ETAQA: An Efficient Technique for the Analysis of QBD-processes by Aggregation

Gianfranco Ciardo Evgenia Smirni*
Department of Computer Science
College of William and Mary
Williamsburg, VA 23187-8795

Abstract

In this paper we present ETAQA, an Efficient Technique for the Analysis of QBD-processes by Aggregation. We concentrate on processes satisfying a particular repetitive structure that frequently occurs in modeling of computer and communication systems. The proposed methodology exploits this special structure to evaluate the *aggregate* probability distribution of the states in each of the equivalence classes corresponding to a specific partitioning of the state space. Although the method does not compute the probability distribution of *all* states in the chain, not even in implicit recursive form, it provides the necessary information to easily compute an extensive set of Markov reward functions such as the queue length or any of its higher moments. The proposed technique has excellent computational and storage complexity and results in significant savings when compared with other traditional solution techniques such as the matrix geometric approach.

1 Introduction

Over the last two decades, considerable effort has been put into the development of techniques for the exact analysis of a general and frequently encountered class of queuing models. In these models, the embedded Markov chains are two-dimensional generalizations of those arising from the embedding of elementary M/G/1 or G/M/1 queues [7]. The intersection of these two cases corresponds to the so-called quasi-birth-death (QBD) processes with a repetitive "matrix-geometric" structure, which occur frequently in queuing models of computer and communication systems.

In his seminal contribution, Neuts showed that any Markov chain exhibiting such a matrix-geometric form can be solved using iterative algorithms that exploit the repetitive structure of the infinitesimal generator to express the stationary probabilities of the states in a recursive way [15]. If the state space is $\mathcal{S}$, it is partitioned into $\mathcal{S}^{(j)}$, $j \geq 0$, where $\mathcal{S}^{(0)} = \{s_1^{(0)}, \ldots, s_m^{(0)}\}$ contains m "boundary" states and, for $j \geq 1$, each set $\mathcal{S}^{(j)}$ contains the same number n of states, $\mathcal{S}^{(j)} = \{s_1^{(j)}, \ldots, s_n^{(j)}\}$. The boundary portion of the state space usually represent the state configurations that arise when the queue is empty or below a certain threshold of occupancy, while the remaining states correspond to the "repetitive" (infinite) portion of the state space [12, 15]. Intuitively, the addition of one customer "shifts" the state space from "level" $\mathcal{S}^{(j)}$ to level $\mathcal{S}^{(j+1)}$, by increasing the length of the

*This research has been supported by a William and Mary Summer Research Grant.

waiting queue while the server(s) occupancies remains the same [13]. Neuts observed that, if the stationary probability vector $\boldsymbol{\pi}$ is partitioned accordingly into $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, $\ldots$, the recursive relation $\boldsymbol{\pi}^{(j+1)} = \boldsymbol{\pi}^{(j)} \cdot \mathbf{R}$ holds, so that the problem is essentially reduced to the computation of the "rate" matrix $\mathbf{R}$.

MAGIC [22] and MGMtool [4] are examples of computer packages that provide a solution framework implementing this idea. Although widely applicable, the matrix-geometric method suffers from a major drawback: the need for computing and storing the full square matrix $\mathbf{R}$, required for the calculation of both the state probabilities and the associated performance measures [13]. Explicit solutions exist that avoid the iterative calculation [1, 10, 20] by reducing the global balance equations of the repetitive portion of the chain from second order difference equations to first order difference equations. Additionally, if the interaction of the repetitive part of the chain is of a specific form [17, 13], the solution of the system becomes less expensive than the traditional matrix-geometric technique because there are ways to circumvent the iterative calculation of the matrix $\mathbf{R}$.

In this paper, we introduce ETAQA (pronounced Ithaca), an Efficient Technique for the Analysis of QBD-processes by Aggregation. We restrict our attention to QBD processes with matrix-geometric form for which the "return" from level $\mathcal{S}^{(j+1)}$ to level $\mathcal{S}^{(j)}$ is always directed toward a single state in $\mathcal{S}^{(j)}$, which we identify as $s_n^{(j)}$, by convention. If this condition is met, ETAQA offers a computationally *efficient* method for the solution of the system, i.e., for the computation of steady-state probabilities and the computation of a rich set of performance measures that can be expressed as state reward functions.

Our approach is derived from first principles using the classic Chapman-Kolmogorov [7] equations that relate the flow into and out of each state in equilibrium. We exploit the structure of the repetitive portion of the chain and instead of evaluating the probability distribution of *all* states in the chain, we calculate the *aggregate* probability distribution of the n equivalence classes $\mathcal{T}_i = \{s_i^{(j)} : j \geq 2\}$, for $1 \leq i \leq n$, corresponding to a specific partitioning of the repetitive portion of the chain (see Figure 1), as well as the stationary probability distribution of all states in states in $\mathcal{S}^{(0)} \cup \mathcal{S}^{(1)}$.

ETAQA has excellent computational and storage complexity, since it only requires the solution of one linear system in $(m + 2n)$ unknowns, to compute the stationary probabilities. Furthermore, only the original matrices specifying the Markov chain need to be stored. These matrices are ordinarily quite sparse, hence very large models can be solved by using sparse storage techniques [16] and iterative numerical methods. We stress that, instead, the computational complexity of the matrix-geometric method is much higher, even if our requirement that transitions from states in level $\mathcal{S}^{(j+1)}$ to level $\mathcal{S}^{(j)}$ are always directed toward a single state $s_n^{(j)} \in \mathcal{S}^{(j)}$ is taken into account when computing $\mathbf{R}$. Furthermore, the explicit storage of $\mathbf{R}$ in the matrix-geometric method still requires n^2 memory locations, as our restriction does not imply a particular structure in $\mathbf{R}$ that can be exploited for storage purposes.

ETAQA differs from the matrix-geometric approach in that we do not evaluate the probability of *all* states in the chain, not even in implicit recursive form. Nevertheless, it provides the necessary information to easily compute various Markov reward functions of interest: any linear reward function (such as the expected queue length) requires the additional solution of one linear system in n unknowns; in general, the computation of the expected reward rate when the reward rates of the states in $\mathcal{S}^{(j)}$ is a k^{th}-degree polynomial in j (e.g., to compute the k^{th} moment of the queue length), requires the successive solution of k linear systems in n unknowns.

This paper is organized as follows. Section 2 overviews the matrix-geometric approach and other

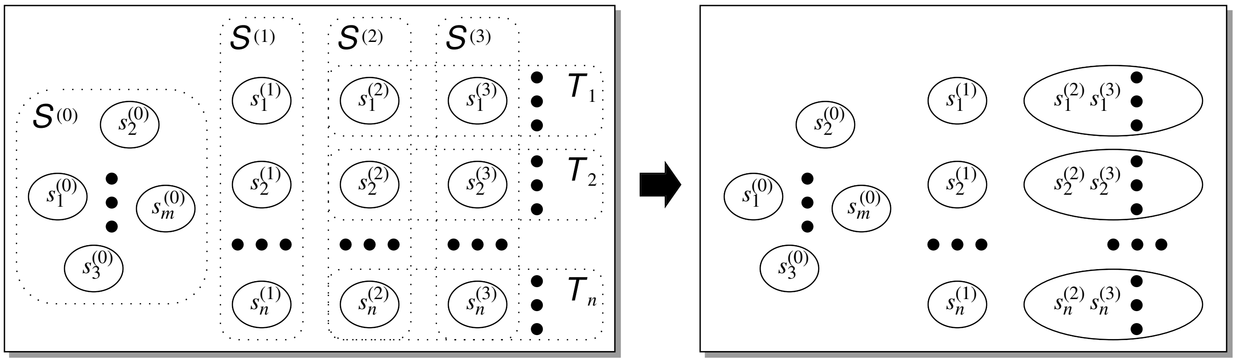

Figure 1: Partitioning of the state space (left) and its aggregation (right).

background information useful in defining our method and is followed by a motivational example in Section 3. In Section 4 we formally define the proposed method and the conditions under which it is applicable. Section 5 continues with the formalization of the methodology for the computation of performance measures of interest via the use of reward functions. In Section 6 we show how to extend the applicability of our method by repartitioning the state space. Performance comparisons and the relation of ETAQA with alternative solution methods are given in Section 7. Section 8 gives a short survey of several published applications where ETAQA applies. Finally, our conclusions are given in Section 9.

2 Background: the matrix-geometric approach

If a Markov process exhibits a matrix-geometric structure, then its state space can be 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 $j \geq 1$, which correspond to the repetitive portion of the chain. The repetitive structure allows for a recursive formulation of the stationary probabilities of the states in the chain and facilitates their computation. We focus on continuous-time Markov chains representing QBD processes. Their infinitesimal generator $\mathbf{Q}$ is then expressed as [12]:

$$\begin{bmatrix} \mathbf{L}^{(00)} & \mathbf{F}^{(01)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{B}^{(10)} & \mathbf{L} & \mathbf{F} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

where $\mathbf{L}^{(00)}$ describes the "local" transition rates between states in $\mathcal{S}^{(0)}$, $\mathbf{F}^{(01)}$ describes the "forward" transition rates from states in $\mathcal{S}^{(0)}$ to states in $\mathcal{S}^{(1)}$, $\mathbf{B}^{(10)}$ describes the "backward" transition rates from states in $\mathcal{S}^{(1)}$ to states in $\mathcal{S}^{(0)}$, $\mathbf{F}$ describes transition rates from states in $\mathcal{S}^{(j)}$ to states in $\mathcal{S}^{(j+1)}$, $\mathbf{L}$ describes transition rates between states in $\mathcal{S}^{(j)}$, $\mathbf{B}$ describes transition rates from states in $\mathcal{S}^{(j)}$ to states in $\mathcal{S}^{(j-1)}$ (for $j \geq 1$), and $\mathbf{0}$ is a matrix of all zeros of the appropriate dimension.

