ETAQA-MG1: An Efficient Technique for the Analysis of M/G/1-type processes by Aggregation*

Gianfranco Ciardo Weizhen Mao Alma Riska Evgenia Smirni
Department of Computer Science
College of William and Mary
Williamsburg, VA 23187-8795
{ciardo,wm,riska,esmirni}@cs.wm.edu

September 7, 2000

Abstract

We extend the ETAQA approach, initially proposed for the efficient numerical solution of a class of quasi birth-death processes, to the more complex case of M/G/1-type Markov processes. The new technique is numerically stable and can be used for the exact solution of a class of M/G/1-type models by computing the solution of a finite linear system. We further demonstrate the utility of the method by describing the exact computation of an extensive set of Markov reward functions which include the expected queue length or its higher moments. We also provide an algorithm that finds an appropriate state reordering satisfying our applicability conditions, if one such order exists. We illustrate the method, discuss its complexity, and present comparisons with other traditional techniques.

1 Introduction

We consider Markov chains on an infinite state space having an M/G/1-type structure. Such processes are often the modeling tool of choice for modern computer and communication systems [12]. As a consequence, considerable effort has been placed into the development of exact analysis techniques for them. In the continuous-time case, the infinitesimal generator of such processes is upper block Hessenberg with a repetitive structure, and matrix-analytic methods have been proposed for their solution [14]. The key in the matrix-analytic solution is the computation of an auxiliary matrix called $\mathbf{G}$. Similarly, for Markov chains the of GI/M/1-type, which have a lower block Hessenberg form, matrix-geometric solutions have been proposed [13]. Again, the key in the matrix-geometric solution is the computation of an auxiliary matrix, called $\mathbf{R}$. The traditional solution algorithms compute the stationary probability vector with a recursive function 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), and iterative procedures

*A preliminary version of this paper appeared at the 3rd Meeting on the Numerical Solution of Markov Chains, Zaragoza, Spain, 1999. This work was supported by National Science Foundation under grand no. EIA-9974992, by the National Aeronautics and Space Administration under NASA Grant NAG-1-2168, and by a William and Mary Summer Research Grant.

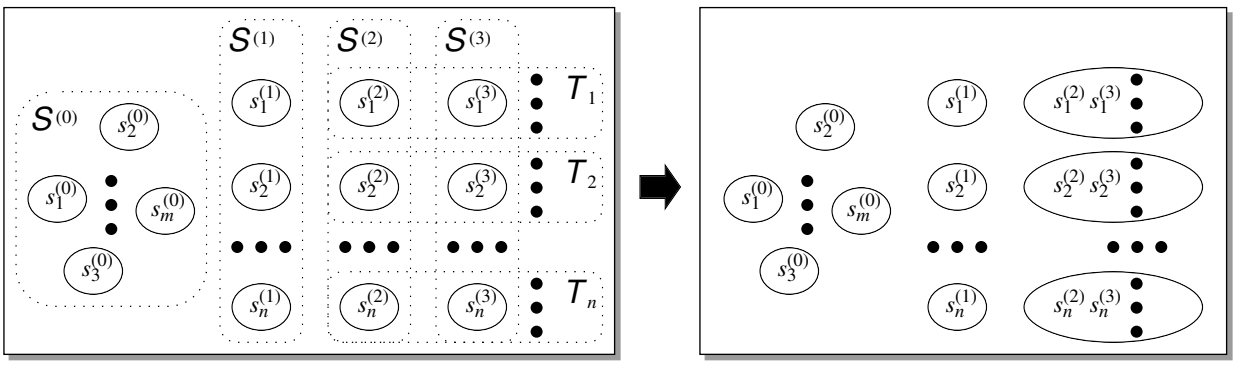

Figure 1: Aggregation of an infinite $\mathcal{S}$ into a finite number of states.

are used for determining $\mathbf{G}$ or $\mathbf{R}$. Alternative algorithms for the computation of $\mathbf{G}$ or $\mathbf{R}$ have been proposed (e.g., the work of Latouche [6] for the efficient computation of $\mathbf{R}$ and of Meini [11] for the efficient computation of $\mathbf{G}$). See also [7] for a discussion of the fundamental aspects of matrix-analytic methods.

The primary distinction between our research and the above works is that we restrict our attention to a family of M/G/1-type processes with a specific form, for which "returns" from a higher level of states to the immediately lower level are always directed toward a single state (for such a subclass, the computation of the matrix $\mathbf{G}$ is trivial, but the remaining solution steps using traditional methods are still expensive). We instead recast the problem into that of solving a finite linear system in $m + 2n$ 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 are able to obtain exact results.

Our approach is an extension of the ETAQA method we introduced for the efficient solution of quasi-birth-death processes with matrix-geometric form [2]. The proposed methodology uses basic well-known results for Markov chains. Assuming that the state space $\mathcal{S}$ 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, $\mathcal{S}^{(j)} = \{s_1^{(j)}, \ldots, s_n^{(j)}\}$, 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 n equivalence classes, appropriately defined. These n equivalence classes $\mathcal{T}_i = \{s_i^{(j)} : j \geq 2\}$, for $1 \leq i \leq n$, correspond to a specific partitioning of the repetitive portion of the chain (see Fig. 1), as well as the stationary probability distribution of each state in $\mathcal{S}^{(0)} \cup \mathcal{S}^{(1)}$.

The new ETAQA-MG1 approach is both efficient and exact, but also numerically stable, since it does not use any recursive formula for the computation of the stationary probability vector. Indeed, our method does not explicitly compute the *entire* stationary probability vector but only the part of it that is related to the boundary states. The traditional matrix-analytic method uses a recursive formula for the computation of the stationary probability vector. If this formula entails subtractions, then there is the possibility of numerical instability [17, 14]. The preferred method for the recursive computation of the stationary probability vector uses Ramaswami's formula [17], a recursive formula that entails only additions. Thus, we compare the computation and storage complexity of our methodology with that of the matrix-analytic method based on Ramaswami's formula.

Our paper is organized as follows. Sect. 2 presents terminology and related work. In Sect. 3 we present the basic theorem that extends ETAQA to M/G/1-type processes. We demonstrate how the methodology can be used for the computation of Markov reward functions in Sect. 4. We continue by showing the applicability of ETAQA-MG1 to bounded bulk arrivals in Sect. 5. In Sect. 6 we compare the computation and storage complexity of ETAQA-MG1 with the matrix-analytic method. We briefly refer in Sect. 7 to an algorithm that can repartition an infinitesimal generator into a form such that ETAQA-MG1 can be applied and elaborate on this algorithm in Appendix A. Sect. 8 presents an applications from the computer systems area that can be analyzed using the ETAQA-MG1 approach, and numerical results. Finally, Sect. 9 summarizes our contributions.

2 Background

We briefly review the terminology used to describe the class of processes we consider. We restrict ourselves to the case of continuous-time Markov chains (hence we refer to the infinitesimal generator matrix $\mathbf{Q}$), but the theory can just as well be applied to the discrete case.

Neuts [13] defines various classes of infinite-state Markov chains with a repetitive structure. In all cases, the state space 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 $j \geq 1$. For the class of M/G/1-type Markov chains, the infinitesimal generator $\mathbf{Q}$ is block-partitioned as:

$$\mathbf{Q} = \begin{bmatrix} \widehat{\mathbf{L}}^{(0)} & \widehat{\mathbf{F}}^{(1)} & \widehat{\mathbf{F}}^{(2)} & \widehat{\mathbf{F}}^{(3)} & \widehat{\mathbf{F}}^{(4)} & \widehat{\mathbf{F}}^{(5)} & \cdots \\ \widehat{\mathbf{B}} & \mathbf{L} & \mathbf{F}^{(1)} & \mathbf{F}^{(2)} & \mathbf{F}^{(3)} & \mathbf{F}^{(4)} & \cdots \\ \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F}^{(1)} & \mathbf{F}^{(2)} & \mathbf{F}^{(3)} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F}^{(1)} & \mathbf{F}^{(2)} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F}^{(1)} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{1}$$

(we use the letters "L", "F", and "B" according to whether they describe "local", 'forward", and "backward" transition rates, respectively, and a "$\widehat{\ }$" for matrices related to $\mathcal{S}^{(0)}$).

For the solution of M/G/1-type processes, several recursive algorithms exist [3, 11, 14]. Here, we outline an algorithm that provides a stable calculation for the values of $\boldsymbol{\pi}^{(j)}$, the stationary probability vector for states in $\mathcal{S}^{(j)}$. Using Ramaswami's recursive formula [17] we have:

$$\forall j \geq 1, \quad \boldsymbol{\pi}^{(j)} = -\left(\boldsymbol{\pi}^{(0)}\widehat{\mathbf{S}}^{(j)} + \sum_{l=1}^{j-1} \boldsymbol{\pi}^{(l)}\mathbf{S}^{(j-l)}\right) {\mathbf{S}^{(0)}}^{-1} \tag{2}$$

where $\widehat{\mathbf{S}}^{(j)}$ and $\mathbf{S}^{(j)}$ are defined as follows:

$$\widehat{\mathbf{S}}^{(j)} = \sum_{l=j}^{\infty} \widehat{\mathbf{F}}^{(l)}\mathbf{G}^{l-j}, \quad j \geq 1, \qquad \mathbf{S}^{(j)} = \sum_{l=j}^{\infty} \mathbf{F}^{(l)}\mathbf{G}^{l-j}, \quad j \geq 0 \quad (\text{letting } \mathbf{F}^{(0)} = \mathbf{L}),$$

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

$$\mathbf{B} + \mathbf{L}\mathbf{G} + \sum_{j=1}^{\infty} \mathbf{F}^{(j)}\mathbf{G}^{j+1} = \mathbf{0}. \tag{3}$$

Several iterative algorithms for the computation of $\mathbf{G}$ exist [14, 3], the most efficient being the one based on Toeplitz matrices [11]. See [16] for cases where $\mathbf{G}$ can be explicitly defined and does not require any calculation.

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 equations

$$\boldsymbol{\pi}^{(0)} \left[\left(\widehat{\mathbf{L}}^{(0)} - \widehat{\mathbf{S}}^{(1)} {\mathbf{S}^{(0)}}^{-1} \widehat{\mathbf{B}} \right)^{\diamond} \;\middle|\; \mathbf{1}^T - \left(\sum_{j=1}^{\infty} \widehat{\mathbf{S}}^{(j)} \right) \left(\sum_{j=0}^{\infty} \mathbf{S}^{(j)} \right)^{-1} \mathbf{1}^T \right] = [\mathbf{0} \mid 1], \tag{4}$$

where the symbol "$\diamond$" 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 $j = 1, 2, \ldots$, stopping when the accumulated probability mass is close to one.

3 Extending ETAQA to M/G/1-type processes

We consider the following structure in the infinitesimal generator matrix $\mathbf{Q}$:

$$\mathbf{Q} = \begin{bmatrix} \widehat{\mathbf{L}}^{(0)} & \widehat{\mathbf{F}}^{(1)} & \widehat{\mathbf{F}}^{(2)} & \widehat{\mathbf{F}}^{(3)} & \widehat{\mathbf{F}}^{(4)} & \widehat{\mathbf{F}}^{(5)} & \cdots \\ \widehat{\mathbf{B}} & \mathbf{L}^{(1)} & \mathbf{F}^{(1)} & \mathbf{F}^{(2)} & \mathbf{F}^{(3)} & \mathbf{F}^{(4)} & \cdots \\ \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F}^{(1)} & \mathbf{F}^{(2)} & \mathbf{F}^{(3)} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F}^{(1)} & \mathbf{F}^{(2)} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F}^{(1)} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{5}$$

which differs from that of Eq. 1 only in block $\mathbf{L}^{(1)}$, at the intersection of the rows and columns for $\mathcal{S}^{(1)}$, which is not restricted to be equal $\mathbf{L}$. This allows us to have (a) local transitions from $\mathcal{S}^{(1)}$ to $\mathcal{S}^{(1)}$ with a pattern other than those from $\mathcal{S}^{(j)}$ to $\mathcal{S}^{(j)}$, $j \geq 2$, and (b) a total rate from each state $s_i^{(1)}$ to the states in $\mathcal{S}^{(0)}$ other than the total rate from state $s_i^{(j)}$ to the states in $\mathcal{S}^{(j-1)}$, i.e., the row sums of $\widehat{\mathbf{B}}$ and $\mathbf{B}$ can be different.

