns in $\mathbf{X}$ by keeping their respective first two upper blocks in the first block column of $\mathbf{Q}_{GI/M/1}$ and substituting the remaining lower blocks with one block that results as a linear combination of the remaining lower blocks within the same block column of $\mathbf{Q}_{GI/M/1}$. We obtained the fourth block column in $\mathbf{X}$ by keeping the first two upper blocks from the third block column of $\mathbf{Q}_{GI/M/1}$ and substituting the rest with one block that results as a linear combination of the remaining lower blocks of the third block column in $\mathbf{Q}_{GI/M/1}$ plus all remaining blocks in $\mathbf{Q}_{GI/M/1}$ (i.e., from the fourth block column of $\mathbf{Q}_{GI/M/1}$ onward). Substituting anyone of these columns with a column of ones, we obtain the rank $m + 2n$. □

4.1. Computing Measures of Interest for GI/M/1-Type Processes

For GI/M/1-type processes, as for M/G/1-type processes, ETAQA allows computation of the reward rate of state $s_i^{(j)}$, for $j \geq 2$ and $i = 1, \ldots, n$, if it is a polynomial of degree k in j with arbitrary coefficients $\mathbf{a}_i^{[0]}, \mathbf{a}_i^{[1]}, \ldots, \mathbf{a}_i^{[k]}$:

$$\forall\, j \geq 2,\ \forall\, i \in \{1, 2, \ldots, n\}, \quad \boldsymbol{\rho}_i^{(j)} = \mathbf{a}_i^{[0]} + \mathbf{a}_i^{[1]} j + \cdots + \mathbf{a}_i^{[k]} j^k.$$

We follow the same steps as those in Section 3.1. $\mathbf{r}^{[k]}$ is obtained by solving the system of linear equations

$$\mathbf{r}^{[k]}\left[\left(\mathbf{F} + \mathbf{L} + \sum_{i=1}^{\infty} \mathbf{R}^{i-1}\mathbf{B}^{(i)}\right)^{\diamond} \middle| \left((\mathbf{I} - \mathbf{R})\sum_{j=2}^{\infty}\sum_{i=j}^{\infty} \mathbf{R}^{i-2}\widehat{\mathbf{B}}^{(i)} + \sum_{j=1}^{\infty}\sum_{i=j}^{\infty} \mathbf{R}^{i-1}\mathbf{B}^{(i)} - \mathbf{F}\right)\mathbf{1}^T\right] = [(\mathbf{b}^{[k]})^{\diamond} \mid c^{[k]}], \tag{35}$$

where

$$\mathbf{b}^{[k]} = -\left(\boldsymbol{\pi}^{(0)}2^k\hat{\mathbf{F}} + \boldsymbol{\pi}^{(1)}(2^k\mathbf{L} + 3^k\mathbf{F}) + \sum_{l=1}^{k}\binom{k}{l}\mathbf{r}^{[k-l]}(\mathbf{L} + 2^l\mathbf{F})\right)$$

and

$$c^{[k]} = -\left(2^k\boldsymbol{\pi}^{(1)}\mathbf{F} + \sum_{l=1}^{k}\binom{k}{l}\mathbf{r}^{[k-l]} \cdot \left(\sum_{j=2}^{\infty}\sum_{i=j}^{\infty}((i-2)^l\mathbf{I} - (i-1)^l\mathbf{R})\mathbf{R}^{i-2}\widehat{\mathbf{B}}^{(i)} + \sum_{j=0}^{\infty} j^l \sum_{i=j}^{\infty} \mathbf{R}^i\mathbf{B}^{(i+1)} - \mathbf{F}\right)\right)\mathbf{1}^T.$$

The $n \times n$ matrix in (35) has full rank, because the proof follows the same steps as those in the proof of Theorem 2.