Neuts [15] proposed an algorithm that takes advantage of the repetitive structure of the chain to compute the stationary probabilities of *all* states. Let $\boldsymbol{\pi}^{(j)}$ be the stationary probability vector for states in $\mathcal{S}^{(j)}$ (for $j \geq 0$). The equation for the repetitive portion of the process in block matrix form is:

$$\boldsymbol{\pi}^{(j-1)} \cdot \mathbf{F} + \boldsymbol{\pi}^{(j)} \cdot \mathbf{L} + \boldsymbol{\pi}^{(j+1)} \cdot \mathbf{B} = 0, \qquad j \geq 2$$

and since the values of $\boldsymbol{\pi}^{(j)}, j \geq 2$ have a matrix-geometric form, then

$$\boldsymbol{\pi}^{(j)} = \boldsymbol{\pi}^{(1)} \cdot \mathbf{R}^{j-1}, \qquad j \geq 2 \ ,$$

which implies

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

This quadratic equation in the matrix $\mathbf{R}$ can be solved numerically using the following iterative procedure:

$$\begin{aligned} \mathbf{R}(0) &= 0 \\ \mathbf{R}(k+1) &= -\mathbf{F} \cdot \mathbf{L}^{-1} - \mathbf{R}^2(k) \cdot \mathbf{B} \cdot \mathbf{L}^{-1}, \qquad k \geq 0, \end{aligned} \tag{1}$$

stopping when $||\mathbf{R}(k+1) - \mathbf{R}(k)||$ is sufficiently small. To determine the stationary probabilities, one focuses on the equations for the initial portion of the matrix, which can be written in matrix form as

$$[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}] \cdot \begin{bmatrix} \mathbf{L}^{(00)} & \mathbf{F}^{(01)} \\ \mathbf{B}^{(10)} & \mathbf{L} + \mathbf{R} \cdot \mathbf{B} \end{bmatrix} = \mathbf{0} \tag{2}$$

and, together with the normalization condition

$$\boldsymbol{\pi}^{(0)} \cdot \mathbf{1}^T + \boldsymbol{\pi}^{(1)} \cdot \sum_{j=1}^{\infty} \mathbf{R}^{j-1} \cdot \mathbf{1}^T = 1 \qquad \text{that is} \qquad \boldsymbol{\pi}^{(0)} \cdot \mathbf{1}^T + \boldsymbol{\pi}^{(1)} \cdot (\mathbf{I} - \mathbf{R})^{-1} \cdot \mathbf{1}^T = 1,$$

(where $\mathbf{1}^T$ is a column vector of ones of the appropriate dimension) yields a unique solution.

Once $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{\pi}^{(1)}$ have been obtained, any other $\boldsymbol{\pi}^{(j)}$, for $j \geq 2$, can be obtained numerically as $\boldsymbol{\pi}^{(j)} = \boldsymbol{\pi}^{(1)} \cdot \mathbf{R}^{j-1}$. Of course, only a finite number of such vectors can be obtained in practice, but this is not a problem since their norm decreases toward zero according to a geometric factor equal to the spectral radius $\rho(\mathbf{R}) < 1$. Even more importantly, though, many useful performance metrics, such as expected system utilization, throughput, or queue length, can be computed exactly from $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\mathbf{R}$ alone.

2.1 Conditions for stability

For the simple birth-death process describing an $M/M/1$ queue, the stability condition is simply that the arrival rate λ must be less than the service rate μ. The condition for the existence of a stationary solution for a QBD process is analogous: from some level j onward, the drift toward the left, $\boldsymbol{\pi}^{(j)} \cdot \mathbf{B} \cdot \mathbf{1}^T$, must be greater than the drift toward the right, $\boldsymbol{\pi}^{(j)} \cdot \mathbf{F} \cdot \mathbf{1}^T$.

However, we need to check for stability before we solve the CTMC, so $\boldsymbol{\pi}^{(j)}$ is not yet known at this point. Fortunately, for a large enough j, the effect of the initial portion of the CTMC is negligible, so we can say that

$$\forall a \in \{1, \ldots, n\}, \qquad \lim_{j \to \infty} \frac{\boldsymbol{\pi}_a^{(j)}}{\boldsymbol{\pi}^{(j)} \cdot \mathbf{1}^T} = \boldsymbol{\gamma}_a,$$

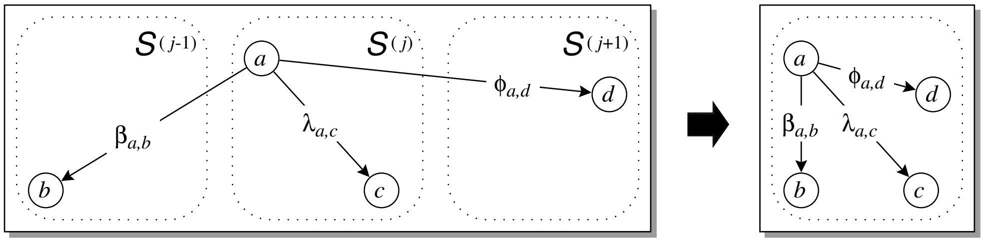

Figure 2: Testing for stability in a QBD process.

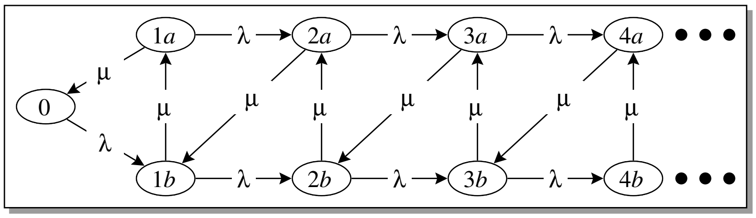

Figure 3: An example.

where $\boldsymbol{\gamma}$ is the stationary probability vector for the n-state CTMC obtained by rerouting transitions from $s_a^{(j)}$ to state $s_b^{(j-1)}$ or $s_b^{(j+1)}$ so that they become transitions to state $s_b^{(j)}$ instead (see Fig. 2). In other words, $\boldsymbol{\gamma}$ is the probability vector satisfying

$$\boldsymbol{\gamma} \cdot (\mathbf{B} + \mathbf{L} + \mathbf{F}) = \mathbf{0}$$

and can be obtained using any method ordinarily employed to compute the stationary probability vector of an ergodic CTMC.

After computing $\boldsymbol{\gamma}$, the condition for stability is then simply

$$\boldsymbol{\gamma} \cdot \mathbf{B} \cdot \mathbf{1}^T > \boldsymbol{\gamma} \cdot \mathbf{F} \cdot \mathbf{1}^T. \tag{3}$$

We stress that this discussion of stability assumes that the CTMC having infinitesimal generator $(\mathbf{B} + \mathbf{L} + \mathbf{F})$ is ergodic. If instead this CTMC contains multiple recurrent classes, we can still test for stability by checking that condition (3) holds individually for each recurrent class.

3 Motivational example

Consider the CTMC in Fig. 3, modeling an $M/E_r/1$ queue with arrival rate Expo(λ) and service distribution Erlang($2, \mu$). State 0 corresponds to an empty system, while state Nx described the system with N customers, the first one being served in phase x, for $N = 1, 2, \ldots$ and $x = a$ (second phase) or $x = b$ (first phase).

Let $\mathcal{A} = \{2a, 3a, \ldots\}$ and $\mathcal{B} = \{2b, 3b, \ldots\}$. Hence the state space can be written as $\mathcal{S} = \{0, 1a, 1b\} \cup \mathcal{A} \cup \mathcal{B}$. Instead of applying the matrix geometric method for the computation of the stationary probability vector $[\boldsymbol{\pi}_0, \boldsymbol{\pi}_{1a}, \boldsymbol{\pi}_{1b}, \ldots]$, we derive a system of linear equations in the five variables $\boldsymbol{\pi}_0$, $\boldsymbol{\pi}_{1a}$, $\boldsymbol{\pi}_{1b}$, $\boldsymbol{\pi}_{\mathcal{A}} = \sum_{i \in \mathcal{A}} \boldsymbol{\pi}_i$, and $\boldsymbol{\pi}_{\mathcal{B}} = \sum_{i \in \mathcal{S}} \boldsymbol{\pi}_i$.

One equation is given by the normalization condition

$$\boldsymbol{\pi}_0 + \boldsymbol{\pi}_{1a} + \boldsymbol{\pi}_{1b} + \boldsymbol{\pi}_{\mathcal{A}} + \boldsymbol{\pi}_{\mathcal{B}} = 1, \tag{4}$$

so we need to find four more (linearly-independent) equations. We do this by expressing the flow balance for various sets of states.

We can express the flow balance for $\{0\}$:

$$\boldsymbol{\pi}_0 \lambda = \boldsymbol{\pi}_{1a} \mu, \tag{5}$$

for $\{1a\}$:

$$\boldsymbol{\pi}_{1a}(\lambda + \mu) = \boldsymbol{\pi}_{1b} \mu, \tag{6}$$

and for each state in $\mathcal{A}$ as:

$$\forall i \geq 2, \qquad \boldsymbol{\pi}_{ia}(\lambda + \mu) = \boldsymbol{\pi}_{(i-1)a} \lambda + \boldsymbol{\pi}_{ib} \mu. \tag{7}$$

Summing by parts Eq. 7 over all $i \geq 2$ we derive:

$$\boldsymbol{\pi}_{\mathcal{A}} \mu = \boldsymbol{\pi}_{1a} \lambda + \boldsymbol{\pi}_{\mathcal{B}} \mu. \tag{8}$$

which expresses the flow balance for $\mathcal{A}$. If we try to express the flow balance for $\mathcal{B}$ in a similar manner (i.e., by first expressing the flow balance of each state in $\mathcal{B}$ and then summing the resulting equations by parts), we realize that we must introduce the probability of state $2a$ explicitly:

$$\boldsymbol{\pi}_{\mathcal{B}} \mu = \boldsymbol{\pi}_{1b} \lambda + (\boldsymbol{\pi}_{\mathcal{A}} - \boldsymbol{\pi}_{2a}) \mu,$$

hence we cannot use this equation, as it force us to increase the number of unknowns (and to look for yet another equation, and so on). We could also express the flow balance for $\{1b\} \cup \mathcal{S}$,

$$(\boldsymbol{\pi}_{1b} + \boldsymbol{\pi}_{\mathcal{B}}) \mu = \boldsymbol{\pi}_0 \lambda + \boldsymbol{\pi}_{\mathcal{A}} \mu,$$

but this equation is a linear combination of equations (5), (6), and (8), so it would not add any new information either.