For $\mathbf{Q}$ to be an infinitesimal generator, the infinite sets of matrices $\{\widehat{\mathbf{F}}^{(j)} : j \geq 1\}$ and $\{\mathbf{F}^j : j \geq 1\}$ must be summable. In practice, we must also be able to describe them using a finite representation, so we assume that they obey the following geometric expression:

$$\forall j \geq 1, \qquad \widehat{\mathbf{F}}^{(j)} = \widehat{\mathbf{A}}^{j-1} \widehat{\mathbf{F}} \qquad \text{and} \qquad \mathbf{F}^{(j)} = \mathbf{A}^{j-1} \mathbf{F},$$

where $\widehat{\mathbf{A}}$ and $\mathbf{A}$ are nonnegative diagonal matrices with elements strictly less than one:

$$\widehat{\mathbf{A}} = \begin{bmatrix} \widehat{\alpha}_1 & 0 & 0 & 0 \\ 0 & \widehat{\alpha}_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \widehat{\alpha}_m \end{bmatrix} \qquad \text{and} \qquad \mathbf{A} = \begin{bmatrix} \alpha_1 & 0 & 0 & 0 \\ 0 & \alpha_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_n \end{bmatrix}.$$

This ensures that the infinite sums $\sum_{j=0}^{\infty} \widehat{\mathbf{A}}^j = (\mathbf{I} - \widehat{\mathbf{A}})^{-1}$ and $\sum_{j=0}^{\infty} \mathbf{A}^j = (\mathbf{I} - \mathbf{A})^{-1}$ exist. However, our methodology is not bound to the special geometric expression of the matrices $\widehat{\mathbf{F}}^{(j)}$ and $\mathbf{F}^{(j)}$, but rather to the requirement that these matrices are (efficiently) summable.

If we also partition the stationary probability vector satisfying $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$ as $\boldsymbol{\pi} = \left[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(2)}, \ldots\right]$ with $\boldsymbol{\pi}^{(0)} \in \mathbb{R}^m$ and $\boldsymbol{\pi}^{(j)} \in \mathbb{R}^n$ for $j \geq 1$, we can then write $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$ as:

$$\left\{\begin{array}{lllllllllllllll}
\boldsymbol{\pi}^{(0)} & \widehat{\mathbf{L}}^{(0)} & + & \boldsymbol{\pi}^{(1)} & \widehat{\mathbf{B}} & & & & & & & & & = & \mathbf{0} \\
\boldsymbol{\pi}^{(0)} & \widehat{\mathbf{F}}^{(1)} & + & \boldsymbol{\pi}^{(1)} & \mathbf{L}^{(1)} & + & \boldsymbol{\pi}^{(2)} & \mathbf{B} & & & & & & = & \mathbf{0} \\
\boldsymbol{\pi}^{(0)} & \widehat{\mathbf{F}}^{(2)} & + & \boldsymbol{\pi}^{(1)} & \mathbf{F}^{(1)} & + & \boldsymbol{\pi}^{(2)} & \mathbf{L} & + & \boldsymbol{\pi}^{(3)} & \mathbf{B} & & & = & \mathbf{0} \\
\boldsymbol{\pi}^{(0)} & \widehat{\mathbf{F}}^{(3)} & + & \boldsymbol{\pi}^{(1)} & \mathbf{F}^{(2)} & + & \boldsymbol{\pi}^{(2)} & \mathbf{F}^{(1)} & + & \boldsymbol{\pi}^{(3)} & \mathbf{L} & + & \boldsymbol{\pi}^{(4)} & \mathbf{B} & = & \mathbf{0} \\
 & & & & & & & \ldots & & & & & & & &
\end{array}\right. . \tag{6}$$

3.1 Conditions for stability

We briefly review the conditions that enable us to assert that the CTMC described by the infinitesimal generator $\mathbf{Q}$ in (5) is stable, that is, admits a probability vector satisfying $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$ and $\boldsymbol{\pi}\mathbf{1}^T = 1$.

First, observe that the matrix $\widetilde{\mathbf{Q}} = \mathbf{B} + \mathbf{L} + \sum_{j=1}^{\infty} \mathbf{F}^{(j)}$ is an infinitesimal generator, since it has zero row sums and non-negative off-diagonal entries. If $\widetilde{\mathbf{Q}}$ is irreducible, any state $s_i^{(j)}$ in the original process can reach any state $s_{i'}^{(j')}$, for $2 \leq j \leq j'$ and $1 \leq i, i' \leq n$, without having to go through states in $\mathcal{S}^{(l)}$, $l < j$. In this case, for "large values of j", the conditional probability of being in $s_i^{(j)}$ given that we are in $\mathcal{S}^{(j)}$ tends to $\widetilde{\boldsymbol{\pi}}_i$, where $\widetilde{\boldsymbol{\pi}}$ is the unique solution of $\widetilde{\boldsymbol{\pi}}\widetilde{\mathbf{Q}} = \mathbf{0}$, subject to $\widetilde{\boldsymbol{\pi}}\mathbf{1}^T = 1$, that is, the stationary solution of the ergodic CTMC having $\widetilde{\mathbf{Q}}$ as the infinitesimal generator.

Then, the M/G/1-type process is stable as long as, for large values of j, the forward drift from $\mathcal{S}^{(j)}$ is less than the backward drift from it:

$$\widetilde{\boldsymbol{\pi}} \left(\sum_{j=1}^{\infty} j\mathbf{F}^{(j)} \right) \mathbf{1}^T < \widetilde{\boldsymbol{\pi}}\mathbf{B}\mathbf{1}^T.$$

This condition can be verified numerically, and it is easy to see that it is equivalent to the one given by Neuts in [14], $\widetilde{\boldsymbol{\pi}}\boldsymbol{\beta} < 1$, where, in our terminology, the column vector $\boldsymbol{\beta}$ is given by $\boldsymbol{\beta} = \left(\mathbf{L} + \sum_{j=1}^{\infty} j\mathbf{F}^{(j)}\right)\mathbf{1}^T$. As in the scalar case, the condition where $\widetilde{\boldsymbol{\pi}}\boldsymbol{\beta}$ is exactly equal to 1 results in a null-recurrent CTMC. If $\widetilde{\mathbf{Q}}$ is instead reducible, this stability condition must be applied to each subset of $\{1, \ldots, n\}$ corresponding to a recurrent class in the CTMC described by $\widetilde{\mathbf{Q}}$.

3.2 Main theorem

As done in [2], we require that $\mathbf{B}$ contains nonzero entries only in one column, which we assume to be the last one, without loss of generality. Then, we derive $m + 2n$ equations in $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and the new vector of n unknowns $\boldsymbol{\pi}^{(*)}$. Under these assumptions we can formulate the following theorem.

Theorem 1 (ETAQA-MG1). Given an ergodic CTMC with infinitesimal generator $\mathbf{Q}$ having the structure shown in (5) such that the first $n-1$ columns of $\mathbf{B}$ are null, $\mathbf{B}_{1:n,1:n-1} = \mathbf{0}$, and with stationary probability vector $\boldsymbol{\pi} = \left[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(2)}, \ldots\right]$, the system of linear equations

$$\mathbf{x}\mathbf{M} = [1, \mathbf{0}] \tag{7}$$

where $\mathbf{M} \in \mathbb{R}^{(m+2n)\times(m+2n)}$ is defined as follows:

$$\mathbf{M} = \left[\begin{array}{c|c|c|c|c} \mathbf{1}^T & \widehat{\mathbf{L}}^{(0)} & \widehat{\mathbf{F}}^{(1)}_{1:m,1:n-1} & (\widehat{\mathbf{A}}(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\widehat{\mathbf{F}})_{1:n,1:n-1} & \widehat{\mathbf{A}}(\mathbf{I}-\widehat{\mathbf{A}})^{-2}\widehat{\mathbf{F}}\mathbf{1}^T \\ \mathbf{1}^T & \widehat{\mathbf{B}} & \mathbf{L}^{(1)}_{1:n,1:n-1} & ((\mathbf{I}-\mathbf{A})^{-1}\mathbf{F})_{1:n,1:n-1} & (\mathbf{I}-\mathbf{A})^{-2}\mathbf{F}\mathbf{1}^T \\ \mathbf{1}^T & \mathbf{0} & \mathbf{0} & ((\mathbf{I}-\mathbf{A})^{-1}\mathbf{F}+\mathbf{L})_{1:n,1:n-1} & ((\mathbf{I}-\mathbf{A})^{-2}\mathbf{F}-\mathbf{B})\mathbf{1}^T \end{array}\right] \tag{8}$$

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

Proof: We first show that $\left[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)}\right]$ is a solution of (7) by verifying that it satisfies five matrix equations corresponding to the the five sets of columns we used to define $\mathbf{M}$.

(i) The first equation is the normalization constraint:

$$\boldsymbol{\pi}^{(0)}\mathbf{1}^T + \boldsymbol{\pi}^{(1)}\mathbf{1}^T + \sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)}\mathbf{1}^T = 1. \tag{9}$$

(ii) The second set of m equations is the first line in (6):

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

(iii) The third set of $n-1$ equations is from the second line in (6), which defines n equations, fortunately only the last one actually containing a contribution from $\boldsymbol{\pi}^{(2)}$, due to the structure of $\mathbf{B}$. This is of fundamental importance, since $\boldsymbol{\pi}^{(2)}$ is not one of our variables. Hence, we consider only the first $n-1$ equations and write:

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

(iv) Another set of $n-1$ equations is obtained as follows. First, the sum of the remaining lines in (6) gives

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

which, since $\sum_{j=2}^{\infty} \widehat{\mathbf{F}}^{(j)} = \widehat{\mathbf{A}}(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\widehat{\mathbf{F}}$ and $\sum_{j=1}^{\infty} \mathbf{F}^{(j)} = (\mathbf{I}-\mathbf{A})^{-1}\mathbf{F}$, can be written as

$$\boldsymbol{\pi}^{(0)}\widehat{\mathbf{A}}(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\widehat{\mathbf{F}} + \boldsymbol{\pi}^{(1)}(\mathbf{I}-\mathbf{A})^{-1}\mathbf{F} + \sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)}\left((\mathbf{I}-\mathbf{A})^{-1}\mathbf{F}+\mathbf{L}\right) + \left(\sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)} - \boldsymbol{\pi}^{(2)}\right)\mathbf{B} = \mathbf{0}.$$

Again, we remove the explicit mention of $\boldsymbol{\pi}^{(2)}$ by considering only the first $n-1$ equations:

$$\begin{aligned} \boldsymbol{\pi}^{(0)}\left(\widehat{\mathbf{A}}(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\widehat{\mathbf{F}}\right)_{1:n,1:n-1} + \boldsymbol{\pi}^{(1)}\left((\mathbf{I}-\mathbf{A})^{-1}\mathbf{F}\right)_{1:n,1:n-1} &+ \\ \sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)}\left((\mathbf{I}-\mathbf{A})^{-1}\mathbf{F}+\mathbf{L}\right)_{1:n,1:n-1} &= \mathbf{0} \end{aligned} \tag{12}$$