4.2. Time and Storage Complexity

In this section, we present a detailed comparison of our ETAQA solution with the matrix-geometric solution in Section 2.2. The complexity analysis is within the accuracy of O-notation. We assume that up to p of the $\widehat{\mathbf{B}}^{(j)}$, and $\mathbf{B}^{(j)}$, $j \geq 1$ matrices are stored. We outline the required steps for each method and analyze the computation and storage complexity up to the computation of the expected queue length. Since both methods require $\mathbf{R}$, we do not include this cost in our analysis and assume that it is efficiently computed.

Analysis of ETAQA-GI/M/1 solution:

- Computation of the aggregate stationary probability vectors $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$
  - —$O(p(m\eta\{\widehat{\mathbf{B}}, \mathbf{R}\} + n\eta\{\mathbf{B}, \mathbf{R}\}))$ to compute $\sum_{i=2}^{\infty} \mathbf{R}^{i-j}(\mathbf{I} - \mathbf{R})\widehat{\mathbf{B}}^{(i)}$ for $j = 1, 2$, and $\sum_{i=1}^{\infty} \mathbf{R}^{i-j}(\mathbf{I} - \mathbf{R})\mathbf{B}^{(i)}$ for $j = 0, 1$.
  - —$O^L(\eta\{\hat{\mathbf{L}}, \hat{\mathbf{F}}, \widehat{\mathbf{B}}, \mathbf{L}, \mathbf{B}, \mathbf{R}\})$ to solve a system of $m + 2n$ linear equations.
- Storage requirements for computation of $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$
  - —$O(mn + n^2)$ to store $\sum_{i=2}^{\infty} \mathbf{R}^{i-j}(\mathbf{I} - \mathbf{R})\widehat{\mathbf{B}}^{(i)}$ for $j = 1, 2$ and $\sum_{i=1}^{\infty} \mathbf{R}^{i-j}(\mathbf{I} - \mathbf{R})\mathbf{B}^{(i)}$ for $j = 0, 1$.
  - —n^2 to store $\mathbf{R}$.
  - —$m + 2n$ to store $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$.
- Computation of the queue length
  - —$O^L(\eta\{\mathbf{F}, \mathbf{L}, \mathbf{B}, \mathbf{R}\})$ to solve a system of n linear equations.
  - —$O(p^2(m\eta\{\widehat{\mathbf{B}}, \mathbf{R}\} + n\eta\{\mathbf{B}, \mathbf{R}\}))$ for the sums required to construct matrices of the system of linear equations.

Analysis of matrix-geometric solution:

- Computation of the boundary stationary probability vectors $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{\pi}^{(1)}$

—$O(p(m\eta\{\hat{\mathbf{B}}, \mathbf{R}\} + n\eta\{\mathbf{B}, \mathbf{R}\}))$ to compute $\sum_{i=2}^{\infty} \mathbf{R}^{i-j}\hat{\mathbf{B}}^{(i)}$ for $j = 1, 2$ and $\sum_{i=1}^{\infty} \mathbf{R}^{i-j}\mathbf{B}^{(i)}$ for $j = 0, 1$.

—$O(n^3)$ to compute $(\mathbf{I} - \mathbf{R})^{-1}$.

—$O^L(\eta\{\hat{\mathbf{L}}, \hat{\mathbf{F}}, \hat{\mathbf{B}}, \mathbf{L}, \mathbf{F}, \mathbf{B}, \mathbf{R}\})$ to solve a system of $m + n$ linear equations.

• Storage requirements for computation of $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{\pi}^{(1)}$

—$O(mn + n^2)$ to store $\sum_{i=2}^{\infty} \mathbf{R}^{i-j}\hat{\mathbf{B}}^{(i)}$ for $j = 1, 2$ and $\sum_{i=1}^{\infty} \mathbf{R}^{i-j}\mathbf{B}^{(i)}$ for $j = 0, 1$.

—$O(n^2)$ to store $\mathbf{R}$ and $(\mathbf{I} - \mathbf{R})^{-1}$.

—$m + n$ to store $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{\pi}^{(1)}$.