To derive a fifth linearly-independent equation, we then consider the flow balance between $\{1a, 1b\}$ and $\{2a, 2b\}$,

$$\boldsymbol{\pi}_{1a} \lambda + \boldsymbol{\pi}_{1b} \lambda = \boldsymbol{\pi}_{2a} \mu,$$

the flow balance between $\{2a, 2b\}$ and $\{3a, 3b\}$,

$$\boldsymbol{\pi}_{2a} \lambda + \boldsymbol{\pi}_{2b} \lambda = \boldsymbol{\pi}_{3a} \mu$$

and so on, and sum all these equations, obtaining:

$$(\boldsymbol{\pi}_{1a} + \boldsymbol{\pi}_{\mathcal{A}}) \lambda + (\boldsymbol{\pi}_{1b} + \boldsymbol{\pi}_{\mathcal{B}}) \lambda = \boldsymbol{\pi}_{\mathcal{A}} \mu. \tag{9}$$

Equations (4), (5), (6), (8), and (9) are linearly independent and can then be used to compute our five unknowns, by solving the linear system

$$[\boldsymbol{\pi}_0, \boldsymbol{\pi}_{1a}, \boldsymbol{\pi}_{1b}, \boldsymbol{\pi}_{\mathcal{A}}, \boldsymbol{\pi}_{\mathcal{B}}] \cdot \begin{bmatrix} 1 & -\lambda & 0 & 0 & 0 \\ 1 & \mu & -(\lambda + \mu) & \lambda & \lambda \\ 1 & 0 & \mu & 0 & \lambda \\ 1 & 0 & 0 & -\mu & \lambda - \mu \\ 1 & 0 & 0 & \mu & \lambda \end{bmatrix} = [1, 0, 0, 0, 0],$$

resulting in

$$[\boldsymbol{\pi}_0, \boldsymbol{\pi}_{1a}, \boldsymbol{\pi}_{1b}, \boldsymbol{\pi}_{\mathcal{A}}, \boldsymbol{\pi}_{\mathcal{B}}] = \left[\frac{\mu - 2\lambda}{\mu}, \frac{\lambda(\mu - 2\lambda)}{\mu^2}, \frac{\lambda(\lambda+\mu)(\mu - 2\lambda)}{\mu^3}, \frac{2\lambda^2}{\mu^2}, \frac{\lambda^2(\mu + 2\lambda)}{\mu^3}\right].$$

To compute a performance measure such as queue length we follow similar steps. Let $q_{\mathcal{A}} = \sum_{i=2}^{\infty} i\boldsymbol{\pi}_{ia}$ and $q_{\mathcal{B}} = \sum_{i=2}^{\infty} i\boldsymbol{\pi}_{ib}$ be the contribution to the expected queue length due to the states in $\mathcal{A}$ and $\mathcal{B}$, respectively. To compute $q_{\mathcal{A}}$, we multiply the equations that express the flow balance for each state in $\mathcal{A}$ by the appropriate factor:

$$\forall i \geq 2, \qquad i\boldsymbol{\pi}_{ia}(\lambda + \mu) = i\boldsymbol{\pi}_{(i-1)a}\lambda + i\boldsymbol{\pi}_{ib}\mu \tag{10}$$

and by summing by parts Eq. (10) over all $i \geq 2$ we derive:

$$q_{\mathcal{A}}(\lambda + \mu) = 2\boldsymbol{\pi}_{1a}\lambda + (q_{\mathcal{A}} + \boldsymbol{\pi}_{\mathcal{A}})\lambda + q_{\mathcal{B}}\mu. \tag{11}$$

We continue by multiplying by the appropriate factor the set of equations that express the flow balance between $\{ia, ib\}$ and $\{(i+1)a, (i+1)b\}$, for $i \geq 1$:

$$(i+1)\boldsymbol{\pi}_{ia}\lambda + (i+1)\boldsymbol{\pi}_{ib}\lambda = (i+1)\boldsymbol{\pi}_{(i+1)a}\mu, \tag{12}$$

and, after summing by parts Eq. (12) over all $i \geq 1$, we obtain:

$$2\boldsymbol{\pi}_{1a}\lambda + 2\boldsymbol{\pi}_{1b}\lambda + (q_{\mathcal{A}} + \boldsymbol{\pi}_{\mathcal{A}})\lambda + (q_{\mathcal{B}} + \boldsymbol{\pi}_{\mathcal{B}})\lambda = q_{\mathcal{A}}\lambda. \tag{13}$$

Since Equations (11) and (13) are linearly independent, we can express $q_{\mathcal{A}}$ and $q_{\mathcal{B}}$ as a function of λ, μ, plus the quantities $\boldsymbol{\pi}_{1a}$, $\boldsymbol{\pi}_{1b}$, $\boldsymbol{\pi}_{\mathcal{A}}$, and $\boldsymbol{\pi}_{\mathcal{B}}$, which are already known. We then obtain the expected queue length as

$$\boldsymbol{\pi}_{1a} + \boldsymbol{\pi}_{1b} + q_{\mathcal{A}} + q_{\mathcal{B}}.$$

4 Formalization

We now formalize the ideas used in the previous section and show how to apply them to a specific class of CTMCs. We consider CTMCs describing QBD processes having an infinitesimal generator matrix with the following structure:

$$\mathbf{Q} = \left[\begin{array}{c|c|c|c|c|c} \mathbf{L}^{(00)} & \mathbf{F}^{(01)} & & & & \\ \hline \mathbf{B}^{(10)} & \mathbf{L}^{(11)} & \mathbf{F} & & & \\ \hline & \mathbf{B} & \mathbf{L} & \mathbf{F} & & \\ \hline & & \mathbf{B} & \mathbf{L} & \mathbf{F} & \\ \hline & & & \cdots & \cdots & \cdots \end{array}\right]. \tag{14}$$

Note that $\mathbf{L}^{(11)}$ is not required to equal $\mathbf{L}$; while they are usually the same (except possibly in the diagonal, since $\mathbf{Q}$ is an infinitesimal generator and $\mathbf{B}^{(10)} \cdot \mathbf{1}^T$ could differ from $\mathbf{B} \cdot \mathbf{1}^T$), this is not required by our approach.

The stationary probability vector satisfying $\boldsymbol{\pi} \cdot \mathbf{Q} = \mathbf{0}$ can be partitioned accordingly into

$$\boldsymbol{\pi} = \left[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(2)}, \cdots\right]$$

with $\boldsymbol{\pi}^{(0)} \in \mathbb{R}^m$ and $\boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(2)}, \cdots \in \mathbb{R}^n$.

Then, we can write $\boldsymbol{\pi} \cdot \mathbf{Q} = \mathbf{0}$ as:

$$\begin{cases} & & \boldsymbol{\pi}^{(0)} \cdot \mathbf{L}^{(00)} & + & \boldsymbol{\pi}^{(1)} \cdot \mathbf{B}^{(10)} & = & \mathbf{0} \\ \boldsymbol{\pi}^{(0)} \cdot \mathbf{F}^{(01)} & + & \boldsymbol{\pi}^{(1)} \cdot \mathbf{L}^{(11)} & + & \boldsymbol{\pi}^{(2)} \cdot \mathbf{B} & = & \mathbf{0} \\ \boldsymbol{\pi}^{(1)} \cdot \mathbf{F} & + & \boldsymbol{\pi}^{(2)} \cdot \mathbf{L} & + & \boldsymbol{\pi}^{(3)} \cdot \mathbf{B} & = & \mathbf{0} \\ \boldsymbol{\pi}^{(2)} \cdot \mathbf{F} & + & \boldsymbol{\pi}^{(3)} \cdot \mathbf{L} & + & \boldsymbol{\pi}^{(4)} \cdot \mathbf{B} & = & \mathbf{0} \\ & & \cdots & & & & \end{cases} . \tag{15}$$

Now, assume that $\mathbf{B}$ is a matrix of zeros except for the last column (i.e., all backward transitions from states in $\mathcal{S}^{(j)}$ to $\mathcal{S}^{(j-1)}$ are restricted to go to $s_n^{(j-1)}$, the last state in $\mathcal{S}^{(j-1)}$). Let $\mathbf{A}_{r_1:r_2,c_1:c_2}$ be the submatrix of $\mathbf{A}$ corresponding to the rows from r_1 to r_2 and the columns from c_1 to c_2.

We are going to show how to derive $m+2n$ equations in $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and a new vector of n unknowns, $\boldsymbol{\pi}^{(*)} = \sum_{i=2}^{\infty} \boldsymbol{\pi}^{(i)}$, where $\boldsymbol{\pi}_j^{(*)}$ represents the stationary probability of being in the macro-state $\mathcal{T}_j = \{s_j^{(i)} : i \geq 2\}$.

- The normalization constraints offers one equation:

$$\boldsymbol{\pi}^{(0)} \cdot \mathbf{1}^T + \boldsymbol{\pi}^{(1)} \cdot \mathbf{1}^T + \boldsymbol{\pi}^{(*)} \cdot \mathbf{1}^T = 1. \tag{16}$$

- The first row in (15) provides m equations:

$$\boldsymbol{\pi}^{(0)} \cdot \mathbf{L}^{(00)} + \boldsymbol{\pi}^{(1)} \cdot \mathbf{B}^{(10)} = \mathbf{0}. \tag{17}$$

- Due to the structure of $\mathbf{B}$, we can observe that the first $n-1$ entries of $\boldsymbol{\pi}^{(2)} \cdot \mathbf{B}$ are null, hence the second row in (15) provides $n-1$ equations:

$$\boldsymbol{\pi}^{(0)} \cdot \mathbf{F}^{(01)}_{1:m,1:n-1} + \boldsymbol{\pi}^{(1)} \cdot \mathbf{L}^{(11)}_{1:n,1:n-1} = \mathbf{0}. \tag{18}$$

- If we sum all the remaining equations in (15) we obtain

$$\sum_{i=1}^{\infty} \boldsymbol{\pi}^{(i)} \cdot \mathbf{F} + \sum_{i=2}^{\infty} \boldsymbol{\pi}^{(i)} \cdot \mathbf{L} + \sum_{i=3}^{\infty} \boldsymbol{\pi}^{(i)} \cdot \mathbf{B} = \mathbf{0},$$

  which, again by using the structure of $\mathbf{B}$, results in another $n-1$ equations:

$$\boldsymbol{\pi}^{(1)} \cdot \mathbf{F}_{1:n,1:n-1} + \boldsymbol{\pi}^{(*)} \cdot (\mathbf{F} + \mathbf{L})_{1:n,1:n-1} = \mathbf{0}. \tag{19}$$

- Finally, we can consider the equations describing the balance of flow between successive sets of states, $\mathcal{S}^{(j)}$ and $\mathcal{S}^{(j+1)}$, for $j \geq 1$:

$$\begin{cases} \boldsymbol{\pi}^{(1)} \cdot \mathbf{F} \cdot \mathbf{1}^T & - & \boldsymbol{\pi}^{(2)} \cdot \mathbf{B} \cdot \mathbf{1}^T & = & 0 \\ \boldsymbol{\pi}^{(2)} \cdot \mathbf{F} \cdot \mathbf{1}^T & - & \boldsymbol{\pi}^{(3)} \cdot \mathbf{B} \cdot \mathbf{1}^T & = & 0 \\ \boldsymbol{\pi}^{(3)} \cdot \mathbf{F} \cdot \mathbf{1}^T & - & \boldsymbol{\pi}^{(4)} \cdot \mathbf{B} \cdot \mathbf{1}^T & = & 0 \\ & & \cdots & & \end{cases} . \tag{20}$$

  By summing all these equations we obtain:

$$\boldsymbol{\pi}^{(1)} \cdot \mathbf{F} \cdot \mathbf{1}^T + \boldsymbol{\pi}^{(*)} \cdot (\mathbf{F} - \mathbf{B}) \cdot \mathbf{1}^T = 0. \tag{21}$$

The following theorem states that these equations are sufficient to compute $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$.