(v) For the last equation, consider the flow balance equations between $\cup_{l=0}^{j} \mathcal{S}^{(l)}$ and $\cup_{l=j+1}^{\infty} \mathcal{S}^{(l)}$, for

each $j \geq 1$,

$$\left\{\begin{array}{lllll}
\boldsymbol{\pi}^{(0)} \sum_{l=2}^{\infty} \widehat{\mathbf{F}}^{(l)} \mathbf{1}^T & + & \boldsymbol{\pi}^{(1)} \sum_{l=1}^{\infty} \mathbf{F}^{(l)} \mathbf{1}^T & & = \boldsymbol{\pi}^{(2)} \mathbf{B} \mathbf{1}^T \\
\boldsymbol{\pi}^{(0)} \sum_{l=3}^{\infty} \widehat{\mathbf{F}}^{(l)} \mathbf{1}^T & + & \boldsymbol{\pi}^{(1)} \sum_{l=2}^{\infty} \mathbf{F}^{(l)} \mathbf{1}^T & + \boldsymbol{\pi}^{(2)} \sum_{l=1}^{\infty} \mathbf{F}^{(l)} \mathbf{1}^T & = \boldsymbol{\pi}^{(3)} \mathbf{B} \mathbf{1}^T \\
 & & \cdots & & \\
\boldsymbol{\pi}^{(0)} \sum_{l=j+1}^{\infty} \widehat{\mathbf{F}}^{(l)} \mathbf{1}^T & + & \boldsymbol{\pi}^{(1)} \sum_{l=j}^{\infty} \mathbf{F}^{(l)} \mathbf{1}^T & + \boldsymbol{\pi}^{(2)} \sum_{l=j-1}^{\infty} \mathbf{F}^{(l)} \mathbf{1}^T & + \\
\cdots & + & \boldsymbol{\pi}^{(j)} \sum_{l=1}^{\infty} \mathbf{F}^{(l)} \mathbf{1}^T & & = \boldsymbol{\pi}^{(j+1)} \mathbf{B} \mathbf{1}^T \\
 & & \cdots & &
\end{array}\right. \tag{13}$$

and sum these equations by parts:

$$\boldsymbol{\pi}^{(0)} \sum_{j=2}^{\infty} \sum_{l=j}^{\infty} \widehat{\mathbf{F}}^{(l)} \mathbf{1}^T + \boldsymbol{\pi}^{(1)} \sum_{j=1}^{\infty} \sum_{l=j}^{\infty} \mathbf{F}^{(l)} \mathbf{1}^T + \sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)} \sum_{j=1}^{\infty} \sum_{l=j}^{\infty} \mathbf{F}^{(l)} \mathbf{1}^T = \sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)} \mathbf{B} \mathbf{1}^T.$$

Since $\sum_{j=2}^{\infty} \sum_{l=j}^{\infty} \widehat{\mathbf{F}}^{(l)} = \widehat{\mathbf{A}} (\mathbf{I} - \widehat{\mathbf{A}})^{-2} \widehat{\mathbf{F}}$ and $\sum_{j=1}^{\infty} \sum_{l=j}^{\infty} \mathbf{F}^{(l)} = (\mathbf{I} - \mathbf{A})^{-2} \mathbf{F}$, we finally obtain

$$\boldsymbol{\pi}^{(0)} \widehat{\mathbf{A}} (\mathbf{I} - \widehat{\mathbf{A}})^{-2} \widehat{\mathbf{F}} \mathbf{1}^T + \boldsymbol{\pi}^{(1)} (\mathbf{I} - \mathbf{A})^{-2} \mathbf{F} \mathbf{1}^T + \sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)} ((\mathbf{I} - \mathbf{A})^{-2} \mathbf{F} - \mathbf{B}) \mathbf{1}^T = 0. \tag{14}$$

The vector $[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)}]$ satisfies (9), (10), (11), (12), and (14), hence it is a solution of (7). Now we have to show that this solution is unique. For this, it is enough to prove that the rank of $\mathbf{M}$ is $m + 2n$, by showing that its $m + 2n$ rows are linearly independent.

Since $\mathbf{Q}$ is ergodic, we know that the vector $\mathbf{1}^T$ and the 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 (5), corresponding to the transitions into state s_n^1. The result is then the countably infinite set of linearly independent column vectors of $\mathbb{R}^{\mathbb{N}}$, $\{\mathbf{v}^{[1]}, \mathbf{v}^{[2]}, \ldots\}$, shown in Fig. 2. We then define n new vectors of $\mathbb{R}^{\mathbb{N}}$, $\{\mathbf{z}^{[1]}, \ldots, \mathbf{z}^{[n]}\}$ as follows (see Fig. 2):

- 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 of each block of columns, starting from the block corresponding to transitions into level $\mathcal{S}^{(2)}$. $\mathbf{B}$ doesn't appear in the expression of these vectors because $\mathbf{B}_{1:n,1:n-1} = \mathbf{0}$.
- Let $\mathbf{z}^{[n]} = \sum_{j=1}^{\infty} \sum_{l=j}^{\infty} \sum_{i=1}^{n} \mathbf{v}^{[m+ln+i]}$. Fig. 2 shows the steps required to obtained a closed-form expression for $\mathbf{z}^{[n]}$, which are proved by recalling that, since $\mathbf{Q}$ is an infinitesimal generator, $(\mathbf{B} + \mathbf{L} + (\mathbf{I} - \mathbf{A})^{-1} \mathbf{F}) \mathbf{1}^T = \mathbf{0}$, which also implies $(\mathbf{L} + (\mathbf{I} - \mathbf{A})^{-1} \mathbf{F}) \mathbf{1}^T = -\mathbf{B} \mathbf{1}^T$.

Then, we can show that the $m+2n$ vectors $\{\mathbf{v}^{[1]}, \ldots, \mathbf{v}^{[m+n]}, \mathbf{z}^{[1]}, \ldots, \mathbf{z}^{[n]}\}$ are linearly independent:

- The original set $\{\mathbf{v}^{[1]}, \mathbf{v}^{[2]}, \ldots\}$ is linearly independent.
- The vectors $\{\mathbf{z}^{[1]}, \ldots, \mathbf{z}^{[n]}\}$ are obtained as linear combinations of different subsets of vectors from $\{\mathbf{v}^{[m+n+1]}, \mathbf{v}^{[m+n+2]}, \ldots\}$:

$$
\begin{array}{c|c|c|c|c|c|c}
 & \mathbf{v}^{[2]} & \mathbf{v}^{[m+2]} & \mathbf{v}^{[m+n+1]} & \mathbf{v}^{[m+2n+1]} & & \mathbf{z}^{[1]} \\
\mathbf{v}^{[1]} & \text{through} & \text{through} & \text{through} & \text{through} & \cdots & \text{through} \\
 & \mathbf{v}^{[m+1]} & \mathbf{v}^{[m+n]} & \mathbf{v}^{[m+2n]} & \mathbf{v}^{[m+3n]} & & \mathbf{z}^{[n-1]} \\
\hline
\mathbf{1}^T & \widehat{\mathbf{L}}^{(0)} & \widehat{\mathbf{F}}^{(1)}_{1:m,1:n-1} & \widehat{\mathbf{F}}^{(2)} & \widehat{\mathbf{F}}^{(3)} & \cdots & (\widehat{\mathbf{A}}(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\widehat{\mathbf{F}})_{1:m,1:n-1} \\
\mathbf{1}^T & \widehat{\mathbf{B}} & \mathbf{L}^{(1)}_{1:n,1:n-1} & \mathbf{F}^{(1)} & \mathbf{F}^{(2)} & \cdots & ((\mathbf{I}-\mathbf{A})^{-1}\mathbf{F})_{1:n,1:n-1} \\
\mathbf{1}^T & \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{F}^{(1)} & \cdots & ((\mathbf{I}-\mathbf{A})^{-1}\mathbf{F}+\mathbf{L})_{1:n,1:n-1} \\
\mathbf{1}^T & \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \cdots & ((\mathbf{I}-\mathbf{A})^{-1}\mathbf{F}+\mathbf{L})_{1:n,1:n-1} \\
\mathbf{1}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B} & \cdots & ((\mathbf{I}-\mathbf{A})^{-1}\mathbf{F}+\mathbf{L})_{1:n,1:n-1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots
\end{array}
$$

$$
\mathbf{z}^{[n]} = \sum_{j=1}^{\infty}\sum_{l=j}^{\infty}\left(\sum_{i=1}^{n}\mathbf{v}^{[m+ln+i]}\right) = \sum_{j=1}^{\infty}\sum_{l=j}^{\infty}\begin{bmatrix}\widehat{\mathbf{F}}^{(l+1)}\mathbf{1}^T\\ \mathbf{F}^{(l)}\mathbf{1}^T\\ \vdots\\ \mathbf{F}^{(1)}\mathbf{1}^T\\ \mathbf{L}\mathbf{1}^T\\ \mathbf{B}\mathbf{1}^T\\ \mathbf{0}\\ \mathbf{0}\\ \vdots\end{bmatrix} = \sum_{j=1}^{\infty}\begin{bmatrix}\widehat{\mathbf{A}}^j(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\widehat{\mathbf{F}}\mathbf{1}^T\\ \mathbf{A}^{j-1}(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\mathbf{F}\mathbf{1}^T\\ \vdots\\ \mathbf{A}^0(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\mathbf{F}\mathbf{1}^T\\ \left(\mathbf{L}+(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\mathbf{F}\right)\mathbf{1}^T\\ \left(\mathbf{B}+\mathbf{L}+(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\mathbf{F}\right)\mathbf{1}^T\\ \left(\mathbf{B}+\mathbf{L}+(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\mathbf{F}\right)\mathbf{1}^T\\ \vdots\end{bmatrix} =
$$

$$
\sum_{j=1}^{\infty}\begin{bmatrix}\widehat{\mathbf{A}}^j(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\widehat{\mathbf{F}}\mathbf{1}^T\\ \mathbf{A}^{j-1}(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\mathbf{F}\mathbf{1}^T\\ \vdots\\ \mathbf{A}^0(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\mathbf{F}\mathbf{1}^T\\ \left(\mathbf{L}+(\mathbf{I}-\widehat{\mathbf{A}})^{-1}\mathbf{F}\right)\mathbf{1}^T\\ \mathbf{0}\\ \mathbf{0}\\ \vdots\end{bmatrix} = \begin{bmatrix}\widehat{\mathbf{A}}(\mathbf{I}-\widehat{\mathbf{A}})^{-2}\widehat{\mathbf{F}}\mathbf{1}^T\\ (\mathbf{I}-\mathbf{A})^2\mathbf{F}\mathbf{1}^T\\ \left(\mathbf{L}+(\mathbf{I}-\mathbf{A})^{-1}\mathbf{F}\right)\mathbf{1}^T+(\mathbf{I}-\mathbf{A})^{-2}\mathbf{F}\mathbf{1}^T\\ \left(\mathbf{L}+(\mathbf{I}-\mathbf{A})^{-1}\mathbf{F}\right)\mathbf{1}^T+(\mathbf{I}-\mathbf{A})^{-2}\mathbf{F}\mathbf{1}^T\\ \vdots\end{bmatrix} = \begin{bmatrix}\widehat{\mathbf{A}}(\mathbf{I}-\widehat{\mathbf{A}})^{-2}\widehat{\mathbf{F}}\mathbf{1}^T\\ (\mathbf{I}-\mathbf{A})^2\mathbf{F}\mathbf{1}^T\\ \left((\mathbf{I}-\mathbf{A})^{-2}\mathbf{F}-\mathbf{B}\right)\mathbf{1}^T\\ \left((\mathbf{I}-\mathbf{A})^{-2}\mathbf{F}-\mathbf{B}\right)\mathbf{1}^T\\ \vdots\end{bmatrix}
$$

Figure 2: The column vectors used to prove linear independence.

- Disjoint subsets of vectors are used to build $\{\mathbf{z}^{[1]},\ldots,\mathbf{z}^{[n-1]}\}$.
- $\mathbf{z}^{[n]}$ 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 $\{\mathbf{z}^{[1]},\ldots,\mathbf{z}^{[n-1]}\}$.

