Triangular M/G/1-type and tree-like QBD Markov chains

Benny Van Houdt
University of Antwerp, Middelheimlaan 1, B2020-Antwerp, Belgium, benny.vanhoudt@ua.ac.be
Johan S.H. van Leeuwaarden
Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands, j.s.h.v.leeuwaarden@tue.nl

In applying matrix-analytic methods to M/G/1-type and tree-like QBD Markov chains, it is crucial to determine the solution to a (set of) nonlinear matrix equation(s). This is usually done via iterative methods. We consider the highly structured subclass of triangular M/G/1-type and tree-like QBD Markov chains that allows for an efficient direct solution of the matrix equation.

Key words: matrix-analytic methods, quasi-birth-death processes, triangular M/G/1-type Markov chain, tree-like QBD, non-linear matrix equation

1. Introduction

Over the last three decades, broad classes of frequently encountered queueing models have been analyzed by *matrix-analytic methods* (Latouche and Ramaswami, 1999; Neuts, 1981, 1989). The embedded Markov chains in these models are two-dimensional generalizations of the classical birth-death processes, and M/G/1 and GI/M/1 queues. Matrix-analytic models include notions such as the Markovian arrival process (MAP) and the phase-type distribution, both in discrete and continuous time. Considerable efforts have been put into the development of efficient and numerically stable methods for their analysis (Bini et al., 2005) and software tools that implement these solutions are available online (Bini et al., 2006a,b; Riska and Smirni, 2002). Apart from the well-known quasi-birth-death (QBD), M/G/1-type and GI/M/1-type Markov chain paradigms, one of the more recent developments in matrix-analytic methods has been the generalization to (discrete-time) bivariate Markov chains with a tree structure (Bini et al., 2003; Takine et al., 1995; Yeung and Sengupta, 1994; Yeung and Alfa, 1999).

The key step in determining the steady-state vector of an M/G/1-type Markov chain exists in finding the smallest non-negative solution G to a nonlinear matrix equation of the form

$$G = \sum_{k=0}^{N} A_k G^k, \tag{1}$$

where $A_0, \ldots, A_N$ are nonnegative $m \times m$ matrices such that $A = \sum_{k=0}^{N} A_k$ is substochastic (see Section 2) and X^0 is assumed to be the identity matrix, for any square matrix X. The class of QBD Markov chains is the subclass of the M/G/1-type Markov chains obtained by setting $N = 2$. The $m \times m$ matrix G is typically computed by iterative methods, where each iteration has a complexity of at least $O(m^3 N)$ (Bini et al., 2006a).

The smallest nonnegative solution G is hardly ever known explicitly, the main exceptions being M/G/1-type Markov chains for which A_0 has rank one, and QBD Markov chains for which A_2 has rank one (Liu and Zhao, 1996). The present paper extends the results in Van Leeuwaarden and Winands (2006) and Van Leeuwaarden et al. (2009), in which explicit solutions were obtained for QBD Markov chains where the A_0, A_1 and A_2 matrices are triangular; see Section 2.4. Indeed, for this case, the sparsity of the matrices, and the structural properties of the model, may lead to determining G in a particularly efficient manner. This was already advocated by Neuts (1981, Section 6.5).

In Section 2, we consider the class of M/G/1-type Markov chains, where all the matrices A_k, for $k = 0, \ldots, N$, are triangular. This class includes several well known models, such as the longest queue model (Cohen, 1987; Flatto, 1989), a two-class priority model (Jaiswal, 1968; Van Velthoven et al., 2006a), a re-entrant line with infinite supply of work (Adan and Weiss, 2006), a make-to-order model (Adan and van der Wal, 1998), the $M/M/c$ model with different service rates for non-waiting customers (Neuts, 1981), queueing systems with multi-task servers (Zhang and Tian, 2004) and more. We first obtain the entries on the main diagonal of the (triangular) solution of G, which corresponds to determining the eigenvalues of G. Next, as opposed to standard spectral decomposition techniques (Grassmann, 1993, 1994), we develop a simple recursive algorithm to compute G one diagonal at a time in an overall time complexity of $O(m^3 N)$, clearly outperforming any existing iterative algorithm (for general M/G/1-type Markov chains) and avoiding the need to construct the corresponding eigenspaces, which may potentially cause some numerical difficulties when eigenvalues with a multiplicity larger than one occur. Each of the diagonal entries/eigenvalues is the smallest nonnegative root of a degree N polynomial, for which we derive an explicit expression in the form of an infinite series and discuss its probabilistic interpretation. However, these explicit expressions do not result in a fast computational technique (unless $N \leq 4$).