Theorem 4.1 Given an ergodic CTMC with infinitesimal generator $\mathbf{Q}$ having the structure shown in (14) and such that the first $n-1$ columns of $\mathbf{B}$ are null, $\mathbf{B}_{1:n,1:n-1} = \mathbf{0}$, the system of linear equations

$$[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(*)}] \cdot \mathbf{X} = [1, \mathbf{0}], \tag{22}$$

with $\mathbf{X}$ defined as

$$\mathbf{X} = \begin{bmatrix} \mathbf{1}^T & \mathbf{L}^{(00)} & \mathbf{F}^{(01)}_{1:m,1:n-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1}^T & \mathbf{B}^{(10)} & \mathbf{L}^{(11)}_{1:n,1:n-1} & \mathbf{F}_{1:n,1:n-1} & \mathbf{F}\cdot\mathbf{1}^T \\ \mathbf{1}^T & \mathbf{0} & \mathbf{0} & (\mathbf{F}+\mathbf{L})_{1:n,1:n-1} & (\mathbf{F}-\mathbf{B})\cdot\mathbf{1}^T \end{bmatrix}, \tag{23}$$

admits a unique solution satisfying $\boldsymbol{\pi}^{(*)} = \sum_{i=2}^{\infty} \boldsymbol{\pi}^{(i)}$, where $[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(2)}, \ldots]$ is the stationary probability vector of the CTMC.

Proof It is easy to show that (22) admits at least one solution. This is because, if the original CTMC is ergodic, the equation $\boldsymbol{\pi} \cdot \mathbf{Q} = \mathbf{0}$ subject to $\boldsymbol{\pi} \cdot \mathbf{1}^T = 1$ admits a unique solution $\hat{\boldsymbol{\pi}}$, which we can partition as $[\hat{\boldsymbol{\pi}}^{(0)}, \hat{\boldsymbol{\pi}}^{(1)}, \hat{\boldsymbol{\pi}}^{(2)}, \ldots]$. Clearly, the vector $[\hat{\boldsymbol{\pi}}^{(0)}, \hat{\boldsymbol{\pi}}^{(1)}, \sum_{j=2}^{\infty} \hat{\boldsymbol{\pi}}^{(j)}]$ satisfies (16), (17), (18), (19), and (21), hence it is a solution of (22).

To show that the solution of (22) is unique, however, we must also show that the rank of $\mathbf{X}$ is $m+2n$. We do this by showing that its rows are $m+2n$ linearly independent vectors. It is well-known that, since $\mathbf{Q}$ is ergodic, the vector $\mathbf{1}^T$ plus and set of vectors corresponding to all the columns of $\mathbf{Q}$ except one (any one of them) are linearly independent. In particular, we choose to remove from $\mathbf{Q}$ the n^{th} column of the second block of columns in (14), corresponding to transitions into state s^1_n. In our case, the result is the countably infinite set of vectors $\mathbf{v}^{[1]}, \mathbf{v}^{[2]}, \ldots$ in $\mathbb{R}^{\mathbb{N}}$ shown in Fig. 4.

We now define n new vectors $\mathbf{z}^{[1]}$ through $\mathbf{z}^{[n]}$ in $\mathbb{R}^{\mathbb{N}}$, also shown in Fig. 4, as follows:

- For $i = 1, \ldots, n-1$, let $\mathbf{z}^{[i]} = \sum_{j=1}^{\infty} \mathbf{v}^{[m+jn+i]}$, that is, we sum the i^{th} column from each block of columns starting from the block corresponding to transitions into $\mathcal{S}^{(2)}$. Note that $\mathbf{B}$ does not appear in the expression of these vectors, because $\mathbf{B}_{1:n,1:n-1} = \mathbf{0}$.
- For the last vector, $\mathbf{z}^{[n]} = \sum_{j=1}^{\infty} \sum_{i=1}^{n} \sum_{k=j}^{\infty} \mathbf{v}^{[m+kn+i]}$. To justify the expression for $\mathbf{z}^{[m+2n]}$, it is convenient to consider its derivation in steps: $\mathbf{z}^{[n]} = \sum_{j=1}^{\infty} \mathbf{y}^{[j]}$, where $\mathbf{y}^{[j]} = \sum_{i=1}^{n} \mathbf{x}^{[(j-1)n+i]}$, and $\mathbf{x}^{[(j-1)n+i]} = \sum_{k=j}^{\infty} \mathbf{v}^{[m+kn+i]}$. Note that $(\mathbf{B}+\mathbf{L}+\mathbf{F})\cdot\mathbf{1}^T = \mathbf{0}$, because $\mathbf{Q}$ is an infinitesimal generator, and this also implies that $(\mathbf{L}+\mathbf{F})\cdot\mathbf{1}^T = -\mathbf{B}\cdot\mathbf{1}^T$.

It is then easy to see that the $m+2n$ vectors $\{\mathbf{v}^{[1]}, \ldots, \mathbf{v}^{[m+n]}, \mathbf{z}^{[1]}, \ldots, \mathbf{z}^{[n]}\}$ are linearly independent since (a) the original set $\{\mathbf{v}^{[1]}, \mathbf{v}^{[2]}, \ldots\}$ is linearly independent, (b) $\{\mathbf{z}^{[1]}, \ldots, \mathbf{z}^{[n]}\}$ are obtained as linear combinations of different subsets of vectors from $\{\mathbf{v}^{[m+n+1]}, \mathbf{v}^{[m+n+2]}, \ldots\}$, (c) disjoint subsets of vectors are used to build $\{\mathbf{z}^{[1]}, \ldots, \mathbf{z}^{[n-1]}\}$, and, finally (d) $\mathbf{z}^{[n]}$ is built using vectors already used for $\{\mathbf{z}^{[1]}, \ldots, \mathbf{z}^{[n-1]}\}$, but also vectors of the form $\mathbf{v}^{[m+jn]}$, which are not used to build any of the vectors in $\{\mathbf{z}^{[1]}, \ldots, \mathbf{z}^{[n-1]}\}$.

Thus, the rank of the matrix $[\mathbf{v}^{[1]}, \ldots, \mathbf{v}^{[m+n]}, \mathbf{z}^{[1]}, \ldots, \mathbf{z}^{[n]}]$ is $m+2n$, which implies we must be able to find $m+2n$ linearly independent rows in it. Since row $m+jn+i$ is the same as row $m+n+i$, for $i = 1, \ldots, n$ and $j \geq 2$, it is clear that the first $m+2n$ rows are linearly independent. These are the rows of our matrix $\mathbf{X}$, hence the proof is complete. $\square$

$\mathbf{v}^{[1]}$	$\mathbf{v}^{[2]}$ through $\mathbf{v}^{[m+1]}$	$\mathbf{v}^{[m+2]}$ through $\mathbf{v}^{[m+n]}$	$\mathbf{v}^{[m+n+1]}$ through $\mathbf{v}^{[m+2n]}$	$\mathbf{v}^{[m+2n+1]}$ through $\mathbf{v}^{[m+3n]}$	$\mathbf{v}^{[m+3n+1]}$ through $\mathbf{v}^{[m+4n]}$	$\cdots$
$\mathbf{1}^T$	$\mathbf{L}^{(00)}$	$\mathbf{F}^{(01)}_{1:m,1:n-1}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\cdots$
$\mathbf{1}^T$	$\mathbf{B}^{(10)}$	$\mathbf{L}^{(11)}_{1:n,1:n-1}$	$\mathbf{F}$	$\mathbf{0}$	$\mathbf{0}$	$\cdots$
$\mathbf{1}^T$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{L}$	$\mathbf{F}$	$\mathbf{0}$	$\cdots$
$\mathbf{1}^T$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{B}$	$\mathbf{L}$	$\mathbf{F}$	$\cdots$
$\mathbf{1}^T$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{B}$	$\mathbf{L}$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$

$\mathbf{z}^{[1]}$ through $\mathbf{z}^{[n-1]}$
$\mathbf{0}$
$\mathbf{F}_{1:n,1:n-1}$
$(\mathbf{F}+\mathbf{L})_{1:n,1:n-1}$
$(\mathbf{F}+\mathbf{L})_{1:n,1:n-1}$
$(\mathbf{F}+\mathbf{L})_{1:n,1:n-1}$
$\vdots$

$\mathbf{x}^{[1]}$ through $\mathbf{x}^{[n]}$	$\mathbf{x}^{[n+1]}$ through $\mathbf{x}^{[2n]}$	$\cdots$
$\mathbf{0}$	$\mathbf{0}$	$\cdots$
$\mathbf{F}$	$\mathbf{0}$	$\cdots$
$\mathbf{L}+\mathbf{F}$	$\mathbf{F}$	$\cdots$
$\mathbf{B}+\mathbf{L}+\mathbf{F}$	$\mathbf{L}+\mathbf{F}$	$\cdots$
$\mathbf{B}+\mathbf{L}+\mathbf{F}$	$\mathbf{B}+\mathbf{L}+\mathbf{F}$	$\cdots$
$\vdots$	$\vdots$	$\cdots$

$\mathbf{y}^{[1]}$	$\mathbf{y}^{[2]}$	$\cdots$
$\mathbf{0}$	$\mathbf{0}$	$\cdots$
$\mathbf{F}\cdot\mathbf{1}^T$	$\mathbf{0}$	$\cdots$
$(\mathbf{L}+\mathbf{F})\cdot\mathbf{1}^T$	$\mathbf{F}\cdot\mathbf{1}^T$	$\cdots$
$\mathbf{0}$	$(\mathbf{L}+\mathbf{F})\cdot\mathbf{1}^T$	$\cdots$
$\mathbf{0}$	$\mathbf{0}$	$\cdots$
$\vdots$	$\vdots$	$\cdots$

$\mathbf{z}^{[n]}$
$\mathbf{0}$
$\mathbf{F}\cdot\mathbf{1}^T$
$(\mathbf{F}-\mathbf{B})\cdot\mathbf{1}^T$
$(\mathbf{F}-\mathbf{B})\cdot\mathbf{1}^T$
$(\mathbf{F}-\mathbf{B})\cdot\mathbf{1}^T$
$\vdots$

Figure 4: The column vectors used in our proof.

It is worth noting that, even if Eq. 22 resembles the classical equation obtained removing one equation from $\boldsymbol{\pi}\cdot\mathbf{Q}=\mathbf{0}$ and adding the normalization condition $\sum\boldsymbol{\pi}_i=1$ for a finite CTMC, the nature of the matrix $\mathbf{X}$ defined by our theorem is very different. For example, simply consider the matrix $\mathbf{X}$ in Section 3, where, since $\mu>\lambda$, there are two negative entries in the row corresponding to $\mathcal{A}$.