Hence, the matrix having as column the vectors $\{\mathbf{v}^{[1]},\ldots,\mathbf{v}^{[m+n]},\mathbf{z}^{[1]},\ldots,\mathbf{z}^{[n]}\}$ has rank $m+2n$, which implies that we must be able to find $m+2n$ linearly independent rows in it. Since row $m+jn+i$ is identical to row $m+n+i$ for $j>1$ and $i=1,\ldots,n$, the first $m+2n$ rows must be linearly independent. These are also the rows of our matrix $\mathbf{M}$, and so the proof is complete. □

4 Computing the measures of interest

We now consider the problem of obtaining stationary measures of interest once $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$ have been computed. 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, if we wanted 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, we would let $\boldsymbol{\rho}_i^{(j)} = j$, while, to compute the second moment of the queue length, we would 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 explictly 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 $\mathbf{a}_i^{[0]}, \mathbf{a}_i^{[1]}, \ldots, \mathbf{a}_i^{[k]}$:

$$\forall j \geq 2,\ \forall i \in \{1, 2, \ldots, n\}, \qquad \boldsymbol{\rho}_i^{(j)} = \mathbf{a}_i^{[0]} + \mathbf{a}_i^{[1]} j + \cdots + \mathbf{a}_i^{[k]} j^k. \tag{15}$$

In this case, then,

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

and the problem is reduced to the 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 the equations in (6) from the second line on by the appropriate factor j^k results in

$$\left\{ \begin{array}{lllllllllll}
2^k \boldsymbol{\pi}^{(0)} & \widehat{\mathbf{F}}^{(1)} & + & 2^k \boldsymbol{\pi}^{(1)} & \mathbf{L}^{(1)} & + & 2^k \boldsymbol{\pi}^{(2)} & \mathbf{B} & & & = \mathbf{0} \\
3^k \boldsymbol{\pi}^{(0)} & \widehat{\mathbf{F}}^{(2)} & + & 3^k \boldsymbol{\pi}^{(1)} & \mathbf{F}^{(1)} & + & 3^k \boldsymbol{\pi}^{(2)} & \mathbf{L} & + & 3^k \boldsymbol{\pi}^{(3)} & \mathbf{B} \quad = \mathbf{0} \\
 & & & & \cdots & & & & & &
\end{array} \right. .$$

Summing these equations by parts we obtain

$$\underbrace{\boldsymbol{\pi}^{(0)} \sum_{j=0}^{\infty} (j+2)^k \widehat{\mathbf{A}}^j \widehat{\mathbf{F}}}_{\stackrel{\text{def}}{=}\ \mathbf{f}^{[k,0]}} + \underbrace{\boldsymbol{\pi}^{(1)} \left(2^k \mathbf{L}^{(1)} + \sum_{j=0}^{\infty} (j+3)^k \mathbf{A}^j \mathbf{F} \right)}_{\stackrel{\text{def}}{=}\ \mathbf{f}^{[k,1]}} +$$

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

which can then be rewritten as

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

Exchanging the order of summations we obtain

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

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

$$\mathbf{r}^{[k]} \left(\mathbf{L} + \sum_{j=0}^{\infty} \mathbf{A}^j \mathbf{F} \right) + \mathbf{r}^{[k]} \mathbf{B} = -\mathbf{f}^{[k,0]} - \mathbf{f}^{[k,1]} - \sum_{l=1}^{k} \binom{k}{l} \mathbf{r}^{[k-l]} \left(\mathbf{L} + \sum_{j=0}^{\infty} (j+2)^l \mathbf{A}^j \mathbf{F} \right),$$

which is a linear system of the form $\mathbf{r}^{[k]}(\mathbf{B} + \mathbf{L} + (\mathbf{I} - \mathbf{A})^{-1}\mathbf{F}) = \mathbf{b}^{[k]}$, 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 $(\mathbf{B}+\mathbf{L}+(\mathbf{I}-\mathbf{A})^{-1}\mathbf{F})$ is $n-1$, so the above system is under-determined. We then remove the equation corresponding to the last column, resulting in

$$\mathbf{r}^{[k]}((\mathbf{I} - \mathbf{A})^{-1}\mathbf{F} + \mathbf{L})_{1:n,1:n-1} = \mathbf{b}^{[k]}_{1:n-1}, \tag{16}$$

and obtain one additional equation for $\mathbf{r}^{[k]}$ from the equations in (13), again after multiplying them by the appropriate factor j^k,

$$\begin{cases} 2^k \boldsymbol{\pi}^{(0)} & \sum_{l=2}^{\infty} \widehat{\mathbf{F}}^{(l)} \mathbf{1}^T + 2^k \boldsymbol{\pi}^{(1)} & \sum_{l=1}^{\infty} \mathbf{F}^{(l)} \mathbf{1}^T & = 2^k \boldsymbol{\pi}^{(2)} & \mathbf{B}\mathbf{1}^T \\ 3^k \boldsymbol{\pi}^{(0)} & \sum_{l=3}^{\infty} \widehat{\mathbf{F}}^{(l)} \mathbf{1}^T + 3^k \boldsymbol{\pi}^{(1)} & \sum_{l=2}^{\infty} \mathbf{F}^{(l)} \mathbf{1}^T + 3^k \boldsymbol{\pi}^{(2)} & \sum_{l=1}^{\infty} \mathbf{F}^{(l)} \mathbf{1}^T & = 3^k \boldsymbol{\pi}^{(3)} & \mathbf{B}\mathbf{1}^T \\ & & \cdots & & & \end{cases}.$$

Noting that, for $j \geq 1$, $\sum_{l=j}^{\infty} \mathbf{F}^{(l)} = \mathbf{A}^{j-1}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{F}$, we can sum these equations by parts and obtain:

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

which, with steps analogous to those just performed to obtain (16), can be written as

$$\mathbf{r}^{[k]}((\mathbf{I} - \mathbf{A})^{-2}\mathbf{F} - \mathbf{B})\mathbf{1}^T = c^{[k]} \tag{17}$$

where $c^{[k]}$ is, again, an expression containing $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and the vectors $\mathbf{r}^{[0]}$ through $\mathbf{r}^{[k-1]}$.

Note that the $n \times n$ matrix $[((\mathbf{I}-\mathbf{A})^{-1}\mathbf{F}+\mathbf{L})_{1:n,1:n-1}|((\mathbf{I}-\mathbf{A})^{-2}\mathbf{F}-\mathbf{B})\mathbf{1}^T]$ has full rank. This follows from the fact that we already proved that the matrix $\mathbf{M}$ of Theorem 1 has full rank, hence its last n rows are linearly independent. But the first $m+n$ entries in each of those n rows are identical, hence their last n columns must be linearly independent; that is exactly the matrix we are considering here. Hence, we can compute $\mathbf{r}^{[k]}$ using (16) and (17), i.e., solving a linear system in n unknowns (of course, we must do so first for $l = 1, \ldots, k-1$).

We note that the expressions for $\mathbf{b}^{[k]}$ and $c^{[k]}$ contain sums of the form

$$\sum_{j=0}^{\infty}(j+d)^l\mathbf{A}^j = \left[\sum_{j=d}^{\infty} j^l\mathbf{A}^j\right]\mathbf{A}^{-d} = \left(\sum_{j=1}^{\infty} j^l\mathbf{A}^j - \sum_{j=1}^{d-1} j^l\mathbf{A}^j\right)\mathbf{A}^{-d}, \qquad d=2,3 \qquad l=1,\ldots,k.$$

Thus, we need to compute infinite sums of the form $\sum_{j=1}^{\infty} j^l\mathbf{A}^j$, which converge since the spectral radius of $\mathbf{A}$ is strictly less than one. It can be shown that

$$\sum_{j=1}^{\infty} j^l\mathbf{A}^j = \left(\sum_{i=1}^{l} A(l,i)\mathbf{A}^i\right)(\mathbf{I}-\mathbf{A})^{-(l+1)}$$

where $A(l,i)$ are the Eulerian numbers [18], computed using the recursion

$$A(l,i) = iA(l-1,i) + (l-i+1)A(l-1,i-1),$$

with $A(1,1) = 1$ and $A(l,i) = 0$ for $i < 1$ or $i > l$.

As an example, we consider $\mathbf{r}^{[1]}$, which is used to compute measures such as the first moment of the queue length. In this case,

$$\begin{aligned}
\mathbf{b}^{[1]} &= -\boldsymbol{\pi}^{(0)}(2(\mathbf{I}-\widehat{\mathbf{A}})^{-1} + \widehat{\mathbf{A}}(\mathbf{I}-\widehat{\mathbf{A}})^{-2})\widehat{\mathbf{F}} \\
&\quad -\boldsymbol{\pi}^{(1)}\left(2^k\mathbf{L}^{(1)} + (3(\mathbf{I}-\mathbf{A})^{-1} + \mathbf{A}(\mathbf{I}-\mathbf{A})^{-2})\mathbf{F}\right) \\
&\quad -\mathbf{r}^{[0]}\left(\mathbf{A}(\mathbf{I}-\mathbf{A})^{-2}\mathbf{F} + 2(\mathbf{I}-\mathbf{A})^{-1}\mathbf{F} + \mathbf{L}\right) \\
c^{[1]} &= -\boldsymbol{\pi}^{(0)}((\mathbf{I}-\widehat{\mathbf{A}})^{-3} + (\mathbf{I}-\widehat{\mathbf{A}})^{-2})\widehat{\mathbf{A}}\widehat{\mathbf{F}}\mathbf{1}^T - \boldsymbol{\pi}^{(1)}(\mathbf{A}(\mathbf{I}-\mathbf{A})^{-3} + 2(\mathbf{I}-\mathbf{A})^{-2})\mathbf{F}\mathbf{1}^T \\
&\quad -\mathbf{r}^{[0]}(\mathbf{A}(\mathbf{I}-\mathbf{A})^{-3} + (\mathbf{I}-\mathbf{A})^{-2})\mathbf{F}\mathbf{1}^T.
\end{aligned}$$

We conclude by observing that, when the sequences $\{\widehat{\mathbf{F}}^{(j)} : j \geq 1\}$ and $\{\mathbf{F}^{(j)} : j \geq 1\}$ do not have the geometric form we assume, the treatment in this section can be modified appropriately. However, 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).

5 General M/G/1-type processes and bounded bulk arrivals

In the previous section, we assume that the infinite sets of matrices $\{\widehat{\mathbf{F}}^{(j)} : j \geq 1\}$ and $\{\mathbf{F}^j : j \geq 1\}$ follow a geometric expression so that their infinite sums are summable. In this section, we extend Theorem 1 by allowing $\{\widehat{\mathbf{F}}^{(j)} : j \geq 1\}$ and $\{\mathbf{F}^j : j \geq 1\}$ to be general. As a special case, this allows us to manage processes with bounded bulk arrivals. To study this type of models with the classic methodology, a cut-off point is determined such that the process is treated as a queueing system

with bounded bulk arrivals, i.e., it can jump forward by at most p levels. This results in a good approximation, provided that p is sufficiently high [3, 7, 14].

We stress that we consider the case of finite bulk arrivals explicitly also because, while such type of process could be solved using the version of ETAQA presented in [2] or the matrix-geometric method [13], by merging every p levels into a larger single level, the results of this section show how to compute the solution without the corresponding overhead (a factor of p for both execution time and storage in ETAQA, a factor of p^3 for execution time and p^2 for storage in the matrix-geometric method).

If we restrict the process so that it is allowed to jump forward by at most p levels, its infinitesimal generator $\mathbf{Q}$ still has the structure of (5), except that $\widehat{\mathbf{F}}^{(j)}$ and $\mathbf{F}^{(j)}$ are zero for $j > p$:

$$\mathbf{Q} = \begin{bmatrix} \widehat{\mathbf{L}}^{(0)} & \widehat{\mathbf{F}}^{(1)} & \widehat{\mathbf{F}}^{(2)} & \cdots & \widehat{\mathbf{F}}^{(p)} & \mathbf{0} & \mathbf{0} & \cdots \\ \widehat{\mathbf{B}} & \mathbf{L}^{(1)} & \mathbf{F}^{(1)} & \cdots & \mathbf{F}^{(p-1)} & \mathbf{F}^{(p)} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{B} & \mathbf{L} & \cdots & \mathbf{F}^{(p-2)} & \mathbf{F}^{(p-1)} & \mathbf{F}^{(p)} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \cdots & \mathbf{F}^{(p-3)} & \mathbf{F}^{(p-2)} & \mathbf{F}^{(p-1)} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{18}$$

Since the number of matrices on any row is finite, there is no reason to require any particular relation between the matrices $\widehat{\mathbf{F}}^{(j)}$ or $\mathbf{F}^{(j)}$. We can then formulate the following lemma.