Therefore, we propose to use a scalar Newton iteration for each of the diagonal entries.

Furthermore, without going into detail, we argue that a similar approach can be taken for GI/M/1-type Markov chains, where the key equation takes on the form $R = \sum_{k=0}^{N} R^k A_k$. Apart from M/G/1-type and GI/M/1-type Markov chains, we also consider triangular tree-like QBD Markov chains. Explained in more detail in Section 3, the key equation for such an Markov chain is of the form

$$V = B + \sum_{j=1}^{d} U_j (I - V)^{-1} D_j, \tag{2}$$

where B, U_j and D_j for $j = 1$ to d characterize the Markov chain behavior away from the boundary. We will focus on the specific case were these matrices are all triangular. Such Markov chains occur when applying the framework presented in Van Velthoven et al. (2006b) to any *scalar* tree-like QBD Markov chain; various examples of such scalar Markov chains are given in He (2000). As with the G matrix for M/G/1-type Markov chains, we develop a simple recursive algorithm to compute the diagonals of the smallest nonnegative (triangular) solution V one at a time in $O(m^3 d)$. In this case, we also have an explicit expression for the entries on the main diagonal (as these are the solutions of a quadratic equation) and elaborate on their stochastic interpretation. The computational gain for tree-like QBD Markov chains is even more significant than for the M/G/1 case, because most of the iterative algorithms for tree-like QBDs converge linearly or require the solution of a large linear system during each iteration (Bini et al., 2003).

2. Triangular M/G/1-type Markov chains

A discrete-time M/G/1-type Markov chain with a general boundary condition is characterized by a transition matrix P of the form

$$P = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & \dots \\ C_1 & A_1 & A_2 & A_3 & \ddots \\ C_2 & A_0 & A_1 & A_2 & \ddots \\ C_3 & 0 & A_0 & A_1 & \ddots \\ C_4 & 0 & 0 & A_0 & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{3}$$

where B_0 and A_1 are square matrices of dimensions m_0 and m, respectively. Hence, P is the transition matrix of a bivariate Markov chain $\{(X_t, N_t), t \geq 0\}$ with (X_t, N_t) taking values in $\{(0,i) | 1 \leq i \leq m_0\} \cup \{(l,i) | l \geq 1, 1 \leq i \leq m\}$. We will refer to X_t as the *level* of the chain and to N_t as its *phase* at time t. Traditionally, the matrices C_k, for $k \geq 2$, are assumed to be zero. However, their presence only affects the boundary condition of the Markov chain and Eqn. (1) is still the key equation that needs to be solved.

We focus on a subclass of these Markov chains, for which the A_k matrices are triangular, that is,

$$A_k = \begin{bmatrix} a_{1,1}^{(k)} & a_{1,2}^{(k)} & \dots & \dots & a_{1,m}^{(k)} \\ 0 & a_{2,2}^{(k)} & \ddots & \ddots & a_{2,m}^{(k)} \\ 0 & 0 & a_{3,3}^{(k)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_{m-1,m}^{(k)} \\ 0 & \dots & 0 & 0 & a_{m,m}^{(k)} \end{bmatrix},$$

and $A_k = 0$ for $k > N$. Note that, due to the presence of the matrices C_k, for $k \geq 2$, the sum of the matrices A_k, for $k \geq 0$, is not necessarily a stochastic matrix. The key in finding the steady-state probability vector $\pi = (\pi_0, \pi_1, \ldots)$ of P (if π exists), is to find the smallest nonnegative solution to the nonlinear equation (1). Due to the triangular structure of the A_k matrices this solution G is also triangular:

$$G = \begin{bmatrix} g_{1,1} & g_{1,2} & \dots & \dots & g_{1,m} \\ 0 & g_{2,2} & \ddots & \ddots & g_{2,m} \\ 0 & \ddots & g_{3,3} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & g_{m,m} \end{bmatrix}.$$