• Computation of queue length is $O(n^2)$ because it calculates the closed-form formula: $\boldsymbol{\pi}^{(1)}\mathbf{R}(\mathbf{I} - \mathbf{R})^{-2}\mathbf{1}^T$.

The appeal of the classic matrix-geometric method is its simplicity. The geometric relation between the vectors of the stationary probability distribution allows for simple closed-form formulas for computation of measures of interest such as the expected queue length. The ETAQA-GI/M/1 method performs better when we are interested only in computation of the probability vectors, depending on the system sparsity, the size of the matrices, and the number of stored matrices that capture the behavior of the whole process, but not when we are interested in computing measures of interest.

5. ETAQA for QBD Processes

QBD processes are essentially a subcase of both M/G/1-type and GI/M/1-type processes and can be therefore solved with either the matrix-analytic method from Section 2.1 or the matrix-geometric method from Section 2.2. Matrix-geometric is better because it is simple and provides closed-form formulas for measures of interest such as expected queue length. However, we solve QBDs using ETAQA-M/G/1, because from the complexity analysis in Sections 3.2 and 4.2, ETAQA-M/G/1 is more efficient. Assuming knowledge of $\mathbf{G}$ for a QBD process with the infinitesimal generator as in (11), the aggregate solution for the QBD process is as follows:

THEOREM 3. *Given an ergodic CTMC with infinitesimal generator* $\mathbf{Q}_{\text{QBD}}$ *having structure* (11), *with stationary probability vector* $\boldsymbol{\pi} = [\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(2)}, \ldots]$, *the system of linear equations*

$$\mathbf{x}\mathbf{X} = [1, \mathbf{0}], \tag{36}$$

where $\mathbf{X} \in \mathbb{R}^{(m+2n)\times(m+2n)}$, *defined as*

$$\mathbf{X} = \left[\begin{array}{c|c|c|c} \mathbf{1}^T & \hat{\mathbf{L}} & \hat{\mathbf{F}} & \mathbf{0}^{\diamond} \\ \hline \mathbf{1}^T & \hat{\mathbf{B}} & \mathbf{L} & \mathbf{F}^{\diamond} \\ \hline \mathbf{1}^T & \mathbf{0} & \mathbf{B} - \mathbf{F}\mathbf{G} & (\mathbf{L} + \mathbf{F} + \mathbf{F}\mathbf{G})^{\diamond} \end{array}\right], \tag{37}$$

admits a unique solution $\mathbf{x} = [\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(*)}]$, *where* $\boldsymbol{\pi}^{(*)} = \sum_{i=2}^{\infty} \boldsymbol{\pi}^{(i)}$.

PROOF. The steps are identical to those in the Proof of Theorem 1, because QBDs are a special case of M/G/1-type processes. □

5.1. Computing Measures of Interest for QBD Processes

As in the M/G/1 case, ETAQA allows the computation of the reward rate of state $s_i^{(j)}$, for $j \geq 2$ and $i = 1, \ldots, n$, if it is a polynomial of degree k in j with arbitrary coefficients $\mathbf{a}_i^{[0]}, \mathbf{a}_i^{[1]}, \ldots, \mathbf{a}_i^{[k]}$:

$$\forall\, j \geq 2,\ \forall\, i \in \{1, 2, \ldots, n\}, \quad \boldsymbol{\rho}_i^{(j)} = \mathbf{a}_i^{[0]} + \mathbf{a}_i^{[1]} j + \cdots + \mathbf{a}_i^{[k]} j^k.$$