Lemma 1 (bulk arrivals). Given an ergodic CTMC with infinitesimal generator $\mathbf{Q}$, having the structure shown in (18) such that the first $n-1$ columns of $\mathbf{B}$ are null, $\mathbf{B}_{1:n,1:n-1} = \mathbf{0}$, and with stationary probability vector $\boldsymbol{\pi} = \left[\boldsymbol{\pi}^{(0)}, \boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(2)}, \ldots\right]$, the system of linear equations

$$\mathbf{x} \left[\begin{array}{c|c|c|c|c} \mathbf{1}^T & \widehat{\mathbf{L}}^{(0)} & \widehat{\mathbf{F}}^{(1)}_{1:m,1:n-1} & \left(\sum_{j=2}^{p} \widehat{\mathbf{F}}^{(j)}\right)_{1:n,1:n-1} & \left(\sum_{j=2}^{p} (j-1)\widehat{\mathbf{F}}^{(j)}\right) \mathbf{1}^T \\ \mathbf{1}^T & \widehat{\mathbf{B}} & \mathbf{L}^{(1)}_{1:n,1:n-1} & \left(\sum_{j=1}^{p} \mathbf{F}^{(j)}\right)_{1:n,1:n-1} & \left(\sum_{j=1}^{p} j\mathbf{F}^{(j)}\right) \mathbf{1}^T \\ \mathbf{1}^T & \mathbf{0} & \mathbf{0} & \left(\sum_{j=1}^{p} \mathbf{F}^{(j)} + \mathbf{L}\right)_{1:n,1:n-1} & \left(\sum_{j=1}^{p} j\mathbf{F}^{(j)} - \mathbf{B}\right) \mathbf{1}^T \end{array}\right] = [1, \mathbf{0}]$$

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 of the proof are exactly the same as those of the theorem introduced in Sect. 3, hence they are omitted. □

The measure of interest r can also be derived as done in Sect. 4. Using the same definitions for r, $\boldsymbol{\rho}$, and $\mathbf{r}^{[l]}$, we need to show how to express the vectors $\mathbf{r}^{[l]}$, $l = 0, \ldots, k$. $\mathbf{r}^{[k]}$ can be computed recursively by solving the equation

$$\mathbf{r}^{[k]} \left[\begin{array}{c|c} \left(\sum_{j=1}^{p} \mathbf{F}^{(j)} + \mathbf{L}\right)_{1:n,1:n-1} & \left(\sum_{j=1}^{p} j\mathbf{F}^{(j)} - \mathbf{B}\right) \mathbf{1}^T \end{array}\right] = \mathbf{b}^{[k]},$$

where $\mathbf{b}^{[k]}$ is computed using $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\mathbf{r}^{[0]}$ through $\mathbf{r}^{[k-1]}$. In the following, we derive the above equation.

Recall that, for the base case, $\mathbf{r}^{[0]}$ is simply $\boldsymbol{\pi}^{(*)}$. Now, assuming that we know how to compute $\mathbf{r}^{[h]}$ for $0 \leq h \leq k-1$, we show how to compute $\mathbf{r}^{[k]}$. Since p is finite, we define the following sums:

$$\widehat{\mathbf{F}}_h = \sum_{j=h}^{p} \widehat{\mathbf{F}}^{(j)}, \quad \mathbf{F}_h = \sum_{j=h}^{p} \mathbf{F}^{(j)} \ \ (1 \leq h \leq p) \quad \text{and} \quad \widehat{\mathbf{F}}_{[c,l]} = \sum_{j=1}^{p} (j+c)^l \widehat{\mathbf{F}}^{(j)}, \quad \mathbf{F}_{[c,l]} = \sum_{j=1}^{p} (j+c)^l \mathbf{F}^{(j)}$$

Multiplying the equations in (6) from the second block on by the appropriate factor j^k (recall that, for bulk arrivals, $\widehat{\mathbf{F}}^{(j)}$ and $\mathbf{F}^{(j)}$ are zero for $j > p$) and summing them we obtain

$$\boldsymbol{\pi}^{(0)} \underbrace{\sum_{j=1}^{p} (j+1)^k \widehat{\mathbf{F}}^{(j)}}_{=\widehat{\mathbf{F}}_{[1,k]}} + \boldsymbol{\pi}^{(1)} 2^k \mathbf{L}^{(1)} + \boldsymbol{\pi}^{(1)} \underbrace{\sum_{j=1}^{p} (j+2)^k \mathbf{F}^{(j)}}_{=\widehat{\mathbf{F}}_{[2,k]}} +$$

$$\sum_{i=2}^{\infty} \boldsymbol{\pi}^{(i)} \left[\sum_{j=1}^{p} (j+i+1)^k \mathbf{F}^{(j)} + (i+1)^k \mathbf{L} + i^k \mathbf{B} \right] = \mathbf{0},$$

which can be rewritten as:

$$\boldsymbol{\pi}^{(0)} \widehat{\mathbf{F}}_{[1,k]} + \boldsymbol{\pi}^{(1)} (2^k \mathbf{L}^{(1)} + \widehat{\mathbf{F}}_{[2,k]}) + \sum_{h=2}^{\infty} \boldsymbol{\pi}^{(h)} \sum_{j=1}^{p} (j+h+1)^k \mathbf{F}^{(j)} + \sum_{l=0}^{k-1} \binom{k}{l} \mathbf{r}^{[l]} \mathbf{L} + \mathbf{r}^{[k]} (\mathbf{L} + \mathbf{B}) = \mathbf{0}. \quad (19)$$

We can express the third term of the above summation as:

$$\begin{aligned}
\sum_{h=2}^{\infty} \boldsymbol{\pi}^{(h)} \sum_{j=1}^{p} (j+h+1)^k \mathbf{F}^{(j)} &= \sum_{h=2}^{\infty} \boldsymbol{\pi}^{(h)} \sum_{l=0}^{k} \binom{k}{l} j^l (h+1)^{k-l} \mathbf{F}^{(j)} \\
&= \sum_{h=2}^{\infty} \boldsymbol{\pi}^{(h)} \sum_{l=0}^{k} \binom{k}{k-l} (h+1)^l \mathbf{F}_{[0,k-l]} \\
&= \sum_{l=0}^{k} \binom{k}{k-l} \sum_{h=2}^{\infty} \boldsymbol{\pi}^{(h)} (h+1)^l \mathbf{F}_{[0,k-l]} \\
&= \sum_{l=0}^{k} \binom{k}{k-l} \sum_{t=0}^{l} \binom{l}{t} \sum_{h=2}^{\infty} h^t \boldsymbol{\pi}^{(h)} \mathbf{F}_{[0,k-l]} \\
&= \sum_{l=0}^{k} \binom{k}{k-l} \sum_{t=0}^{l} \mathbf{r}^{[t]} \mathbf{F}_{[0,k-l]} \\
&= \sum_{l=0}^{k-1} \binom{k}{k-l} \sum_{t=0}^{l} \mathbf{r}^{[t]} \mathbf{F}_{[0,k-l]} + \sum_{t=0}^{k-1} \binom{k}{t} \mathbf{r}^{[t]} \mathbf{F}_{[0,0]} + \mathbf{r}^{[k]} \mathbf{F}_{[0,0]}
\end{aligned}$$

and substituting the third term in (19) with the above expression we obtain:

$$\begin{aligned}
\mathbf{r}^{[k]} (\mathbf{F}_{[0,0]} + \mathbf{L} + \mathbf{B}) = \ & - \left(\boldsymbol{\pi}^{(0)} \widehat{\mathbf{F}}_{[1,k]} + \boldsymbol{\pi}^{(1)} (2^k \mathbf{L}^{(1)} + \mathbf{F}_{[2,k]}) + \sum_{l=0}^{k-1} \binom{k}{l} \mathbf{r}^{[l]} \mathbf{L} \right) \\
& - \left(\sum_{l=0}^{k-1} \binom{k}{k-l} \sum_{t=0}^{l} \mathbf{r}^{[t]} \mathbf{F}_{[0,k-l]} + \sum_{t=0}^{k-1} \binom{k}{t} \mathbf{r}^{[t]} \mathbf{F}_{[0,0]} \right)
\end{aligned}$$

which is a linear system of the form $\mathbf{r}^{[k]}(\mathbf{F}_{[0,0]} + \mathbf{L} + \mathbf{B}) = \mathbf{b}^{[k]}$ where $\mathbf{b}^{[k]}$is an expression that can be computed from $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and the vectors $\mathbf{r}^{[0]}$ through $\mathbf{r}^{[k-1]}$. Since we know that the rank of $\mathbf{F}_{[0,0]} + \mathbf{L} + \mathbf{B}$ is $n-1$, we remove the equation corresponding to the last column, resulting in $n-1$ equations

$$\mathbf{r}^{[k]} \left(\mathbf{F}_{[0,0]} + \mathbf{L}\right)_{1:n,1:n-1} = \mathbf{b}^{[k]}_{1:n-1}. \tag{20}$$

One additional equation is then required. We obtain it from the equations in (14), again by multiplying them by the appropriate j^k and summing them:

$$\boldsymbol{\pi}^{(0)} \sum_{j=2}^{p} j^k \underbrace{\sum_{h=j}^{p} \widehat{\mathbf{F}}^{(h)}}_{=\,\widehat{\mathbf{F}}_j} \mathbf{1}^T + \boldsymbol{\pi}^{(1)} \sum_{j=1}^{p} (j+1)^k \underbrace{\sum_{h=j}^{p} \mathbf{F}^{(h)}}_{=\,\mathbf{F}_j} \mathbf{1}^T + \sum_{h=2}^{\infty} \boldsymbol{\pi}^{(h)} \sum_{j=1}^{p} (j+h)^k \underbrace{\sum_{h=j}^{p} \mathbf{F}^{(h)}}_{=\,\mathbf{F}_j} \mathbf{1}^T = \underbrace{\sum_{j=2}^{\infty} \boldsymbol{\pi}^{(j)} j^k \mathbf{B}\mathbf{1}^T}_{=\,\mathbf{r}^{[k]}\mathbf{B}\mathbf{1}^T},$$

which therefore can be rewritten as:

$$\boldsymbol{\pi}^{(0)} \sum_{j=2}^{p} j^k \widehat{\mathbf{F}}_j \mathbf{1}^T + \boldsymbol{\pi}^{(1)} \sum_{j=1}^{p} (j+1)^k \mathbf{F}_j \mathbf{1}^T + \sum_{h=2}^{\infty} \boldsymbol{\pi}^{(h)} \sum_{j=1}^{p} (j+h)^k \mathbf{F}_j \mathbf{1}^T = \mathbf{r}^{[k]} \mathbf{B}\mathbf{1}^T. \tag{21}$$

The third term of the above summation can then be expressed as:

$$\begin{aligned}
\sum_{h=2}^{\infty} \boldsymbol{\pi}^{(h)} \sum_{j=1}^{p} \sum_{t=0}^{k} \binom{k}{t} j^k h^{k-t} \mathbf{F}_j \mathbf{1}^T &= \sum_{h=2}^{\infty} \boldsymbol{\pi}^{(h)} \sum_{t=0}^{k} \binom{k}{t} h^{k-t} \underbrace{\sum_{j=1}^{p} j^k \mathbf{F}_j}_{\stackrel{\text{def}}{=}\, \overline{\mathbf{F}}_{[k]}} \mathbf{1}^T = \\
\sum_{t=0}^{k} \binom{k}{t} \sum_{h=2}^{\infty} \boldsymbol{\pi}^{(h)} h^{k-t} \overline{\mathbf{F}}_{[k]} \mathbf{1}^T &= \sum_{t=0}^{k} \binom{k}{t} \mathbf{r}^{[k-t]} \overline{\mathbf{F}}_{[k]} \mathbf{1}^T.
\end{aligned}$$

Substituting the above in (21) we obtain

$$\mathbf{r}^{[k]} (\overline{\mathbf{F}}_{[k]} - \mathbf{B}) \mathbf{1}^T = -\left(\boldsymbol{\pi}^{(0)} \sum_{j=2}^{p} j^k \widehat{\mathbf{F}}_j \mathbf{1}^T + \boldsymbol{\pi}^{(1)} \sum_{j=1}^{p} (j+1)^k \mathbf{F}_j \mathbf{1}^T + \sum_{t=1}^{k} \binom{k}{t} \mathbf{r}^{[k-t]} \overline{\mathbf{F}}_{[k]} \right) \tag{22}$$

The right-hand side of the above equation is an expression containing $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and the vectors $\mathbf{r}^{[0]}$ through $\mathbf{r}^{[k-1]}$. The $n \times n$ matrix $\left[\left(\mathbf{F}_{[0,0]} + \mathbf{L}\right)_{1:n,1:n-1} | (\overline{\mathbf{F}}_{[k]} - \mathbf{B}) \mathbf{1}^T \right]$ required to compute $\mathbf{r}^{[k]}$ has full rank and the proof follows from the proof of Lemma 1. The above result can be used to find any moment k, provided that the first $k-1$ moments have been previously computed.