It is worth mentioning that $g_{m,m} = 1$ if the sum of the A_k matrices, for $k \geq 0$, is a stochastic matrix, which means that $C_k = 0$ for $k \geq 2$.

2.1. Main diagonal of G

By exploiting the structural properties of Eqn. (1), one finds that the diagonal entries $g_{i,i}$ are the smallest nonnegative solution to the nonlinear equations

$$g_{i,i} = \sum_{k=0}^{N} a_{i,i}^{(k)} (g_{i,i})^k. \tag{4}$$

At the end of this section we show how such equations can be solved explicitly. However, this solution does not naturally lead to an efficient implementation. A fast way to compute the entries $g_{i,i}$ uses Newton's method with starting value $g_{i,i}(0) = a_{i,i}^{(0)}$ and the recursive relation

$$g_{i,i}(n+1) = g_{i,i}(n) - \frac{f(g_{i,i}(n))}{f'(g_{i,i}(n))},$$

where $f(x) = \sum_{k=0}^{N} a_{i,i}^{(k)} x^k - x$ and $f'(x) = \sum_{k=1}^{N} k a_{i,i}^{(k)} x^{k-1} - 1$. It is well known that the convergence is quadratic or faster if the function $f(x)$ is continuously differentiable, its derivative does not vanish at $g_{i,i}$, and it has a second derivative at $g_{i,i}$. Hence, the above iteration converges at least quadratically to $g_{i,i}$, which implies that there exists a positive constant c such that $||g_{i,i}(n+1) - g_{i,i}|| \leq c||g_{i,i}(n) - g_{i,i}||^2$.

2.2. Superdiagonals of G

Having computed the main diagonal of G, we shall now discuss how its remaining entries can be retrieved. Because G is triangular, so is its j-th power G^j. Let $g_{i,i+s}^{(j)}$ denote the $(i, i+s)$-th entry of G^j. Further, let $\Delta(G, s)$ denote the matrix

$$\Delta(G,s) = \begin{bmatrix} g_{1,1} & \cdots & g_{1,s} & & \\ & \ddots & & \ddots & \\ & & & & g_{m-s+1,m} \\ & & & \ddots & \vdots \\ & & & & g_{m,m} \end{bmatrix},$$

that is, the matrix holding the first s (super)diagonals of G. Note that $\Delta(G, s)$ and its j-th power $\Delta^j(G, s)$, for $j \geq 0$, are completely determined by the elements $g_{i,i+k}$, for $k = 0, \ldots, s-1$. Moreover, the first s superdiagonals of $\Delta^j(G, s)$ coincide with those of G^j.

Let $\{X\}_{i,i+s}$ denote the $(i, i+s)$-th entry of the matrix X. Entry $g_{i,i+s}^{(j)}$, for $j > 0$, represents the probability that starting from state $(l + j, i)$ at time $t_0 = 0$, for $l > 0$, the chain eventually visits the set of levels $\{0, \ldots, l\}$ for the first time by visiting state $(l, i+s)$ (see Neuts (1981); Latouche and Ramaswami (1999)). As the level $l + j$ can only decrease by one at a time (unless we have a transition to level 0, which does not contribute to G as $l > 0$), we must first visit level $l + j - 1$ for the first time, followed by level $l + j - 2$ and so on until our first visit to level l. Denote the time epochs at which these visits occur as $t_1, t_2, \ldots, t_j$, and let N_{t_r} denote the phase at time t_r, for $r = 0, \ldots, j$. Further, as the phase i can only increase, the increase from i to $i + s$ can only occur in two manners: either the differences $N_{t_r} - N_{t_{r-1}}$, for $r = 1, \ldots, j$ are all less than s, or the phase remains identical to i up to time t_{k-1}, equals $i + s$ at time t_k and remains identical to $i + s$ from time t_k to time t_j, for some $k \in \{1, \ldots, j\}$. The sum of the probabilities of all the sample paths of the first type are captured by $\{\Delta^j(G, s)\}_{i,i+s}$ as this entry contains the sum of all the products of the form $\prod_{r=1}^{j} g_{i+N_{t_{r-1}}, i+N_{t_r}}$, with $N_{t_r} - N_{t_{r-1}} < s$. The sample paths of the second type have a probability mass of $(g_{i,i})^{k-1}g_{i,i+s}(g_{i+s,i+s})^{j-k}$. These observations allow us to conclude that