We follow the same steps as in Section 3.1, albeit significantly simplified. Observe that that $\mathbf{r}^{[0]}$ is simply $\boldsymbol{\pi}^{(*)}$ while, for $k > 0$, $\mathbf{r}^{[k]}$ can be computed after obtaining $\mathbf{r}^{[l]}$ for $0 \leq l < k$, by solving the system of n linear equations:

$$\begin{cases} \mathbf{r}^{[k]}(\mathbf{B} + \mathbf{L} + \mathbf{F})^{\diamond} = \mathbf{b}^{[k]\diamond} \\ \mathbf{r}^{[k]}(\mathbf{F} - \mathbf{B})\mathbf{1}^T = c^{[k]}, \end{cases} \tag{38}$$

where

$$\mathbf{b}^{[k]} = -\left(2^k\boldsymbol{\pi}^{(0)}\hat{\mathbf{F}} + 2^k\boldsymbol{\pi}^{(1)}\mathbf{L} + 3^k\boldsymbol{\pi}^{(1)}\mathbf{F} + \sum_{l=1}^{k}\binom{k}{l}\left(2^l\mathbf{r}^{[k-l]}\mathbf{F} + \mathbf{r}^{[k-l]}\mathbf{L}\right)\right) \quad \text{and}$$

$$c^{[k]} = -2^k\boldsymbol{\pi}^{(1)}\mathbf{F}\mathbf{1}^T - \sum_{l=1}^{k}\binom{k}{l}\mathbf{r}^{[k-l]}\mathbf{F}\mathbf{1}^T.$$

The rank of the system of linear equations in (38) is n, since QBDs are a special case of M/G/1-type processes.

We conclude by reiterating that to compute the kth moment of the queue length, we must solve k systems of n linear equation each. In particular, the expected queue length is obtained by solving only one system of n linear equations.

5.2. Time and Storage Complexity

In this section, we present a detailed comparison of our aggregate solution for QBD processes with the matrix-geometric method for QBDs in Section 2.3. We outline the required steps for each method and analyze the computation and storage complexity up to the computation of the expected queue length. We assume that the algorithm of choice to compute $\mathbf{R}$ in the matrix-geometric solution for QBDs is logarithmic reduction, as it is the most efficient one. Therefore in our analysis we do not include the cost to compute $\mathbf{G}$, which is the first matrix to be computed by the logarithmic reduction algorithm of Latouche and Ramaswami (1999).

Analysis of ETAQA-QBD:

- Computation of the aggregate stationary probability vector $[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(*)}]$
  - —$O(n\eta\{\mathbf{F}, \mathbf{G}\})$ to compute $\mathbf{FG}$.
  - —$O^L(\hat{\mathbf{L}}, \hat{\mathbf{F}}, \hat{\mathbf{B}}, \mathbf{B}, \mathbf{L}, \mathbf{F}, \mathbf{G})$ to solve the system of $m+2n$ linear equations.
- Storage requirements for computation of $[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(*)}]$
  - —$O(n^2)$ for $\mathbf{FG}$.
  - —$m+2n$ for $[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(*)}]$.
- Computation of the queue length
  - —$O(\eta\{\mathbf{F}, \mathbf{L}, \mathbf{B}\})$ to compute $\mathbf{F}+\mathbf{L}+\mathbf{B}$ and $\mathbf{F}-\mathbf{B}$.
  - —$O^L(\eta\{\mathbf{F}, \mathbf{L}, \mathbf{B}\})$ to solve a system of n linear equations.

Analysis of the matrix-geometric method for QBDs:

- Computation of the boundary stationary probability vector $[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}]$
  - —$O(n^3)$ to compute $\mathbf{R}$ from $\mathbf{G}$ (last step of the logarithmic-reduction algorithm) using the relation $\mathbf{R} = -\mathbf{F}(\mathbf{L}+\mathbf{FG})^{-1}$ (see the Online Supplement to this paper on the journal's website).
  - —$O(n^3)$ to compute $(\mathbf{I}-\mathbf{R})^{-1}$.
  - —$O(n\eta\{\mathbf{R}, \mathbf{B}\})$ to compute $\mathbf{RB}$.
  - —$O^L(\hat{\mathbf{L}}, \hat{\mathbf{F}}, \hat{\mathbf{B}}, \mathbf{L}, \mathbf{B}, \mathbf{R})$ to solve the system of $m+n$ linear equations.
- Storage requirements to compute $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{\pi}^{(1)}$
  - —$O(n^2)$ for $\mathbf{R}$ and $(\mathbf{I}-\mathbf{R})^{-1}$.
  - —$m+n$ for $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{\pi}^{(1)}$.
- Computation of the queue length is $O(n^2)$. The computation uses the formula $\boldsymbol{\pi}^{(1)} \cdot \mathbf{R} \cdot (\mathbf{I}-\mathbf{R})^{-2} \cdot \mathbf{1}^T$.