5 Computing the measures of interest

The goal of a modeling study is to compute measures of interest. In general, a measure r is expressed as a weighted sum of the stationary state probabilities. For example, if we want to compute the expected queue length in steady state for a model where $\mathcal{S}^{(j)}$ contains the system states with j customers in the queue,

$$r=\sum_{j=1}^{\infty} j \sum_{i\in\mathcal{S}^{(j)}} \boldsymbol{\pi}_i^{(j)}.$$

However, our solution approach computes $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$, hence we do not explicitly know the individual values for $\{\boldsymbol{\pi}^{(2)},\boldsymbol{\pi}^{(3)},\ldots\}$. It is then essential to determine under what circumstances the desired measures can nevertheless be obtained. By writing r as

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

(where $\boldsymbol{\rho} = [\boldsymbol{\rho}^{(0)}, \boldsymbol{\rho}^{(1)}, \cdots]$ are the *reward rates* for the states in $\mathcal{S}^{(0)}, \mathcal{S}^{(1)}, \cdots$), it is clear that the definition of $\boldsymbol{\rho}$ is only restricted by our need to compute the above summation. We now show how to do this when 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 $\mathbf{a}_i^{[0]}, \mathbf{a}_i^{[1]}, \ldots, \mathbf{a}_i^{[k]}$:

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

In this case, then,

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

where $\mathbf{r}^{[l]} = \sum_{j=2}^{\infty} j^l \boldsymbol{\pi}^{(j)}$ for $l = 0, \ldots, k$, and its computation can be illustrated by strong induction. For the base case, $\mathbf{r}^{[0]}$ is simply $\boldsymbol{\pi}^{(*)}$. Now, assuming that we know how to compute $\mathbf{r}^{[l]}$ for $0 \leq l \leq k-1$, we show how to compute $\mathbf{r}^{[k]}$. Multiplying the equations in (15) from the second block on by the appropriate factor j^k results in

$$\begin{cases} 2^k \boldsymbol{\pi}^{(0)} \cdot \mathbf{F}^{(01)} + 2^k \boldsymbol{\pi}^{(1)} \cdot \mathbf{L}^{(11)} + 2^k \boldsymbol{\pi}^{(2)} \cdot \mathbf{B} = \mathbf{0} \\ 3^k \boldsymbol{\pi}^{(1)} \cdot \mathbf{F} + 3^k \boldsymbol{\pi}^{(2)} \cdot \mathbf{L} + 3^k \boldsymbol{\pi}^{(3)} \cdot \mathbf{B} = \mathbf{0} \\ 4^k \boldsymbol{\pi}^{(2)} \cdot \mathbf{F} + 4^k \boldsymbol{\pi}^{(3)} \cdot \mathbf{L} + 4^k \boldsymbol{\pi}^{(4)} \cdot \mathbf{B} = \mathbf{0} \\ \cdots \end{cases},$$

and summing them we obtain

$$2^k \boldsymbol{\pi}^{(0)} \cdot \mathbf{F}^{(01)} + 2^k \boldsymbol{\pi}^{(1)} \cdot \mathbf{L}^{(11)} + 3^k \boldsymbol{\pi}^{(1)} \cdot \mathbf{F} + \sum_{j=2}^{\infty} \left[(j+2)^k \boldsymbol{\pi}^{(j)} \cdot \mathbf{F} + (j+1)^k \boldsymbol{\pi}^{(j)} \cdot \mathbf{L} + j^k \boldsymbol{\pi}^{(j)} \cdot \mathbf{B}\right] = 0.$$

which, observing that

$$\sum_{j=2}^{\infty} (j+2)^k \boldsymbol{\pi}^{(j)} = \sum_{l=0}^{k} \binom{k}{l} 2^l \mathbf{r}^{[k-l]}, \quad \sum_{j=2}^{\infty} (j+1)^k \boldsymbol{\pi}^{(j)} = \sum_{l=0}^{k} \binom{k}{l} \mathbf{r}^{[k-l]}, \text{ and } \sum_{j=2}^{\infty} j^k \boldsymbol{\pi}^{(j)} = \mathbf{r}^{[k]},$$

can be expressed as $\mathbf{r}^{[k]} \cdot (\mathbf{F} + \mathbf{L} + \mathbf{B}) = \mathbf{b}$ where the constant right-hand side vector is

$$\mathbf{b} = -2^k \boldsymbol{\pi}^{(0)} \cdot \mathbf{F}^{(01)} - 2^k \boldsymbol{\pi}^{(1)} \cdot \mathbf{L}^{(11)} - 3^k \boldsymbol{\pi}^{(1)} \cdot \mathbf{F} - \sum_{l=1}^{k} \binom{k}{l} \left(2^l \mathbf{r}^{[k-l]} \cdot \mathbf{F} + \mathbf{r}^{[k-l]} \cdot \mathbf{L}\right).$$

Since the rank of $\mathbf{F} + \mathbf{L} + \mathbf{B}$ is $n-1$, we can remove the equation corresponding to the last column, resulting in the $n-1$ equations

$$\mathbf{r}^{[k]} \cdot (\mathbf{L} + \mathbf{F})_{1:n,1:n-1} = \mathbf{b}_{1:n-1}. \tag{24}$$

One additional equation is then required. We obtain it from the equations in (20), again by multiplying them by j^k,

$$\begin{cases} 2^k \boldsymbol{\pi}^{(1)} \cdot \mathbf{F} \cdot \mathbf{1}^T - 2^k \boldsymbol{\pi}^{(2)} \cdot \mathbf{B} \cdot \mathbf{1}^T = 0 \\ 3^k \boldsymbol{\pi}^{(2)} \cdot \mathbf{F} \cdot \mathbf{1}^T - 3^k \boldsymbol{\pi}^{(3)} \cdot \mathbf{B} \cdot \mathbf{1}^T = 0 \\ 4^k \boldsymbol{\pi}^{(3)} \cdot \mathbf{F} \cdot \mathbf{1}^T - 4^k \boldsymbol{\pi}^{(4)} \cdot \mathbf{B} \cdot \mathbf{1}^T = 0 \\ \cdots \end{cases},$$

and summing them, resulting in

$$2^k \boldsymbol{\pi}^{(1)} \cdot \mathbf{F} \cdot \mathbf{1}^T - \sum_{j=2}^{\infty} j^k \boldsymbol{\pi}^{(2)} \cdot \mathbf{B} \cdot \mathbf{1}^T + \sum_{j=2}^{\infty} (j+1)^k \boldsymbol{\pi}^{(2)} \cdot \mathbf{F} \cdot \mathbf{1}^T = 0,$$

which can be written as

$$\mathbf{r}^{[k]} \cdot (\mathbf{F} - \mathbf{B}) \cdot \mathbf{1}^T = c \tag{25}$$

where

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

We can then observe that Eq. (24) and (25) together are sufficient to uniquely determine the value of $\mathbf{r}^{[k]}$, since they define the linear system $\mathbf{r}^{[k]} \cdot \left[(\mathbf{L}+\mathbf{F})_{1:n,1:n-1} | (\mathbf{F}-\mathbf{B}) \cdot \mathbf{1}^T\right] = [\mathbf{b}_{1:n-1}|c]$, and the matrix $\left[(\mathbf{L}+\mathbf{F})_{1:n,1:n-1} | (\mathbf{F}-\mathbf{B}) \cdot \mathbf{1}^T\right]$ has been shown to have full rank already, in the proof of Theorem 4.1. Thus, in conclusion, if the (maximum) degree of the polynomials involved in the specification of the rewards is k, we need to solve k times the non-homogeneous linear system with coefficient matrix $\left[(\mathbf{L}+\mathbf{F})_{1:n,1:n-1} | (\mathbf{F}-\mathbf{B}) \cdot \mathbf{1}^T\right]$, for k different right-hand sides. In particular, the most common and important case of linear reward rates (as required to compute the expected queue lengths) requires the solution of only one linear system of size $n \times n$ in addition to the solution of the $(m+2n) \times (m+2n)$ system required to compute the stationary probabilities.

6 Extending the applicability by state repartitioning

Theorem 4.1 might seem restrictive, since it requires that any backward transition from $\mathcal{S}^{(j)}$ must be directed to "the last state" of $\mathcal{S}^{(j-1)}$, state $s_n^{(j-1)}$. Clearly, given that no restriction on $\mathbf{L}$ and $\mathbf{F}$ exist, an appropriate ordering of the states in $\mathcal{S}^{(j)}$ can always be found to satisfy this condition, as long as all backward transitions go a a single state, which can then be numbered last. But the applicability of this theorem is much wider than this, because in many cases we can change the state partitioning, so that a matrix that does not satisfy the condition of our theorem can be rearranged into one that does.

Consider for example a system with Poisson arrivals with rate λ and a server that needs rest to perform well (Fig. 5 on the top). After an idle period, the server is well rested and starts processing the first arrival at a rate μ_a. However, the server tires easily, so the second arrival is processed at a rate $\mu_b < \mu_a$, the third one at a rate $\mu_c < \mu_b$, and all the subsequent arrivals are processed at a rate $\mu_d < \mu_c$, until the idle state is reached again.

The natural partition into states shown in the grey blocks, corresponding to the population in the system, $\mathcal{S}^{(0)} = \{0\}$, $\mathcal{S}^{(1)} = \{1a, 1b, 1c, 1d\}$, $\mathcal{S}^{(2)} = \{2a, 2b, 2c, 2d\}$, and so on, does not satisfy the theorem assumptions:

$$\mathbf{B}' = \begin{bmatrix} 0 & \mu_a & 0 & 0 \\ 0 & 0 & \mu_b & 0 \\ 0 & 0 & 0 & \mu_c \\ 0 & 0 & 0 & \mu_d \end{bmatrix} \qquad \mathbf{L}' = \begin{bmatrix} -\lambda-\mu_a & 0 & 0 & 0 \\ 0 & -\lambda-\mu_b & 0 & 0 \\ 0 & 0 & -\lambda-\mu_c & 0 \\ 0 & 0 & 0 & -\lambda-\mu_d \end{bmatrix} \qquad \mathbf{F}' = \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda \end{bmatrix}.$$

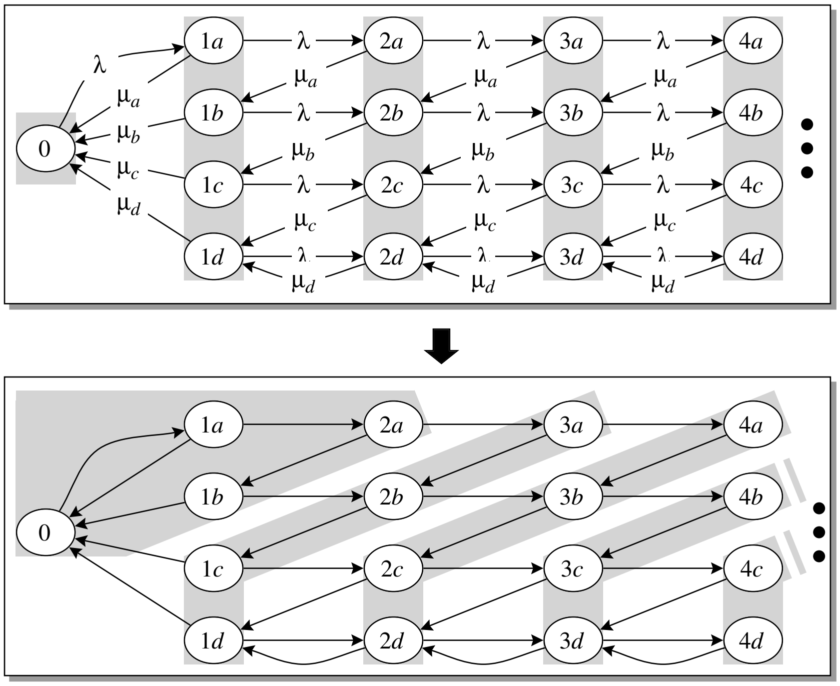

Figure 5: Changing the state partition.