$$g^{(j)}_{i,i+s} = \{\Delta^j(G,s)\}_{i,i+s} + \sum_{k=1}^{j}(g_{i,i})^{k-1}g_{i,i+s}(g_{i+s,i+s})^{j-k}. \tag{5}$$

Therefore, we can retrieve $g_{i,i+s}$, for $s > 0$, from Eqn. (1) as

$$\begin{aligned} g_{i,i+s} &= \sum_{j=0}^{N}\sum_{k=0}^{s} a^{(j)}_{i,i+k} g^{(j)}_{i+k,i+s} \\ &= \frac{\sum_{j=0}^{N}\sum_{k=1}^{s} a^{(j)}_{i,i+k} g^{(j)}_{i+k,i+s} + \sum_{j=2}^{N} a^{(j)}_{i,i}\{\Delta^j(G,s)\}_{i,i+s}}{1-\sum_{j=1}^{N} a^{(j)}_{i,i}\sum_{k=1}^{j}(g_{i,i})^{k-1}(g_{i+s,i+s})^{j-k}}. \end{aligned} \tag{6}$$

Note that the lower index of the sum starts at $j = 2$ as $\{\Delta^j(G,s)\}_{i,i+s}$ is identical to zero for $j = 0$ or 1. In this manner we can henceforth compute the G matrix one diagonal at a time, resulting in the following $O(m^3N)$ algorithm to compute G from $g_{1,1}, \ldots, g_{m,m}$:

- Start by computing the main diagonal of G^j for $j = 2$ to N (in $O(mN)$ time). At this point, the first diagonal of the matrices G^j, for $j = 1, \ldots, N$, has been computed.
- Next, for $s = 1$ to $m - 1$:
  - Compute the $(s + 1)$-th diagonal of $\Delta^j(G,s)$, for $j = 2$ to N (containing the $(i, i + s)$-th entries) from the first s diagonals of the matrices G^{j-1} and G in $O(m^2N)$ time using

$$\begin{aligned} \{\Delta^j(G,s)\}_{i,i+s} &= \sum_{k=0}^{s}\{\Delta^{j-1}(G,s)\}_{i,i+k}\{\Delta(G,s)\}_{i+k,i+s} \\ &= \sum_{k=1}^{s-1} g^{(j-1)}_{i,i+k} g_{i+k,i+s} + \{\Delta^{j-1}(G,s)\}_{i,i+s}g_{i+s,i+s}. \end{aligned}$$

  - Obtain $g_{i,i+s}$, for $i = 1$ to $m - s$, using relation (6) and

$$\sum_{k=1}^{j}(g_{i,i})^{k-1}(g_{i+s,i+s})^{j-k} = g_{i,i}\sum_{k=1}^{j-1}(g_{i,i})^{k-1}(g_{i+s,i+s})^{j-1-k} + g^{j-1}_{i+s,i+s},$$

    in $O(m^2N)$ time.
  - Finally, using relation (5), compute the elements $g^{(j)}_{i,i+s}$, for $j = 2$ to N, from $\{\Delta^j(G,s)\}_{i,i+s}$ in $O(mN)$ time, that is, the $(s + 1)$-th diagonal of G^j.