Sparsity of $\mathbf{G}$ is key to preserving sparsity of the original process in ETAQA-QBD, while the matrix-geometric method uses $\mathbf{R}$ which is usually dense. The ETAQA-QBD solution is as efficient as the matrix-geometric method. We save storage (though this gain is not obvious using O-notation) because the aggregate solution requires only *temporal* storage of $\mathbf{FG}$, while the matrix-geometric method needs *persistent* storage of $\mathbf{R}$ and $(\mathbf{I}-\mathbf{R})^{-1}$.

6. Computational Efficiency

Section 3 analyzed using O-notation the computational and storage efficiency of ETAQA-M/G/1. Here, we present further numerical evidence that ETAQA-M/G/1 is more efficient than other methods. We use the classic Ramaswami formula and Meini's (1997a) FFT implementation of it, the most efficient algorithm to solve M/G/1-type processes. We used Meini's implementation http://www.dm.unipi.it/~meini/ric.html of the cyclic-reduction algorithm to compute $\mathbf{G}$ which is required in all three algorithms. We also used Meini's code for the FFT implementation of Ramaswami's formula, made available to us via personal communication (Meini 1997b). We implemented the ETAQA-M/G/1 method and Ramaswami formula in C. All experiments were conducted on a 450 MHz Sun Enterprise 420R server with 4 GB memory.

The M/G/1 process for our experiments is a general BMAP/M/1 queue. In practice, it is not possible to store an infinite number of $\hat{\mathbf{F}}^{(i)}$ and $\mathbf{F}^{(i)}$ matrices, $1 < i < \infty$. One should stop storing when all entries of $\hat{\mathbf{F}}^{(i)}$ and $\mathbf{F}^{(i)}$ for $i > p$ are below a sufficient threshold (i.e., *very close* to zero in a practical implementation), as suggested in Latouche and Ramaswami (1999). As illustrated in Section 3, computation time depends on both the size (i.e., parameters m and n) and the number (of stored) matrices (i.e., parameter p) that define $\mathbf{Q}$. One last parameter that affects computation time is the number s of vector probabilities that should be computed so that the normalization condition $\sum_{i=1}^{s} \boldsymbol{\pi}^{(i)} = 1$ is reached (again, within a sufficient numerical threshold).

We vary the parameters n, p, and s (for the case of the BMAP/M/1 queue, $m = n$) and provide timing results for computing the stationary vector $\boldsymbol{\pi}$ using the Ramaswami formula and FFT, and computation of $(\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(*)})$ using ETAQA-M/G/1. We also provide timings for computing the queue length for all methods. Results are presented in Figure 3.

The first experiment considers a BMAP/M/1 queue with $n = 16$ and $p = 32$, a relatively small case. The timings of the three algorithms, which do not take into consideration computation of $\mathbf{G}$, are shown as a function of s. Figure 3(a) depicts the computation cost of the probability vector and Figure 3(b) illustrates the computation cost for the queue length. The y-axis is in log-scale. The value of s does affect the execution time of both matrix-analytic approaches, but does not affect ETAQA-M/G/1. As expected, for computing the stationary vector, FFT is superior to the classic Ramaswami formula, behavior that persists when we increase p and n (see Figures 3(c) and 3(e)). ETAQA-M/G/1 consistently outperforms the other two methods and its performance is insensitive to s (see Figures 3(a), 3(c), and 3(e)).