6 ETAQA and alternative solution methods

In this section, we present a detailed comparison of ETAQA-MG1 with the matrix-analytic method outlined in Sect. 2. First, we note that, under our strict condition that matrix $\mathbf{B}$ be a single column matrix, $\mathbf{G}$ does not need to be computed [16], since it is a zero matrix except for the last column, which contains all ones. This observation is further confirmed by the probabilistic interpretation of $\mathbf{G}$: $\mathbf{G}$ expresses the probability of first entering $\mathcal{S}^{(j-1)}$ through each of its states, starting from each state $\mathcal{S}^{(j)}$. Consequently, for the special case of $\mathbf{Q}$ that we consider in this paper, there is no

cost associated for computing and storing $\mathbf{G}$. We therefore concentrate on the cost of computing the probability vectors $\boldsymbol{\pi}^{(j)}$, $j \geq 0$ with the matrix-analytic method, distinguishing two cases: when the matrices $\widehat{\mathbf{F}}^{(j)}, j \geq 1$ and $\mathbf{F}^{(j)}, j \geq 1$ obey the geometric expression $\widehat{\mathbf{F}}^{(j)} = \widehat{\mathbf{A}}^{j-1}\widehat{\mathbf{F}}$ and $\mathbf{F}^{(j)} = \mathbf{A}^{j-1}\mathbf{F}$, and when no assumption is made about their structure.

In the former case, the expressions used for the solution of the system (see Sect. 2) have a closed form. In particular, the vector $\boldsymbol{\pi}^{(0)}$ is obtained by solving the following system of linear equations,

$$\begin{cases} \boldsymbol{\pi}^{(0)} \left(\widehat{\mathbf{L}}^{(0)} - (\widehat{\mathbf{F}} + \widehat{\mathbf{A}}(\mathbf{I} - \widehat{\mathbf{A}})^{-1}\widehat{\mathbf{F}}\mathbf{G})(\mathbf{L} + (\mathbf{I} - \mathbf{A})^{-1}\mathbf{F}\mathbf{G})^{-1}\widehat{\mathbf{B}}\right)^{\diamond} = \mathbf{0} \\ \boldsymbol{\pi}^{(0)} \left(\mathbf{1}^T - ((\mathbf{I} - \widehat{\mathbf{A}})^{-1}\widehat{\mathbf{F}} + \widehat{\mathbf{A}}(\mathbf{I} - \widehat{\mathbf{A}})^{-2}\widehat{\mathbf{F}}\mathbf{G})(\mathbf{L} + (\mathbf{I} - \mathbf{A})^{-1}\mathbf{F} + (\mathbf{I} - \mathbf{A})^{-2}\mathbf{F}\mathbf{G})^{-1}\mathbf{1}^T\right) = 1 \end{cases} \tag{23}$$

and the matrices $\widehat{\mathbf{S}}^{(j)}$ and $\mathbf{S}^{(j)}$ can be expressed as

$$\widehat{\mathbf{S}}^{(j)} = \widehat{\mathbf{A}}^{j-1}\widehat{\mathbf{F}} + \widehat{\mathbf{A}}^{j}(\mathbf{I} - \widehat{\mathbf{A}})^{-1}\widehat{\mathbf{F}}\mathbf{G}, \quad j \geq 2, \qquad \mathbf{S}^{(j)} = \mathbf{A}^{j-1}\mathbf{F} + \mathbf{A}^{j}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{F}\mathbf{G}, \quad j \geq 1.$$

Therefore, we can furnish a closed-form formula for Eq. (2):

$$\boldsymbol{\pi}^{(j)} = \quad - \left(\boldsymbol{\pi}^{(0)} \left(\widehat{\mathbf{A}}^{j-1}\widehat{\mathbf{F}} + \widehat{\mathbf{A}}^{j}(\mathbf{I} - \widehat{\mathbf{A}})^{-1}\widehat{\mathbf{F}}\mathbf{G}\right) + \sum_{l=1}^{j-1} \boldsymbol{\pi}^{(l)} \left(\mathbf{A}^{j-l}\mathbf{F} + \mathbf{A}^{j+1-l}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{F}\mathbf{G}\right) \right)$$

$$(\mathbf{L} + (\mathbf{I} - \mathbf{A})^{-1}\mathbf{F}\mathbf{G})^{-1},$$

For $\boldsymbol{\pi}^{(0)}$, the classic algorithm requires the computation of the inverse of $\mathbf{S}^{(0)}$, $(\mathbf{L}+(\mathbf{I}-\mathbf{A})^{-1}\mathbf{F}\mathbf{G})^{-1}$, which is general, a full matrix. This inverse has to be stored for the duration of computation, increasing the storage complexity for matrix-analytic approach. During the construction of the system of linear equations of Eq. (23), we also need to multiply this full inverse matrix by sparse $m \times n$ and $n \times m$ matrices, for an overall complexity of $O(n^3 + n\eta\{\widehat{\mathbf{B}}, \widehat{\mathbf{F}}\})$, where $\eta\{\cdot\}$ denotes the total number of nonzero entries in the argument matrices. Finally, the resulting linear system is described by a $m \times m$ matrix, which is, in general, full. Instead, for ETAQA-MG1, we only require the solution of a *sparse* system of $m + 2n$ linear independent equations, plus some matrix summations and scalar multiplications (the computation of the required matrix inverses is trivial since $\widehat{\mathbf{A}}$ and $\mathbf{A}$ are nonnegative diagonal matrices).

Unfortunately, it appears that no closed form formula exists for simple measures of interest such as the queue length or any of its higher moments. Therefore, with the matrix-analytic method, we can only compute an approximation of the measures of interest by truncating their summation, considering only the probability vectors $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, ..., $\boldsymbol{\pi}^{(s)}$, up to a large enough s.

We turn now to the general case where the matrices $\widehat{\mathbf{F}}^{(j)}, j \geq 1$ and $\mathbf{F}^{(j)}, j \geq 1$ do not obey a geometric relation, which included the case of bounded bulk arrivals. Both with ETAQA-MG1 and with the matrix-analytic method we employ truncation and consider these matrices only up to a sufficiently large p, as discussed in Sect. 5. However ETAQA's superiority is more obvious in this case. The complexity of the matrix-analytic algorithm has now an additional cost due to the fact that the matrices $\widehat{\mathbf{S}}^{(j)}$ and $\mathbf{S}^{(j)}$ do not have a closed-form expression, therefore they must be computed for all $j \geq 1$, up to the forward matrix truncation point p, which we assume greater than the forward matrix truncation point p. Fortunately, computing these matrices involves only full vector operations, and their storage requires only a full vector each (plus the storage for the forward matrices themselves), due to the structure of $\mathbf{G}$ and to the fact that, in our case, $\mathbf{G}\mathbf{G} = \mathbf{G}$. Also, $\widehat{\mathbf{S}}^{(j)}$ and $\mathbf{S}^{(j)}$ are zero for $j > p$, thus only p full vectors are additionally required to store these matrices, and, using Eq. 2, we only need to keep a sliding window of the $p + 1$ most

recently computed vectors $\boldsymbol{\pi}^{(j)}$. ETAQA-MG1 requires instead only sparse matrix additions and the solution of a sparse system of $m + 2n$ linearly independent equations.

Table 1 summarizes the computational and storage complexity of ETAQA-MG1 and the classic matrix-analytic method. Only ETAQA-MG1 can truly exploit the sparsity of the matrices describing the original infinitesimal generator (e.g., $\widehat{\mathbf{B}}$, $\widehat{\mathbf{L}}^{(0)}$, $\mathbf{L}^{(1)}$, $\mathbf{L}$, $\widehat{\mathbf{F}}$, $\mathbf{F}$, in the case where the forward matrices satisfy a geometric relation) since, in the matrix-analytic method, the computation of inverse or products of sparse but general matrices will usually destroy sparsity. Note that the contribution of $\mathbf{B}$ to the time complexity can be ignored, since it contains at most n nonzero entries.

	Time complexity	**Additional storage**
Computation of $\boldsymbol{\pi}^{(0)}$ (matrix-analytic) or $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$ (ETAQA)		
ETAQA-M/G/1 (geometric)	$O^L(\eta\{\widehat{\mathbf{B}}, \widehat{\mathbf{L}}^{(0)}, \mathbf{L}^{(1)}, \mathbf{L}, \widehat{\mathbf{F}}, \mathbf{F}\})$	none
Matrix-analytic (geometric)	$O^L(m^2) + O(n^3 + n\eta\{\widehat{\mathbf{B}}, \widehat{\mathbf{F}}\})$	$O(m^2 + n^2)$
ETAQA-M/G/1 (general)	$O^L(\eta\{\widehat{\mathbf{B}}, \widehat{\mathbf{L}}^{(0)}, \mathbf{L}^{(1)}, \mathbf{L}, \widehat{\mathbf{F}}, \mathbf{F}\})$	none
Matrix-analytic (general)	$O^L(m^2) + O(n\eta\{\widehat{\mathbf{F}}, \widehat{\mathbf{B}}\} + \eta\{\mathbf{F}\} + n^3)$	$O(m^2 + n^2)$
First moment measures		
ETAQA-M/G/1 (geometric)	$O^L(\eta\{\mathbf{L}, \mathbf{F}\})$	none
Matrix-analytic (geometric)	$O(s(n^2 + \eta\{\widehat{\mathbf{F}}\}) + s^2\eta\{\mathbf{F}\})$	$O(n^2)$
ETAQA-M/G/1 (general)	$O^L(\eta\{\mathbf{L}, \mathbf{F}\})$	none
Matrix-analytic (general)	$O(s(p\eta\{\mathbf{F}\} + n^2) + p\eta\{\widehat{\mathbf{F}}\})$	$O(pn + n^2)$

Table 1: Computational and storage complexity of ETAQA-MG1 and the matrix-analytic method. $O^L(x)$ denotes the time complexity of solving a linear system described by x nonzero entries. In the general case, $\eta\{\widehat{\mathbf{F}}\}$ and $\eta\{\mathbf{F}\}$ should be taken to mean $\eta\{\cup_{j=1}^{p}\widehat{\mathbf{F}}^{(j)}\}$ and $\eta\{\cup_{j=1}^{p}\mathbf{F}^{(j)}\}$, respectively.

7 Repartitioning Q

The approach we introduced in Sect. 3 and 5 is very efficient but its applicability is limited by the requirement that $\mathbf{B}$ has only one nonzero column. There are models where this condition on $\mathbf{B}$ is not immediately satisfied yet it can be achieved through an appropriate *repartitioning* of the states. For illustration simplicity, Fig. 3 shows an example of state repartitioning applied to a QBD process, but exactly the same idea would apply to that model structure if it had a more complex forward arc structure, i.e., if it were an M/G/1-type process.

Fig. 3 (top) shows a QBD process where the transitions from $\mathcal{S}^{(j)}$ to $\mathcal{S}^{(j-1)}$ have two destinations, $s_b^{(j-1)}$ and $s_c^{(j-1)}$. Clearly, the single column condition for $\mathbf{B}$ is not immediately satisfied. Fig. 3 (bottom) shows instead that, by redefining the sets $\mathcal{S}^{(j)}$ in such a way that $\mathcal{S}^{(1)} = \{s_a^{(2)}, s_b^{(1)}, s_c^{(1)}\}$, $\mathcal{S}^{(2)} = \{s_a^{(3)}, s_b^{(2)}, s_c^{(2)}\}$, etc., the transitions from $\mathcal{S}^{(j)}$ to $\mathcal{S}^{(j-1)}$ now go to a single state, $s_c^{(j-1)}$. The condition on $\mathbf{B}$ is then satisfied.