However, for the alternative partition shown in the same figure at the bottom, $\mathcal{S}^{(0)} = \{0, 1a, 1b, 2a\}$, $\mathcal{S}^{(1)} = \{3a, 2b, 1c, 1d\}$, $\mathcal{S}^{(2)} = \{4a, 3b, 2c, 2d\}$, and so on,

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu_c \\ 0 & 0 & 0 & \mu_d \end{bmatrix} \qquad \mathbf{L}' = \begin{bmatrix} -\lambda-\mu_a & \mu_a & 0 & 0 \\ 0 & -\lambda-\mu_b & \mu_b & 0 \\ 0 & 0 & -\lambda-\mu_c & 0 \\ 0 & 0 & 0 & -\lambda-\mu_d \end{bmatrix} \qquad \mathbf{F} = \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda \end{bmatrix},$$

hence our theorem can now be applied. The price paid for this is an increase in the size of the linear system, since, in the second case $\mathcal{S}^{(0)} = \{0, 1a, 1b, 2a\}$, instead of simply $\{0\}$. For this application, if the server has n levels of service, $|\mathcal{S}^{(j)}| = n$ and $\sum_{i=1}^{n-2} i = (n^2 - 3n + 2)/2$ states must be moved into $\mathcal{S}^{(0)}$. Note that the repartitioning effectively moves selected entries from $\mathbf{B}$ to $\mathbf{L}$, that is, some backward transitions become local transitions. It is also possible for entries to move from $\mathbf{L}$ to $\mathbf{F}$, that is local transitions become forward transitions. It is also possible that such a transformation results in a $M/G/1$-type matrix (with finite-size forward jumps) even if we start from a QBD-type matrix. See for example Fig. 6, where the transitions from ja to $(j+1)b$ correspond to a single forward step with the partition $\mathcal{S}^{(j)} = \{ja, jb, jc\}$, but to a double forward step with the partition $\mathcal{S}^{(j)} = \{(j+1)a, jb, jc\}$, which is required in order to obtain a $\mathbf{B}$ matrix with a single nonzero column.

The presence of multiple forward jumps can be managed in two ways. One way would be to extend our main theorem, deriving a similar result for $M/G/1$-type matrices, as long as the backward transitions from $\mathcal{S}^{(j)}$ are still to a single state of $\mathcal{S}^{(j-1)}$. If we think of multiple forward jumps as bulk arrivals in the queue, and if the size b of the bulk is bounded, another way of dealing with this type of matrix is simply to redefine the partitioning of the state space, so that the sets $\mathcal{S}^{(j)}$ are grouped b at a time. For example, in Fig. 6, this would require us to define $\mathcal{S}^{(1)} = \{1b, 1c, 2a, 2b, 2c, 3a\}$, $\mathcal{S}^{(2)} = \{3b, 3c, 4a, 4b, 4c, 5a\}$, and so on. Then, any forward transition would again be single-step,

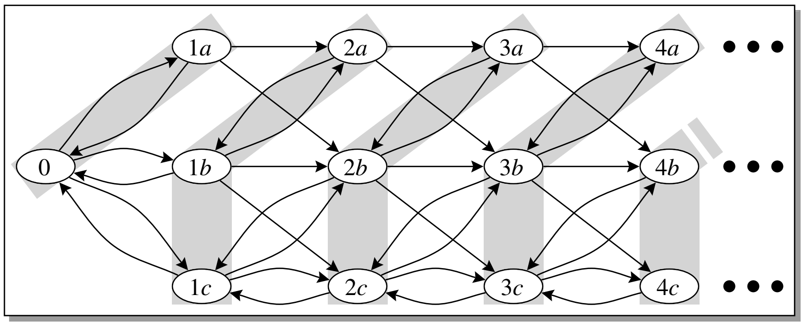

Figure 6: A multiple forward jump cause by repartitioning the states.

ETAQA	
Solution of (22)	$O(I_{ETAQA} \cdot \eta(\mathbf{X}))$
Computation of k^{th} moment	$O(k \cdot I_{measure} \cdot \eta(\mathbf{L} + \mathbf{F}))$
Matrix-geometric approach	
Computation of $\mathbf{R}$	$O(n^3 + I_{\mathbf{R}} \cdot n^2)$ or $O(n \cdot I_{ALT} \cdot \eta(\mathbf{L}))$
Solution of (2)	$O(n^2 + I_{MG} \cdot (\eta(\mathbf{L}^{(00)}) + \eta(\mathbf{B}^{(10)}) + \eta(\mathbf{F}^{(01)}) + \eta(\mathbf{L})))$
Computation of k^{th} moment	$O(k \cdot n^3)$

Table 1: Computational complexity of the MG approach and ETAQA.

and our theorem could be directly applied.

7 Computational complexity

To apply ETAQA, we must solve a linear system in $m + 2n$ unknowns described by the matrix $\mathbf{X}$ defined in Eq. (23). Then, if the rewards include a polynomial of degree k (e.g., we want to obtain the k^{th} moment of the queue length), we must solve k linear systems in n unknowns. The resulting complexity is recalled in Table 1. In it, quantities of the form I_x indicate the number of iterations required for convergence to a given tolerance in a fixed-point-type numerical method, and $\eta(\mathbf{A})$ indicates the number of nonzero entries a in matrix $\mathbf{A}$. In particular, I_{ETAQA} and $I_{measure}$ are the number of iterations required for convergence in the first linear system and in each of the k smaller linear systems (on average), respectively. We stress that the original matrices describing the QBD process are usually very sparse, but we assume they have at least one entry per row on average, hence we drop terms like $\eta(\mathbf{R} \cdot \mathbf{B})$ from any complexity expression already containing the number of nonzero in other matrices, since $\eta(\mathbf{R} \cdot \mathbf{B}) \leq n$.

7.1 The standard matrix-geometric approach

ETAQA can be applied to only a subset of the QBD processes that can be solved by the matrix-geometric approach. For example, the CTMC in Fig. 7 has a QBD structure, but it does not satisfy the requirement of a single nonzero column in $\mathbf{B}$, no matter how we partition its state space. This is easily verified by observing that the main diagonal of $\mathbf{B}$ contains two nonzero entries: clearly, repartitioning can never reduce the number of nonzero columns in $\mathbf{B}$ below two in this case.

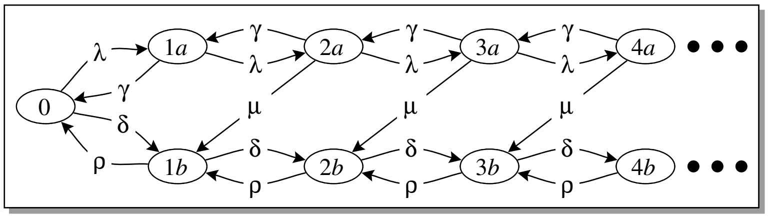

Figure 7: A CTMC not solvable by our approach.

When both ETAQA and the matrix-geometric approach are applicable, however, it makes sense to compare their efficiency in terms of memory and execution time requirements. First of all, one should wonder whether our restriction of a single nonzero column in $\mathbf{B}$ (which we numbered the n^{th} just for convenience) allows for a more efficient computation or storage of the matrix $\mathbf{R}$. We now show that this is only partially the case, thus concluding that ETAQA is much preferable, when applicable.

The interpretation of matrix $\mathbf{R}$ in the standard matrix-geometric approach is as follows [15, p. 35]:

$$\forall k, l \in \{1, \ldots, n\}, \quad \mathbf{R}_{k,l} = \frac{\hat{\mathbf{R}}_{k,l}}{-\mathbf{L}_{l,l}} \cdot (-\mathbf{L}_{k,k}),$$

where $\hat{\mathbf{R}}_{k,l}$ represents the expected number of visits to $s_l^{(j+1)}$ starting from state $s_k^{(j)}$, and before entering any state in $\mathcal{S}^{(j)}$. In other words, $\mathbf{R}_{k,l}$ represents the expected time spent in state $s_l^{(j+1)}$ after a transition out of state $s_k^{(j)}$ and before entering a state in $\mathcal{S}^{(j)}$, measured using $(-\mathbf{L}_{k,k})^{-1}$, the holding time in state $s_k^{(j)}$, as the time unit, for $j \geq 1$.

It is then immediate to see that, given this interpretation of $\mathbf{R}$, the special structure of $\mathbf{B}$ alone does not enforce any special structure in $\mathbf{R}$ itself (i.e., $\mathbf{R}$ can be completely full, and there is no simple relation between its entries, since $\mathbf{L}$ and $\mathbf{F}$ are completely general).

The structure of $\mathbf{B}$, if properly exploited in the iterative equation (1), does however reduce the computational requirements. When performing the product $\mathbf{R}^2(k) \cdot \mathbf{B} \cdot \mathbf{L}^{-1}$, we can compute it as $(\mathbf{R} \cdot (\mathbf{R} \cdot \mathbf{B})) \cdot \mathbf{L}^{-1}$, where the three products to be performed require only $O(n^2)$ operations, since $\mathbf{B}$, $(\mathbf{R} \cdot \mathbf{B})$, and $(\mathbf{R} \cdot (\mathbf{R} \cdot \mathbf{B}))$ are all matrices where only the last column is nonzero. However, the product $(\mathbf{R} \cdot (\mathbf{R} \cdot \mathbf{B})) \cdot \mathbf{L}^{-1}$ is in general a full matrix, since $\mathbf{L}$ can be completely arbitrary in our approach. Analogously, the product $\mathbf{F} \cdot \mathbf{L}^{-1}$ does not enjoy any particular property, since $\mathbf{F}$ can be completely arbitrary as well, hence it requires $O(n^3)$ operations. This last product does not change between iterations, so it can be performed once, before starting the iterations. The overall complexity of using the iteration specified by Eq. (1) when $\mathbf{B}$ satisfies our condition is then $O(n^3 + I_{\mathbf{R}} \cdot n^2)$ operations, where $I_{\mathbf{R}}$ is the number of iterations needed to achieve convergence to a given tolerance for $\mathbf{R}$. Since the complexity of using Eq. (1) for a general QBD process results in $O(I_{\mathbf{R}} \cdot n^3)$ operations (see the excellent treatment in [8] for the detailed complexity of several approaches to compute $\mathbf{R}$), our simple observation to exploit the structure of $\mathbf{B}$ leads to a substantial improvement. However, for large values of n, its complexity is still unacceptable. Even more importantly, the need for storing the square matrices $\mathbf{L}^{-1}$ (or its LU-factorization) and $\mathbf{R}$ in full storage remains.