Figures 3(b), 3(d), and 3(f) illustrate the computation cost of the queue length for the three algorithms for various values of n, p, and s. The two implementations of Ramaswami's formula have the same cost, since the same classic formula is used for computing queue length: first weight appropriately and then sum the probability vector which is already computed. The figures further confirm that the cost of solving a small system of linear equations that ETAQA-M/G/1 requires for computation of queue length is in many cases preferable to the classic methods. If this linear system increases and s is also small, then the classic methods may offer better performance.

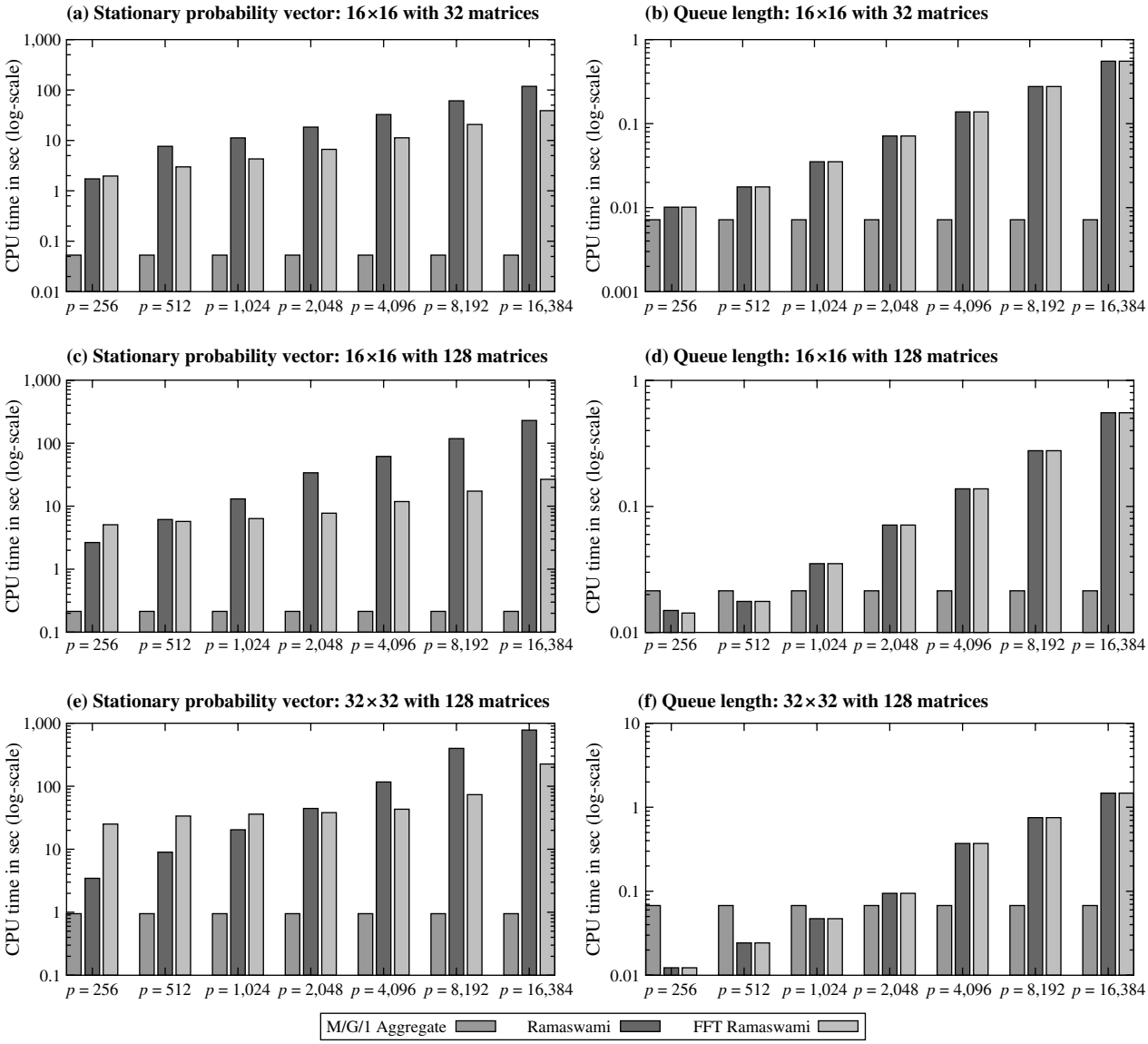

Figure 3 Execution Times in Seconds

7. Concluding Remarks

We have presented ETAQA, an aggregate approach for solving M/G/1-type, GI/M/1-type, and QBD processes. Our exposition focuses on computing efficiently the *exact* probabilities of the boundary states of the process and the aggregate probability distribution of the states in each of the equivalence classes corresponding to a specific partitioning of the remaining infinite portion of the state space. Although the method does not compute the probability distribution of all states, it still provides enough information for "mathematically exact" computation of a rich set of Markov reward functions such as the expected queue length or any of its higher moments.

We presented detailed analysis of the computation and storage complexity of our method. For M/G/1-type processes, ETAQA requires a few simple steps that provide significant savings in both computation and storage when compared with the traditional matrix-analytic and matrix-geometric solutions, respectively. These gains are a direct outcome of the fact that ETAQA computes only the aggregate stationary probability vector instead of the entire stationary probability vector computed by the matrix-analytic methods. Additionally, ETAQA closely preserves the structure (and thus the sparsity) of the original process, thus facilitating computational gains, in contrast to the classic methods that instead introduce structures that destroy the sparsity of the original matrices.