Lemma 2 (repartitioning). An algorithm exists that can accept as input the (finite) block description of the (infinite) infinitesimal generator $\mathbf{Q}$ of an M/G/1-type queue and determine a repartitioning of the state space (if one exists) such that the single-non-zero-column condition

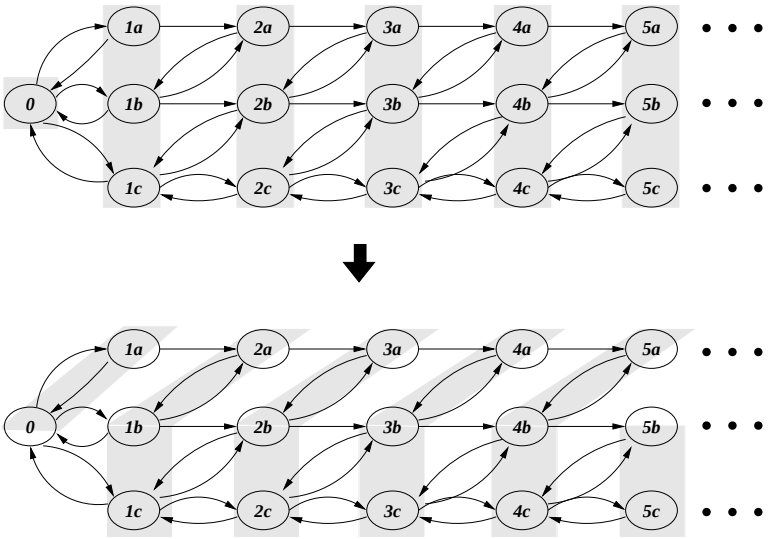

Figure 3: State repartitioning.

for $\mathbf{B}$ is satisfied. The computational complexity of this algorithm is $O(n^2 e \log e)$ and its space complexity is $O(n+e)$, where e is the total number of nonzero entries in $\tilde{\mathbf{Q}} = \mathbf{B} + \mathbf{L} + \sum_{j=1}^{\infty} \mathbf{F}^{(j)}$.
Proof: See Appendix A. □

8 Application: Multiprocessor Scheduling

In this section, we describe an application that can be solved by the ETAQA-MG1 approach. We concentrate on presenting the CTMCs that models the application of interest, focusing on the form of the repeating matrix pattern that allows us to apply our technique, and we present numerical results to compare ETAQA-MG1 with the matrix-analytic approach.

Resource allocation in multiprocessor systems has been a favorite research topic for the performance and operating systems community in recent years. The number of users that attempt to use the system simultaneously, the parallelism of the applications and their respective computational and secondary storage needs, and the need to meet the execution deadlines are examples of issues that exacerbate the difficulty of the resource allocation problem.

A popular way to allocate processors among various competing applications is by space-sharing the processors: processors are partitioned in disjoint groups and each application executes in isolation on one of these groups. Space-sharing can be done in a static, adaptive, or dynamic way. If a job requires a fixed number of processors for execution, this requires a static space-sharing policy [20]. Adaptive space-sharing policies [1] have been proposed for jobs that can configure themselves to the number of processors allocated by the scheduler at the beginning of their execution. Dynamic space-sharing policies [10] have been proposed for jobs that are flexible enough to allow the scheduler to reduce or increase the number of processors allocated to them as they execute, in immediate response to environment changes. Because of their flexibility, dynamic policies can offer optimal performance but are also the ones likely to incur the highest overhead.

Modeling the behavior of scheduling policies in parallel systems often results in CTMCs with matrix-geometric form [22]. Here, we present a CTMC that models the behavior of a class of dynamic

scheduling policies where the reconfiguration cost is reduced by limiting how and when processor reallocations can occur [1]. Fig. 4 illustrates the CTMC modeling a dynamic scheduling policy that can only reduce the number of processor allocated to an application, and only immediately after a service completion. Fig. 4 shows the system behavior under bursty arrival conditions modeled as bulks of finite size (see [19] for an analysis of adaptive space-sharing policies under similar assumptions for the arrival process).

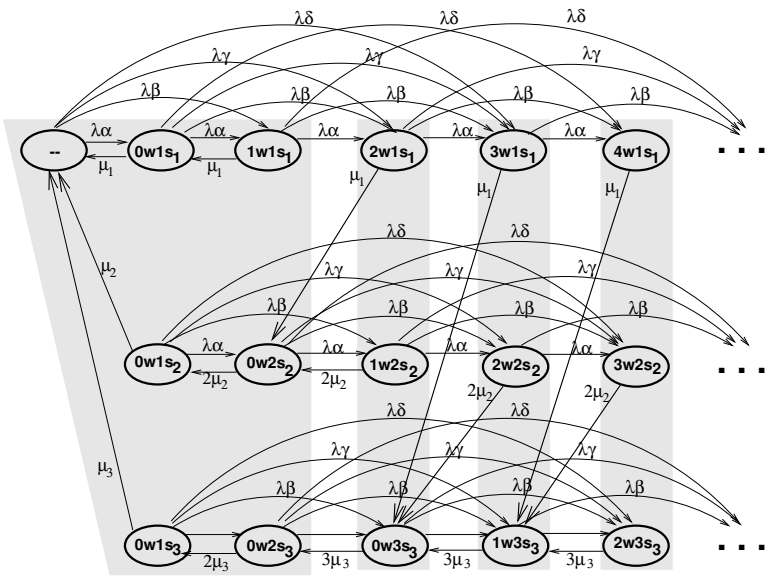

Figure 4: The CTMC that models a dynamic multiprocessor scheduling policy.

The system state is described by the number of applications that are waiting for service in the queue and the number of applications that are in service. $NwMs_r$ denotes that there are N applications in the queue, waiting to be executed, and M applications in service, each having a fraction $\frac{1}{r}$ of the total number of processors allocated to it. Thus, the service rate of an application is μ_r, a function of r. The arrival rate for bulks is λ, and the probability that the size of the bulk is 1, 2, 3, or 4 is α, β, γ, and δ, respectively. For simplicity, the figure assumes that the maximum number of simultaneously executing applications is set to three and that the bulk size is at most four, but any other finite concurrency level and bounded or unbounded bulk sizes can be managed by our approach.

A deterministic set of transition rules specify the successor state for each state in the chain. The policy strives to guarantee that all executing applications are allocated an equal number of processors while minimizing the waiting queue length, but since reallocations occur only at departure times, there are states where there are fewer than three executing applications even when there are applications waiting in the queue. We further assume that the policy incurs no reallocation overhead to illustrate its ideal behavior (this overhead can be accounted for by reducing the service rates when reallocation occurs).

It is easy to observe that our lemma for the bounded bulk arrivals can be readily applied to study the performance analysis of this policy. Since the behavior of this dynamic scheduling policy under different workloads is outside the scope of this paper, we do not present numerical results that illustrate the performance of the policy per se. Instead, we present the time for the solution needed by ETAQA-MG1 and the most efficient matrix-analytic method (as presented in Sect. 2) on a 400Mhz Pentium II with 512KB cache and 384MB memory. We point out that, although Fig. 4

shows that the bulk arrivals have maximum size four and the multiprocessor can be partitioned in up to three partitions, in our experiments we considered maximum bulks of size 50 and 100, and several values for the maximum number of partitions. Table 2 reports the significant runtime improvements achieved when computing $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$, and $\boldsymbol{\pi}^{(*)}$ using ETAQA-MG1, as opposed to computing $\boldsymbol{\pi}^{(0)}$ using the matrix-analytic method, as a function of the maximum number of partitions and the bulk size. Since the maximum number of partitions is p, each repeating level of the CTMC has p states, while the boundary portion has $p^2 - p + 1$ states.

	Max. Bulk Size = 50		Max. Bulk Size = 100	
Max. Partitions	Matrix-analytic	ETAQA-MG1	Matrix-analytic	ETAQA-MG1
5	0.18	0.08	0.51	0.15
10	1.56	0.70	4.80	1.27
15	6.34	2.66	19.61	4.49
20	18.50	8.28	55.76	12.42
25	45.59	21.79	130.39	30.10
30	99.23	55.06	264.66	69.89

Table 2: Execution time (in seconds) for ETAQA-MG1 and the matrix-analytic method.

9 Conclusions and future work

In this paper we presented an extension of the ETAQA approach to M/G/1-type processes. Our exposition focuses on the description of the extended ETAQA methodology and its application to efficiently compute 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 the "mathematically exact" computation of a wide variety of Markov reward functions.

We must emphasize that our treatment applies only to M/G/1-type processes where all transitions from one level to the previous one are directed to a single state (transitions within a level or toward higher levels are completely unrestricted). When this condition is satisfied, the solution is derived by solving a system of $m + 2n$ linear equations and provides significant savings over standard algorithms for the solution of general M/G/1-type processes. While the condition on $\mathbf{B}$ is indisputably restrictive and might not hold on a given CTMC as stated, we derived an algorithm that finds a repartition of the state space so that the condition becomes satisfied, if such a repartition exists.

In the future, we expect to release a software tool that efficiently implements the extended ETAQA. Implementation of the tool is currently underway. We also intend to explore possibilities to extend the methodology to a wider class of M/G/1-type processes than that presented in this paper.

Acknowledgments

We would like to thank our colleague Prof. Paul Stockmeyer for furnishing us with the expression for the sum $\sum_{j=1}^{\infty} j^l \mathbf{A}^j$ and its relation to Euler numbers.

References

[1] S.-H. Chiang, R. Mansharamani, and M. Vernon. Use of application characteristics and limited preemption for run-to-completion parallel processor scheduling policies. In *Proceedings of the 1994 Sigmetrics Conference on Measurement and Modeling of Computer Systems*, pages 33–44, May 1994.

[2] G. Ciardo and E. Smirni. ETAQA: An Efficient Technique for the Analysis of QBD-processes by Aggregation. Submitted for publication.

[3] W. K. Grassman and D. A. Stanford. Matrix Analytic Methods. In W. K. Grassman, editor, *Computational Probability*, pages 153–204, Kluwer Academic Publishers, Boston, MA, 2000.

[4] D. S. Hochbaum and J. Naor, Simple and fast algorithms for linear and integer programs with two variables per inequality, *SIAM Journal on Computing* **23**(6), 1179–1192 (1994).

[5] D. B. Johnson, *Algorithms for Shortest Paths*, Ph.D. Thesis, Cornell University, 1973.

[6] 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.

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

[8] G. Latouche and G. W. Stewart. Numerical methods for M/G/1 type queues. In W. J. Stewart, editor, *Computations with Markov Chains*, pages 571–581. Kluwer, Boston, MA, 1995.

[9] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

[10] C. McCann, R. Vaswani, and J. Zahorjan. A dynamic processor allocation policy for multiprogrammed shared memory multiprocessors. *ACM Transactions on Computer Systems*, 11(2):146–178, 1993.

[11] B. Meini. Solving M/G/1 type Markov chains: Recent advances and applications. *Stochastic Models*, 14(1&2):479–496.

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

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

[14] M. F. Neuts. *Structured stochastic matrices of M/G/1 type and their applications.* Marcel Dekker, New York, NY, 1989.

[15] V. R. Pratt, Two easy theories whose combination is hard, *MIT Technical Report*, 1977.

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

[17] V. Ramswami. A stable recursion for the steady state vector in Markov chains of M/G/1 type. *Commun. Statist.– Stochastic Models*, 4:183–263, 1988.

[18] J. Riordan. *An Introduction to Combinatorial Analysis.* Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, 1958.

[19] 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.

[20] 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.

[21] R. Shostak, Deciding linear inequalities by computing loop residues, *Journal of ACM* **28**(4), 769–779 (1981).