Once $\mathbf{R}$ has been computed, the linear system (2) must be solved to obtain $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{\pi}^{(1)}$. This requires us to compute the product $\mathbf{R} \cdot \mathbf{B}$, which can be done in $O(n^2)$ operations, again by exploiting the structure of $\mathbf{B}$, and then to use a linear system solver, with complexity $O(I_{MG} \cdot (\eta(\mathbf{L}^{(00)}) +$

$\eta(\mathbf{B}^{(10)}) + \eta(\mathbf{F}^{(01)}) + \eta(\mathbf{L} + \mathbf{R} \cdot \mathbf{B})))$, where I_{MG} is the number of iterations required by the solver for convergence.

Finally, measures such as the k^{th} moment can be computed, but expressions of the form $\boldsymbol{\pi}^{(1)} \cdot \mathbf{R}^k \cdot (\mathbf{I} - \mathbf{R})^{-(k+1)}$ must be evaluated [15, p. 36], resulting at a cost of $O(k \cdot n^3)$ operations.

To summarize, the overall computational and storage requirements of the matrix-geometric approach are not simply proportional to the sparsity of the matrices describing $\mathbf{Q}$, as it would be ideally desirable.

7.2 An alternative method to compute R

We now present an alternative method for the computation of $\mathbf{R}$ that explicitly exploits the structure of $\mathbf{B}$ required by our approach. Consider the absorbing CTMC with state space $\{0, 1, 2, \ldots, n\}$ and infinitesimal generator

$$\begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{B} \cdot \mathbf{1}^T & \mathbf{L} + \mathbf{F} \cdot \mathbf{1}^T \cdot \mathbf{e}(n) \end{bmatrix},$$

(where $\mathbf{e}(n)$ is a vector having all entries equal zero except for a one in the n^{th} position)

We can think of these states as $\{s_n^{(j)}, s_1^{(j+1)}, s_2^{(j+1)}, \ldots, s_n^{(j+1)}\}$, in the order. State 0 is absorbing, since it corresponds to the taboo states $\mathcal{S}^{(j)}$ at the lower level: given our structure, state $s_n^{(j)}$ is always the first such state to be reached. The transitions out of states $\{1, 2, \ldots, n\}$ are as for the states in $\mathcal{S}^{(j+1)}$, $j \geq 1$, except that all transitions to states in $\mathcal{S}^{(j+2)}$ in the original CTMC are redirected to state n, that is $s_n^{(j+1)}$, since this is the state that will be reached when transitioning from $\mathcal{S}^{(j+2)}$ back into $\mathcal{S}^{(j+1)}$. Of course, transitions from $s_n^{(j+1)}$ to $s_n^{(j+2)}$ in the original CTMC simply cancel their contribution to the diagonal element $\mathbf{L}_{n,n}$ in the absorbing CTMC just defined. This is not a problem, since it simply corresponds to ignoring transitions from a state to itself in a CTMC.

Then, define $\boldsymbol{\Omega}_{h,l}$, to be the expected time spent in state $l \in \{1, \ldots, n\}$, if the absorbing CTMC is initially in state $h \in \{1, \ldots, n\}$, until absorption. Since $\boldsymbol{\Omega}$ is the solution of

$$\boldsymbol{\Omega} \cdot \left(\mathbf{L} + \mathbf{F} \cdot \mathbf{1}^T \cdot \mathbf{e}(n)\right) = -\mathbf{I},$$

we can compute each row $\boldsymbol{\Omega}_{h,\bullet}$, for $h \in \{1, \ldots, n\}$, one at a time using any iterative solution algorithm for linear systems:

$$\boldsymbol{\Omega}_{h,\bullet} \cdot \left(\mathbf{L} + \mathbf{F} \cdot \mathbf{1}^T \cdot \mathbf{e}(n)\right) = -\mathbf{e}(h). \tag{26}$$

Finally, we can express $\mathbf{R}$ as:

$$\mathbf{R}_{k,l} = \sum_{h \in \{1,\ldots,n\}} \frac{\mathbf{F}_{k,h}}{-\mathbf{L}_{k,k}} \cdot \boldsymbol{\Omega}_{h,l} \cdot (-\mathbf{L}_{k,k}) = \sum_{h \in \{1,\ldots,n\}} \mathbf{F}_{k,h} \cdot \boldsymbol{\Omega}_{h,l}.$$

The first equality is obtained by conditioning on the first state reached upon leaving $s_k^{(j)}$ and recalling the definition of $\boldsymbol{\Omega}_{h,l}$. Note that only states in $\mathcal{S}^{(j+1)}$ need be considered when conditioning, and that the probability of reaching state $s_h^{(j+1)}$ is $-\mathbf{F}_{k,h}/\mathbf{L}_{k,k}$. Then, starting from this state, an average of $\boldsymbol{\Omega}_{h,l}$ time units will be spent in $s_l^{(j+1)}$ before returning to states in $\mathcal{S}^{(j)}$, that is, to $s_n^{(j)}$.

The approach just presented is useful to gain more insight into the implications of our structural restriction: if $\mathbf{B}$ had multiple nonzero columns, we would not be able to easily reroute the transitions in $\mathbf{F}$ toward a single state. However, the complexity of this method is still plagued by the "n^2" problem. Indeed, a linear system in n unknowns must be solved n times, for n different right-hand-sides $\mathbf{e}(h)$, $h = 1, \ldots, n$. The resulting overall complexity is $O\left(n \cdot I_{ALT} \cdot \eta(\mathbf{L})\right)$ where I_{ALT} is the number of iterations required for convergence in the numerical method used to solve the n linear systems (26), and $\eta(\mathbf{L})$ is certainly a number no less than n, but hopefully of the order $O(n)$ for many problems of interest.

7.3 Relations to stochastic complementation

Readers familiar with the concept of stochastic complementation [11] will have realized that it is related to the idea we used in the previous section. Indeed, stochastic complementation has also been used to solve infinite CTMCs of the type we consider [2, 9].

Our requirement of single-point return, from $\mathcal{S}^{(j)}$ to $\mathcal{S}^{(j-1)}$ only through $s_n^{(j-1)}$, when applied to the case $j = 2$, implies that, every time the CTMC makes a forward transition from $\mathcal{S}^{(1)}$, it will reenter $\mathcal{S}^{(1)}$ through state $s_n^{(1)}$ with probability one. Hence, the stationary probability vector $\tilde{\boldsymbol{\pi}} = [\tilde{\boldsymbol{\pi}}^{(0)}, \tilde{\boldsymbol{\pi}}^{(1)}]$ of the CTMC having infinitesimal generator

$$\tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{L}^{(00)} & \mathbf{F}^{(01)} \\ \mathbf{B}^{(10)} & \mathbf{L}^{(11)} + \mathbf{F} \cdot \mathbf{1}^T \cdot \mathbf{e}(n) \end{bmatrix}$$

gives the correct conditional stationary probabilities of the states in $\mathcal{S}^{(0)} \cup \mathcal{S}^{(1)}$ for the original QBD process, that is,

$$\forall i \in \mathcal{S}^{(0)} \cup \mathcal{S}^{(1)}, \quad \tilde{\boldsymbol{\pi}}_i = \frac{\boldsymbol{\pi}_i}{\sum_{l \in \mathcal{S}^{(0)} \cup \mathcal{S}^{(1)}} \boldsymbol{\pi}_l}.$$

If we move one block forward, for $j = 3$, we can instead define

$$\tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{L}^{(00)} & \mathbf{F}^{(01)} & \mathbf{0} \\ \mathbf{B}^{(10)} & \mathbf{L}^{(11)} & \mathbf{F} \\ \mathbf{0} & \mathbf{B} & \mathbf{L} + \mathbf{F} \cdot \mathbf{1}^T \cdot \mathbf{e}(n) \end{bmatrix},$$

from which we can obtain the correct conditional stationary probabilities of the states in $\mathcal{S}^{(0)} \cup \mathcal{S}^{(1)} \cup \mathcal{S}^{(2)}$ for the original QBD process, that is, its stationary probability vector $\tilde{\boldsymbol{\pi}}$ satisfies

$$\forall i \in \mathcal{S}^{(0)} \cup \mathcal{S}^{(1)} \cup \mathcal{S}^{(2)}, \quad \tilde{\boldsymbol{\pi}}_i = \frac{\boldsymbol{\pi}_i}{\sum_{l \in \mathcal{S}^{(0)} \cup \mathcal{S}^{(1)} \cup \mathcal{S}^{(2)}} \boldsymbol{\pi}_l},$$

and so on.

Thus, this suggests a brute-force truncation approach where we simply solve the (finite) truncated QBD process obtained by rerouting the forward transitions from $\mathcal{S}^{(j^*)}$ back to state $s_n^{(j^*)}$, for a sufficiently large j^*. We are then guaranteed that every state probability $\tilde{\boldsymbol{\pi}}_i$ we compute is an upper bound on the exact value $\boldsymbol{\pi}_i$ (hence the expected measures we compute using these probabilities are also upper bounds, assuming all reward rates are nonnegative). This is unlike the general case (no restriction on $\mathbf{B}$), where truncation results in an approximation, but not necessarily in a bound. Of course, the applicability of this method is limited by the need to solve a linear system in $m + j^*n$ unknowns. Informally, if the system being modeled is heavily loaded, state probabilities will decrease slowly as j grows, and this would force us to use a very large j^* in order to obtain results we can trust.