This results in an overall time complexity of $O(m^3N)$ and memory complexity of $O(m^2N)$. Recall that all of the (general purpose) iterative algorithms (see Bini et al. (2006a)) to determine G require at least the same amount of memory and time *per iteration*, while the algorithms with quadratic convergence—like the cyclic reduction, Ramaswami reduction or invariant subspace approach algorithm—require considerably more time *per iteration*.

2.3. Explicit solution for the main diagonal

We shall now derive an explicit expression for the diagonal entries $g_{i,i}$ defined by (4). Using

$$\varphi_i(w) = a_{i,i}^{(0)} + a_{i,i}^{(1)} w + \cdots + a_{i,i}^{(N)} w^N, \tag{7}$$

we see that the diagonal entry $g_{i,i}$ satisfies $g_{i,i} = \varphi_i(g_{i,i})$, where we assume that $\varphi_i(0) = a_{i,i}^{(0)} \neq 0$ (if $a_{i,i}^{(0)} = 0$, we have $g_{i,i} = 0$). In fact, $g_{i,i}$ is the solution $w_i = w \in (0, 1]$ of $w = \varphi_i(w)$. Note that $w_i = 1$ if and only if $\varphi_i(1) = 1$ and $\varphi_i'(1) \leq 1$. Let $[w^k]$ denote the operator that extracts the coefficient of w^k from a power series in w. A standard application of the Lagrange inversion formula (see De Bruijn, 1981) then yields

$$g_{i,i} = \sum_{k=0}^{\infty} \frac{1}{k+1} [w^k] \varphi_i(w)^{k+1}.$$

The coefficient $[w^k]\varphi_i(w)^{k+1}$ can be expressed explicitly (see e.g. Pitman, 1998), leading to

$$g_{i,i} = \sum_{k=0}^{\infty} \frac{1}{k+1} \sum_{\mathcal{S}_k} \binom{k}{n_0, \ldots, n_n} \prod_{j=0}^{n} (a_{i,i}^{(j)})^{n_j}, \tag{8}$$

where the second summation is over the finite set $\mathcal{S}_k$ of all sequences of non-negative integers $(n_j, j = 0, \ldots, n)$ with $\sum n_j = k + 1$ and $\sum j n_j = k$. Here

$$\binom{k}{n_0, \ldots, n_n} = \frac{k!}{n_0! \cdots n_n!}$$

is the multinomial coefficient (for which efficient iterative procedures are available).

Special cases of $\varphi_i(w)$ lead to more tractable expressions. For example, an interesting special case is $\varphi_i(w) = a_{i,i}^{(0)} + a_{i,i}^{(N)} w^N$, for which we have that

$$[w^k]\varphi_i(w)^{k+1} = 0 \quad \text{if} \quad k \bmod N \neq 0,$$

and hence

$$\begin{aligned} g_{i,i} &= \sum_{j=0}^{\infty} \frac{1}{jN+1} [w^{jN}] \varphi_i(w)^{jN+1} \\ &= \sum_{j=0}^{\infty} C_{j,N} (a_{i,i}^{(N)})^j (a_{i,i}^{(0)})^{j(N-1)+1}, \end{aligned} \tag{9}$$

with $C_{j,N}$ being the generalized Catalan numbers

$$C_{j,N} = \frac{1}{j(N-1)+1}\binom{jN}{j}, \quad j = 0, 1, 2, \ldots. \tag{10}$$

Expression (9) is reminiscent of expressions for the busy period in queues with batch arrivals (see Takács (1962)) that can be analyzed through ballot problems. Indeed, $C_{j,N}$ is the solution to the following ballot problem:

> In an election in which candidate A gets j votes and candidate B gets $(N-1)j$ votes, what is the number of ways of counting the ballots so that candidate B never has more than $N-1$ times as many votes as candidate A?