For GI/M/1-type and QBD processes, ETAQA has the same complexity as the matrix-geometric method for computation of the stationary probability vector, but the classic method results in less expensive and more intuitively appealing formulas for the

computation of measures of interest such as the expected queue length.

An issue that often arises in numerical solutions of Markov chains is numerical stability, which has hardly been investigated, if at all, as stated in Latouche and Ramaswami (1999). Methods that recursively compute probability vectors via a formula that entails only addition and multiplication are considered numerically stable. Ramaswami's recursive formula for M/G/1-type processes is a case of a stable algorithm. Once subtractions are involved, the possibility of numerical instability increases because of the loss of significance (as discussed in Neuts 1989, p. 165). Our construction of $\mathbf{X}$ in (16) does introduce subtraction, but in Stathopoulos et al. (2005), we provide theoretical and experimental evidence that ETAQA is numerically stable.

Acknowledgments

The authors thank Beatrice Meini for her implementation of the FFT algorithm for Ramaswami's recursive formula. They also thank Mark Squillante for insightful discussions and Guy Latouche for his valuable feedback in earlier versions of this work. This work was partially supported by the National Science Foundation under Grants ITR-0428330, CCR-0098278, and ACI-0090221.

References

Bini, D. A., B. Meini. 1998. Using displacement structure for solving non-skip-free M/G/1 type Markov chains. A. S. Alfa, S. R. Chakravarthy, eds. *Advances in Matrix Analytic Methods for Stochastic Models*. Notable Publications, Neshanic Station, NJ, 17–37.

Bini, D. A., B. Meini, V. Ramaswami. 2000. Analyzing M/G/1 paradigms through QBDs: The role of the block structure in computing the matrix. G. G. Latouche, P. Taylor, eds. *Advances in Matrix Analytic Methods for Stochastic Models*. Notable Publications, Neshanic Station, NJ, 73–86.

Ciardo, G., E. Smirni. 1999. ETAQA: An efficient technique for the analysis of QBD-processes by aggregation. *Performance Eval.* **36–37** 71–93.

Ciardo, G., W. Mao, A. Riska, E. Smirni. 2004. ETAQA-MG1: An efficient technique for the analysis of M/G/1-type processes by aggregation. *Performance Eval. J.* **57** 235–260.