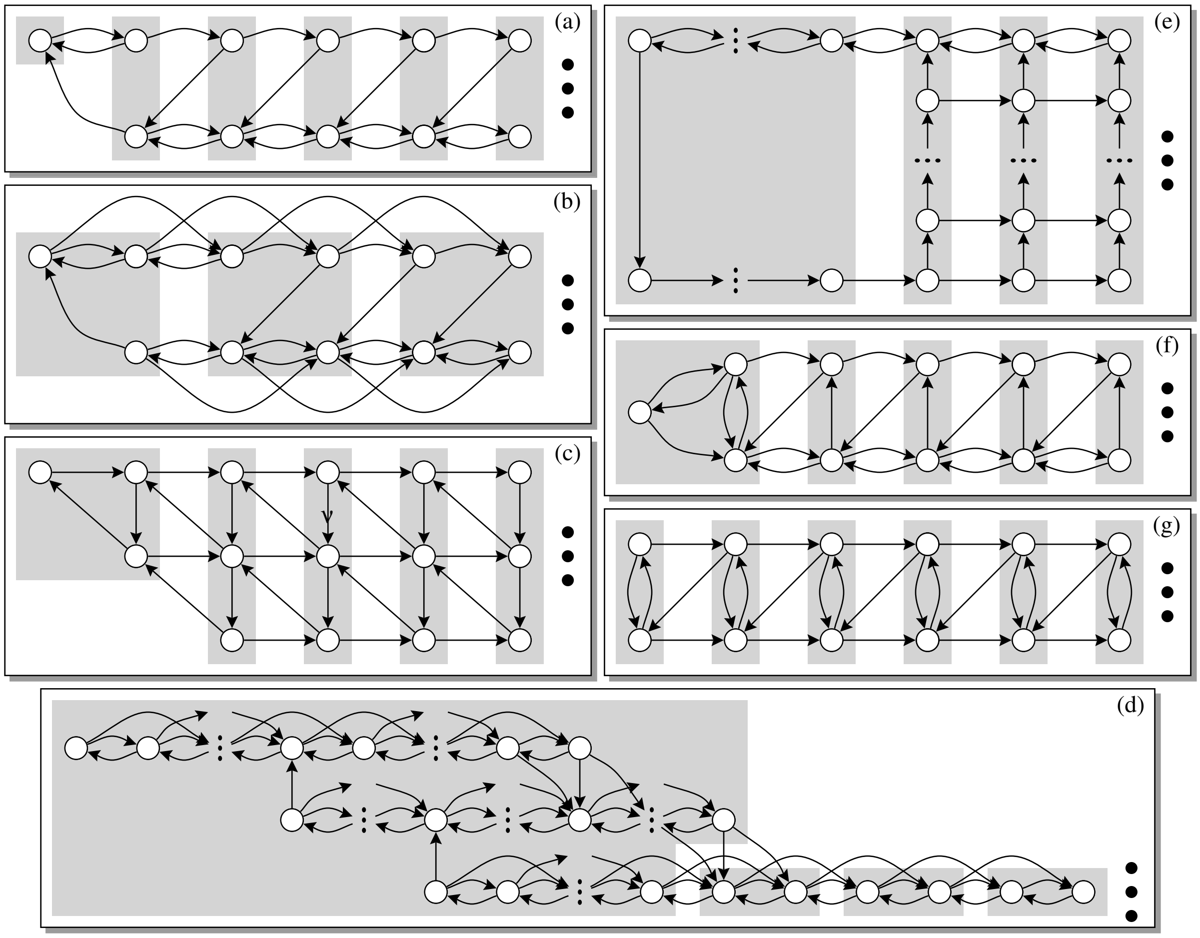

Figure 8: A selection of CTMCs from the published literature where ETAQA applies.

8 Applications

In this section we present a brief survey of applications in the published literature that can be solved by ETAQA. These span from the modeling of multiprocessor systems and databases to operating systems design. In all cases, we only sketch the structure of the CTMC, omitting the actual rates used by the authors (of course, these rates follow the repeating pattern required for the application of the matrix-geometric method).

- Modeling the behavior of different scheduling policies in parallel systems often results in Markov chains with matrix-geometric form [21]. The repeating portion in the resulting chain corresponds to states where all processors are assigned to parallel jobs and further jobs must wait for service in the queue. The forward, local, and backward transitions in each of the repeating portions of the chain are governed by the scheduling strategy. Policies that reduce the number of processors assigned to each job as the number of jobs waiting for service reaches a threshold value [19] result in a CTMC that meets the necessary conditions for ETAQA. Fig. 8(a) illustrates the CTMC of a threshold policy that uses a single threshold only [19].

- In [18], non-work-conserving scheduling policies, which deliberately leave idle processors in the presence of jobs waiting in the queue, are considered. Since [18] focuses on bursty arrival conditions, the behavior of these policies with bulk arrivals is modeled. As illustrated in Fig. 8(b), the backward transitions in the resulting CTMC allow the application of ETAQA.
- [5] considers parallel applications that exhibit a two-stage behavior, so that their execution can be modeled by two portions with different scalability characteristics that are processed in sequence. Dynamic scheduling strategies are evaluated for jobs that exhibit low (high) concurrency at the first stage and high (low) concurrency at the second stage. Fig. 8(c) illustrates the CTMC for a scheduling policy that gives priority to jobs that are in their second stage of execution with the additional restriction that at most $K = 2$ jobs in this stage can execute concurrently. The solution given in [5] is obtained using z-transforms and the authors note that they were unable to provide a solution when $K > 2$. It is interesting to note that, for $K > 2$, the resulting CTMC cannot be solved by ETAQA either.
- Multiserver threshold queuing systems with hysteresis [6, 9] can be used to model systems where the addition and removal of servers is governed by the number of customers in the system. In [6], closed form solutions are derived using the Green's function method and in [9] an alternative solution using stochastic complementation is proposed. Fig. 8(d) illustrates the CTMC of a system with bulk arrivals of limited size where ETAQA applies (the size of the bulk is limited to two in the figure, but any finite size n can be accommodated).
- [23] discusses scheduling strategies for scheduling readers and writers in a database system. Fig. 8(e) shows the CTMC for the Threshold Fastest Emptying (TFE) policy in a reader-saturated system under an open stream of writer arrivals. The system completes the service for the n readers in service but stops admitting additional readers in service when the number of writers waiting reaches a threshold k. For the solution of this system the authors provide a technique that has similarities to the one presented in our paper. However, instead of deriving a relation analogous to our Eq. (21), they use ad-hoc stochastic arguments to obtain one additional equation. It is obvious that our method readily applies to this model.
- Another model from the area of databases appears in [13, Example 9.6], a simplified version of a system discussed in [14]. In a database where data is replicated in two different locations, writes must gain access to both copies, while reads can be satisfied by either copy. The CTMC is in Fig. 8(f); since all returns between levels are to a single state, ETAQA can be applied.
- [3] discusses the scheduling of kernel calls in an operating system. Different scheduling strategies that use a master/slave approach are considered. The CTMC modeling the master processor under the *sampler* scheduling scheme is depicted in Fig. 8(g), and it satisfies the requirements for ETAQA. In [3], this policy was analyzed with the matrix-geometric technique.

We conclude this section, by observing that a classic case where ETAQA applies is that of a single server queue with Poisson arrivals and service rates having a phase-type distribution that can be expressed as the time to absorption of a CTMC with n transient states and a single initial state. When this is the case, any service completion moves the state from $\mathcal{S}^{(j)}$ always to the same state $s_n^{(j-1)}$ of $\mathcal{S}^{(j-1)}$, corresponding to a new job starting its first phase of service, as required by our approach.

9 Conclusion

We have introduced ETAQA, a new solution technique for a class of QBD processes. The approach is derived from first principles and is not based on the recursive expression of the stationary probability of all individual states (unlike the matrix-geometric method, commonly employed for the solution of such models). Nevertheless, a rich set of measures can be obtained, and its computational and storage requirements are vastly better than that of the matrix-geometric method. This is particularly relevant when the matrices describing the model are large but very sparse, a common situation practice.

The only restriction required to apply ETAQA is that all the transitions from states in level j to states in level $j-1$ (i.e., "deaths") are directed toward the same state. While this is a strict subset of the processes that can be solved using the matrix-geometric method, we presented a rich set of applications from the published literature where this condition is satisfied.

Our future efforts will focus on extending the applicability of ETAQA, including the exploration of its use to obtain approximate results.

References

[1] J. Carroll, A. van de Liefvoort, and L. Lipsky. Solutions of M/G/1//N-type loops with extensions to M/G/1 and GI/M/1 queues. *Operations Research*, 30:490–514, 1983.

[2] L. Golubchik and J. C. Lui. Bounding of performance measures of a threashold-based queueing system with hysteresis. In *Proceedings of the 1997 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 147–156, June 1997.

[3] A. G. Greenberg and P. Wright. Design and analysis of master/slave multiprocessors. *IEEE Trans. Comp.*, 40(8):963–976, Aug. 1991.

[4] B. Haverkort, A. van Moorsel, and A. Dijkstra. MGMtool: a performance modelling tool based on matrix geometric techniques. In *Memoranda Informatica 92-35*. University of Twente, Tele-Informatics and Open Systems, 1992.

[5] J.-H. Huang and L. Kleinrock. Performance evaluation of dynamic sharing of processors in two-stage parallel processing systems. *IEEE Trans. Par. and Distr. Syst.*, 4(3):306–317, Mar. 1993.

[6] O. C. Ibe and J. Keilson. Multi-server threshold queues with hysteresis. *Perf. Eval.*, 21:185–213, 1995.

[7] L. Kleinrock. *Queueing Systems Volume I: Theory.* Wiley, 1975.

[8] G. Latouche. Algorithms for infinite Markov chains with repeating columns. In C. Meyer and R. J. Plemmons, editors, *Linear Algebra, Markov Chains, and Queueing Models*, volume 48 of *IMA Volumes in Mathematics and its Applications*, pages 231–265. Springer-Verlag, 1993.

[9] J. Lui and L. Golubchik. Stochastic complement analysis of multi-server threshold queues with hysteresis. Technical Report CUCS-038-96, Columbia University, New York, NY, 1996.

[10] R. A. Marie and J. M. Pellaumail. Steady-state probabilities for a queue with a general service distribution and state-dependent arrivals. *IEEE Transactions on Software Engineering*, SE-9(1):109–113, January 1983.

[11] C. D. Meyer. Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Review*, 31(2):240–271, June 1989.

[12] R. Nelson. Matrix geometric solutions in Markov models: a mathematical tutorial. Research Report RC 16777 (#742931), IBM T.J. Watson Res. Center, Yorktown Heights, NY, Apr. 1991.

[13] R. Nelson. *Probability, Stochastic Processes, and Queueing Theory.* Springer-Verlag, 1995.

[14] R. Nelson and B. R. Iyer. Analysis of a replicated data base. *Perf. Eval.*, 5:133–148, 1985.

[15] M. Neuts. *Matrix-geometric solutions in stochastic models.* Johns Hopkins University Press, Baltimore, MD, 1981.

[16] S. Pissanetzky. *Sparse Matrix Technology.* Academic Press, 1984.

[17] V. Ramswami and G. Latouche. A general class of Markov processes with explicit matrix-geometric solutions. *Operation Research Spectrum*, 8:209–218, Aug. 1986.

[18] E. Rosti, E. Smirni, G. Serazzi, L. Dowdy, and K. Sevcik. On processor saving scheduling policies for multiprocessor systems. *IEEE Trans. Comp.*, 47(2):178–189, Feb. 1998.

[19] E. Smirni, E. Rosti, L. Dowdy, and G. Serazzi. A methodology for the evaluation of multiprocessor non-preemptive allocation policies. *Journal of Systems Architecture*, 44:703–721, 1998.

[20] P. Snyder and W. Stewart. Explicit and iterative numerical approaches to solving queueing models. *Operations Research*, 33(1):183–202, Jan-Feb. 1985.

[21] M. Squillante, F. Wang, and M. Papaefthymiou. Stochastic analysis of gang scheduling in parallel and distributed systems. *Perf. Eval.*, 27/28:273–296, 1996.

[22] M. S. Squillante. MAGIC: a computer performance modeling tool based on matrix-geometric techniques. In G. Balbo. and G. Serazzi, editors, *Computer Performance Evaluation: Modelling Techniques and Tools*, pages 411–425. North-Holland, 1992.

[23] A. Thomasian and V. F. Nicola. Performance evaluation of a threshold policy for scheduling readers and writers. *IEEE Trans. Comp.*, 42(1):83–98, Jan. 1993.