The probabilistic interpretation of the matrix G may further explain the analogy to busy periods and ballot problems. Recall that the entry $g_{i,i}$ denotes the probability that the Markov chain starts in state $(l+1, i)$ and visits level l for the first time in state (l, i). Because we restrict to triangular Markov chains, the only way that this can occur is that the Markov chain, between its start in state $(l+1, i)$ and first visit to state (l, i), remains at phase i. The set of events that gives rise to $g_{i,i}$ then clearly resembles a busy period of a queue in which a customer gets served w.p. $a_{i,i}^{(0)}$ or a new batch of $N-1$ customers arrives w.p. $a_{i,i}^{(N)}$. An alternative formulation is in terms of a ballot problem.

2.4. Triangular QBD Markov chains

A discrete-time QBD Markov chain with a generalized boundary condition is characterized by a transition matrix P as in (3), with $N = 2$ and $B_k = 0$ for $k \geq 2$. As with the M/G/1-type Markov chains, the matrices C_k, for $k \geq 2$, are allowed to be nonzero. In Van Leeuwaarden et al. (2009), generalizing earlier work in Van Leeuwaarden and Winands (2006), explicit solutions were obtained for a QBD with homogeneous transition probabilities that do not depend on the phase, that is, $a_{i,i+k}^{(n)} := a_k^{(n)}$ for all i. It was shown that, for $i, \nu \in \{1, \ldots, m\}$,

$$\begin{aligned} g_{i,i+\nu} \quad &= \quad \sum_{n_1+n_2=\nu} \sum_{j=\max(n_2,n_1-1)}^{\infty} C_{j,2} \cdot \binom{j+1}{n_1}\binom{j}{n_2}\binom{2j+\nu-n_1-n_2}{\nu-n_1-n_2} \\ &\quad \times \left(a_1^{(0)}\right)^{n_1} \left(a_1^{(1)}\right)^{\nu-n_1-n_2} \left(a_1^{(2)}\right)^{n_2} \left(a_0^{(2)}\right)^{j-n_2} \left(a_0^{(0)}\right)^{j+1-n_1}. \end{aligned} \tag{11}$$

Here (9) with $N = 2$ corresponds to the special case $\nu = 0$. In Van Leeuwaarden et al. (2009) it is further shown that the infinite series in (11) can be written in terms of hypergeometric functions. We have not pursued a similar analysis for the present model, as its level of generality considerably increases the complexity of the required combinatorial expressions. Instead, we have presented explicit solutions for the main diagonal elements $g_{i,i}$ in Sections 2.1 and 2.3, and a recursive algorithm to compute the remaining elements of G in Section 2.2.

3. Tree-like QBD Markov chains

Let us first briefly describe the class of tree-like QBDs with a generalized boundary condition. Consider a discrete time bivariate Markov chain $\{(X_t,\, N_t), t \geq 0\}$ in which the values of X_t are represented by nodes of a d-ary tree, for $d \geq 1$, and where N_t takes integer values between 1 and m. We refer to X_t as the *node* and to N_t as the *auxiliary* variable of the Markov chain at time t. The root node of the d-ary tree is denoted as $\emptyset$ and the remaining nodes are denoted as strings of integers, where each integer takes a value between 1 and d. For instance, the k-th child of the root node is represented by k, the l-th child of the node k by kl, and so on. Let $f(J,1)$, for $J \neq \emptyset$, denote the rightmost element of the string J and let $J - f(J,1)$ represent the parent node of J.

The following restrictions need to apply for a Markov chain $\{(X_t, N_t), t \geq 0\}$ to be a tree-like QBD process. At each step the chain can only make a transition to its parent (i.e., $X_{t+1} = X_t - f(X_t,1)$, for $X_t \neq \emptyset$), to itself ($X_{t+1} = X_t$), to one of its children ($X_{t+1} = X_t + s$ for some $1 \leq s \leq d$) or to the root node ($X_{t+1} = \emptyset$). Moreover, the state of the chain at time $t+1$ is determined as follows:

$$P[(X_{t+1}, N_{t+1}) = (J', j) | (X_t, N_t) = (J, i)] = \left\{ \begin{array}{ll} f^{i,j} & J' = J = \emptyset, \\ c^{i,j} & J' = \emptyset, J \neq \emptyset, \\ b^{i,j} & J' = J \neq \emptyset, \\ d_k^{i,j} & J \neq \emptyset, f(J,1) = k, J' = J - f(J,1), \\ u_s^{i,j} & J' = J + s, s = 1, \ldots, d, \\ 0 & \text{otherwise.} \end{array} \right.$$