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

[23] G. W. Stewart. On the solution of block hessenberg systems. *Numerical Linear Algebra with Applications*, 2:287–296, 1995.

[24] J. van Leeuwen, Graph algorithms, in *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, The MIT Press, Cambridge, MA, 1990.

A State repartitioning

A.1 Problem definition

We seek an algorithm that, given an infinitesimal generator $\mathbf{Q}$ block-partitioned as in Eq. (5), corresponding to a state space initially partitioned as $\mathcal{S}^{(0)} \cup \mathcal{S}^{(1)} \cup \mathcal{S}^{(2)} \ldots$, with $|\mathcal{S}^{(0)}| = m$ and $|\mathcal{S}^{(j)}| = n$ for $j \geq 1$, discovers a new partition $\widetilde{\mathcal{S}}^{(0)} \cup \widetilde{\mathcal{S}}^{(1)} \cup \widetilde{\mathcal{S}}^{(2)} \ldots$, with $|\widetilde{\mathcal{S}}^{(0)}| = \widetilde{m}$ and $|\widetilde{\mathcal{S}}^{(j)}| = n$, i.e., possibly different-size boundary class but same-size repetive classes, such that the new corresponding block-partition of Eq. (5) is such that $\mathbf{B}$ contains a single nonzero column, if such a repartition exists, or returns a "no" otherwise.

If we just focus on the zero-nonzero pattern of the infinitesimal generator $\mathbf{Q}$, the result is a directed graph. The repeating pattern of $\mathbf{Q}$ implies that there is an edge from $s_i^{(j)}$ to $s_{i'}^{(j)}$ iff there is an edge from $s_i^{(j')}$ to $s_{i'}^{(j')}$, for $i, i' = 1, \ldots, n$ and $j, j' = 1, 2, \ldots$ (since $\mathbf{L}^{(1)}$ can differ from $\mathbf{L}$, exceptions can occur when $j = j' = 1$, but this inessential). We call $s_i^{(1)}, s_i^{(2)}, \ldots$, for $i = 1, \ldots, n$, *same-position* nodes. We call the index a corresponding to the nonzero column in $\mathbf{B}$ the *anchor*. Furthermore, since $\mathbf{Q}$ has an M/G/1-type structure, we know that there are no arcs from $s_i^{(j)}$ to $s_{i'}^{(j')}$, for $j' < j-1$.

A.2 An algebraic approach

Since the subgraphs induced by $\mathcal{S}^{(1)}, \mathcal{S}^{(2)}, \ldots$ in the original partition are all isomorphic and those defined by the new sets $\widetilde{\mathcal{S}}^{(1)}, \widetilde{\mathcal{S}}^{(2)}, \ldots$ after the repartition are required to be isomorphic as well, and of the same size, any change we make to the definition of $\mathcal{S}^{(j)}$ should be made also to the other sets $\mathcal{S}^{(j')}$, for $j, j' \geq 1$. For example, if we shift one node from $\mathcal{S}^{(2)}$ to $\mathcal{S}^{(1)}$, the same-position node in $\mathcal{S}^{(j+1)}$ must also be shifted to $\mathcal{S}^{(j)}$ for $j \geq 2$. Applying the same moving action (shift left, shift right, or leave alone) to same-position nodes guarantees the isomorphism of the resulting subgraphs and that of the forward and backward edges between different subgraphs. As a result, we only need determine where to move the nodes of one of the repetitive subgraphs, say $\mathcal{S}^{(j)}$, to form the new partition. The other subgraphs follow the same pattern.

Consider this subgraph, which contains nodes $s_1^{(j)}, s_2^{(j)}, \ldots, s_n^{(j)}$. After the repartition some nodes will be shifted left to subgraphs with indices smaller than j, some will go right to subgraphs with indices larger than j, and some will stay in their current home. Let $x_i^{(j)}$ be the index of the subgraph to which node $s_i^{(j)}$ will be moved by the repartition. For the time being, suppose that we know the position a of the anchor. We then apply the following rules to establish a linear system of integer inequalities.

- For each edge with a starting point in $\mathcal{S}^{(j)}$, say $s_i^{(j)}$, to $s_l^{(j+r)}$, l other than the anchor a, we must ensure that this edge will not be a backward edge in the new partition, thus

$$x_i^{(j)} \leq x_l^{(j+r)} = x_l^{(j)} + r.$$

- For each edge with a starting point in $\mathcal{S}^{(j)}$, say $s_i^{(j)}$, to an anchor node $s_a^{(j+r)}$, we must ensure that it will not be a backward edge going back more than one level, thus

$$x_i^{(j)} \leq x_l^{(j+r)} + 1 = x_l^{(j)} + r + 1.$$

(in the above, $r = -1$, $r = 0$, or $r > 0$ if the edge is originally a backward, local, or forward edge, respectively).

We then obtain a system of inequalities in n unknowns, $x_1, \ldots, x_n$, with each inequality of the form

$$x_k \leq x_l + c,$$

where c is an integer no smaller than -1, and we have dropped the superscripts "(j)", since no confusion can arise. Clearly, the number of inequalities to be considered equals the number of off-diagonal nonzero entries in $\mathbf{B} + \mathbf{L} + \sum_{j=1}^{\infty} \mathbf{F}^{(j)}$, we let this quantity be e, the number of "edges". Thus, $\mathbf{Q}$ has a new partition with a pre-specified anchor a satisfying the condition on $\mathbf{B}$ iff the linear system of inequalities has an integer solution.

A.3 A necessary and sufficient condition

For the linear system, denoted by L_a, developed in the previous section under the assumption that a is the anchor, we can construct a graph, denoted by G_a, where there is a node for each variable and there is an edge from node x_l to node x_k with weight c iff $x_k \leq x_l + c$ is one of the inequalities in the system.

The sufficient and necessary condition for the linear system L_a to have an integer solution is that there are no cycles with negative total weight in graph G_a. This biimplication was observed by Pratt [15] and later further discussed by Shostak [21], hence the detailed proof is omitted here.

According to van Leeuwen [24], there are efficient algorithms to determine whether there is a negative-weight cycle in a directed graph. One of these algorithms [5, 9] achieves $O(ne)$ worst-case time complexity, if G_a has n nodes and at e edges. Thus we conclude that determining whether the linear system L_a has an integer solution takes $O(ne)$ time. Next, we need to show how to construct a solution if there is one.

A.4 Solving the linear system

Hochbaum and Naor [4] presented an algorithm that finds a feasible integer solution to a system of monotone inequalities of the form $ax_k \leq bx_l + c$, where a and b are positive. Obviously, our inequalities, $x_k \leq x_l + c$, are monotone. Thus, Hochbaum and Naor's algorithm may be applied. However, the algorithm requires that each variable x_k has a lower bound x_k^{min} and an upper bound x_k^{max}, and its resulting complexity is $O(n^2 e \log e + e \sum_{k=1}^{n}(x_k^{max} - x_k^{min} + 1))$. To use this algorithm, we need to show that our variables are bounded.

Consider any system of inequalities of the form $x_k \leq x_l + c$ with at least one integer solution. Without loss of generality, assume that, among the n variables, there is one variable taking the value of j in the solution, and all other variables have the values relative to j. For example, if $x_1 = j$, $x_2 = j + 1$, $x_3 = j$, and $x_4 = j - 1$ is a solution to a system with four variables, then for nodes in $\mathcal{S}^{(j)}$, $s_1^{(j)}$ and $s_3^{(j)}$ stay in $\mathcal{S}^{(j)}$, $s_2^{(j)}$ is moved to $\mathcal{S}^{(j+1)}$ by the repartition, and $s_4^{(j)}$ is moved to $\mathcal{S}^{(j-1)}$. We next argue that, if there is an integer solution (with reference parameter j) then there is an integer solution with $j - n < x_i < j + n$ for $i = 1, \ldots, n$ (in other words, we never need to shift a state by n or more positions). To do so, we construct a new integer solution with the bounded variables based on the solution given.

Let $\mathcal{V}_i$ be the set of all variables given the value of i in the initial solution. Clearly, $\mathcal{V}_j \neq \emptyset$ according to our assumption. Let a be the smallest index such that $\mathcal{V}_a$ is non-empty and b be the largest index such that $\mathcal{V}_b$ is non-empty. If $a \leq j - n$, there must be at least one empty set ("hole") with index between a and j. Let r be the smallest such index. For variables in $\mathcal{V}_a \cup \cdots \cup \mathcal{V}_{r-1}$, we increase their values by one. We argue that the resulting assignment is still a feasible solution. For an inequality $x_k \leq x_l + c$ with both variables in or not in $\mathcal{V}_a \cup \cdots \cup \mathcal{V}_{r-1}$, the equality still holds under the new assignment, since the difference of the values of x_k and x_l is unchanged. For an inequality $x_k \leq x_l + c$ with one variable in $\mathcal{V}_a \cup \cdots \cup \mathcal{V}_{r-1}$ and the other in $\mathcal{V}_s$, for $s > r$, there are two cases. The easy case is $x_k \in \mathcal{V}_s$ and $x_l \in \mathcal{V}_a \cup \cdots \cup \mathcal{V}_{r-1}$. Under the new assignment, the value of x_k is unchanged while the value of x_l is increased by one. The inequality still holds since the increased variable is on the right-hand side of $\leq$. The second case is $x_k \in \mathcal{V}_a \cup \cdots \cup \mathcal{V}_{r-1}$ and $x_l \in \mathcal{V}_s$. Since there is at least one "hole" between $\mathcal{V}_{r-1}$ and $\mathcal{V}_s$, we must have $r - 1 \leq s - 2$, thus $x_k \leq x_l - 2$ before the increase of x_k. Increasing the value of x_k by one, we get $x_k \leq x_l - 1 \leq x_l + c$. So the inequality remains true under the new assignment.

We continue increasing values of some variables until the smallest value in the solution is larger than $j - n$. To the variables holding the values greater than or equal to $j + n$ in the solution, we can decrease their values by one at a time using the similar method until all variables have values smaller than $j + n$ in the solution.

Using the procedure described above, we can bound the variables within the range between $j - n$

and $j + n$. This does not change the integer feasibility of our linear system. For any variable, the difference of the upper bound and lower bound is smaller than $2n$. Using Hochbaum and Naor's algorithm, we achieve the time complexity of $O(n^2 e \log e + e \sum_{k=1}^{n}(2n+1)) = O(n^2 e \log e + en(2n+1)) = O(n^2 e \log e)$.

A.5 Our algorithm and its complexity

We next outline our algorithm for the repartitioning problem. Given an initial partition $\mathcal{S}^{(0)}, \mathcal{S}^{(1)}, \ldots$, for each possible anchor $s_i^{(j)}$, for $i = 1, 2, \ldots, n$, we construct a linear system using the method described in Sect. A.2 and its corresponding graph G_i. Then we determine whether G_i contains any negative-weight cycle. If not, Hochbaum and Naor's algorithm is called to find an integer solution, which is then converted to a new partition of $\mathbf{Q}$. Otherwise, we check the next node to see if it can be the anchor. If no node can be the anchor, the algorithm returns a negative answer.

Input: Graph $\mathbf{Q}$ with partition $\mathcal{S}^{(0)}, \mathcal{S}^{(1)}, \ldots$
Output: A new partition $\widetilde{\mathcal{S}}^{(0)}, \widetilde{\mathcal{S}}^{(1)}, \ldots$, if there is one, or "No" otherwise
For i from 1 to n
    Construct the linear system, L_i, using $s_i^{(j)}$ as the anchor
    Construct the corresponding graph, G_i
    If G_i does not have any negative-weight cycle
        Use Hochbaum and Naor's algorithm to construct an integer solution
        Output the corresponding partition
        Stop
    End if
End for
Output "No"

In any case Hochbaum and Naor's algorithm is called by our algorithm at most once while the negative-weight cycle detection algorithm may be called n times in the worst case. Therefore, the time complexity of our repartitioning algorithm is $O(n^2 e + n^2 e \log e) = O(n^2 e \log e)$. Since the data structures involved are graphs with n nodes and at most e edges and a linear system with n unknowns and at most e inequalities, the space complexity of our repartitioning algorithm is $O(n + e)$.