Grassman, W. K., D. A. Stanford. 2000. Matrix analytic methods. W. K. Grassman, ed. *Computational Probability*. Kluwer Academic Publishers, Boston, MA, 153–204.

Kleinrock, L. 1975. *Queueing Systems, Volume I: Theory*. Wiley, New York.

Latouche, G. 1993. Algorithms for infinite Markov chains with repeating columns. C. Meyer, R. J. Plemmons, eds. *Linear Algebra, Markov Chains, and Queueing Models*, Vol. 48. IMA Volumes in Mathematics and its Applications, Springer Verlag, Berlin, Germany, 231–265.

Latouche, G., V. Ramaswami. 1999. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM Series on Statistics and Applied Probability, SIAM, Philadelphia, PA.

Latouche, G., G. W. Stewart. 1995. Numerical methods for M/G/1-type queues. W. J. Stewart, ed. *Computations with Markov Chains*. Kluwer Academic Publishers, Boston, MA, 571–581.

Lucantoni, D. M. 1983. *An Algorithmic Analysis of a Communication Model with Retransmission of Flawed Messages*. Pitman, Boston, MA.

Meini, B. 1997a. An improved FFT-based version of Ramaswami's formula. *Comm. Statist. Stochastic Models* **13** 223–238.

Meini, B. 1997b. Implementation of the FFT-based version of Ramaswami's formula. Department of Mathematics, University of Pisa, Pisa, Italy.

Meini, B. 1998. Solving M/G/1-type Markov chains: Recent advances and applications. *Comm. Statist. Stochastic Models* **14** 479–496.

Nelson, R. 1995. *Probability, Stochastic Processes, and Queueing Theory*. Springer-Verlag, Berlin, Germany.

Neuts, M. F. 1981. *Matrix-Geometric Solutions in Stochastic Models*. Johns Hopkins University Press, Baltimore, MD.

Neuts, M. F. 1989. *Structured Stochastic Matrices of M/G/1-Type and Their Applications*. Marcel Dekker, New York.

Ramaswami, V. 1988. A stable recursion for the steady state vector in Markov chains of M/G/1-type. *Comm. Statist. Stochastic Models* **4** 183–189.

Ramaswami, V., G. Latouche. 1986. A general class of Markov processes with explicit matrix-geometric solutions. *OR Spektrum* **8** 209–218.

Ramaswami, V., J. L. Wang. 1996. A hybrid analysis/simulation for ATM performance with application to quality-of-service of CBR traffic. *Telecomm. Systems* **5** 25–48.

Riska, A., E. Smirni. 2002a. Exact aggregate solutions for M/G/1-type Markov processes. *Proc. ACM SIGMETRICS Conf.* Marina del Rey, CA, 86–96.

Riska, A., E. Smirni. 2002b. M/G/1-type Markov processes: A tutorial. M. C. Calzarossa, S. Tucci, eds. *Performance Evaluation of Complex Computer Systems: Techniques and Tools, LNCS 2459*. Springer-Verlag, Berlin, Germany, 36–63.

Squillante, M. S. 1998. Matrix-analytic methods in stochastic parallel-server scheduling models. S. R. Chakravarthy, A. S. Alfa, eds. *Advances in Matrix-Analytic Methods for Stochastic Models: Lecture Notes in Pure and Applied Mathematics*. Notable Publications, Neshanic Station, NJ.

Squillante, M. S. 2000. Matrix-analytic methods: Applications, results and software tools. G. Latouche, P. Taylor, eds. *Advances in Matrix-Analytic Methods for Stochastic Models*. Notable Publications, Neshanic Station, NJ.

Stathopoulos, A., A. Riska, Z. Hua, E. Smirni. 2005. Bridging ETAQA and Ramaswami's formula for the solution of M/G/1 processes. *Performance Eval. J.* **62** 331–348.