We define the $m \times m$ matrices D_k, C, B, F and U_s with respective $(i,j)^{th}$ elements given by $d_k^{i,j}$, $c^{i,j}$, $b^{i,j}$, $f^{i,j}$ and $u_s^{i,j}$, for $k, s = 1, \ldots, d$. This completes the description of the class of tree-like QBD processes; hence, such a process is fully characterized by the matrices D_k, C, B, U_s and F. Note that tree-like QBDs with $d = 1$ correspond to a standard QBD with $B_0 = F$, $B_1 = U_1$, $A_0 = D_1$, $A_1 = B$, $A_2 = U_1$ and $C_i = C$, for $i \geq 1$.

The key in computing the steady state of a tree-like QBD Markov chain exists in determining a set of d matrices that are the smallest nonnegative solution to

$$G_k = D_k + BG_k + \left(\sum_{j=1}^{d} U_j G_j\right) G_k,$$

for $k = 1, \ldots, d$, which can be done by first solving

$$V = B + \sum_{j=1}^{d} U_j (I - V)^{-1} D_j, \tag{12}$$

and using the relation $G_k = (I - V)^{-1} D_k$. When the matrices B, D_k and U_k are triangular, so are the V and G_k matrices.

3.1. Main diagonal of V

To retrieve the diagonal elements of V, denoted as $v_{i,i}$, it suffices to solve the quadratic equation

$$v_{i,i} = b_{i,i} + \sum_{j=1}^{d} u_{i,i}^{(j)} (1 - v_{i,i})^{-1} d_{i,i}^{(j)},$$

where $b_{i,i'}$, $u_{i,i'}^{(j)}$ and $d_{i,i'}^{(j)}$ denote the (i, i')-th elements of B, U_j and D_j, respectively. Hence,

$$v_{i,i} = \frac{(1 + b_{i,i}) - \sqrt{(1 - b_{i,i})^2 - 4 \sum_j u_{i,i}^{(j)} d_{i,i}^{(j)}}}{2},$$

with $g_{i,i}^{(k)}$, the i-th diagonal element of G_k, equal to $d_{i,i}^{(k)}/(1 - v_{i,i})$.

The entries $v_{i,i}$ can also be expressed in a manner similar to (9). That is, using the binomial series

$$\sqrt{1 - 4z} = 1 - 2 \sum_{j=1}^{\infty} C_{j-1,2} z^j,$$

with $\xi = \frac{1}{(1-b_{i,i})^2} \sum_{j=1}^{d} u_{i,i}^{(j)} d_{i,i}^{(j)}$, we find that

$$\begin{aligned} v_{i,i} &= \frac{(1 + b_{i,i}) - (1 - b_{i,i})\sqrt{1 - 4\xi}}{2} \\ &= b_{i,i} + (1 - b_{i,i}) \sum_{j=1}^{\infty} C_{j-1,2} \xi^j \\ &= b_{i,i} + \sum_{k=1}^{d} u_{i,i}^{(k)} \left(\sum_{j=0}^{\infty} \frac{C_{j,2} \xi^j}{1 - b_{i,i}} \right) d_{i,i}^{(k)}. \end{aligned} \tag{13}$$

The probabilistic interpretation of the matrix V explains the above identity. Entry $v_{i,i}$ denotes the taboo probability that starting from state (J,i), with $J \neq \emptyset$, the process eventually returns to node J by visiting (J,i), under the taboo of its parent node. The first term $b_{i,i}$ gives the probability of this happening in a single transition, while the k-th term of the summation considers all the paths that start with a transition from (J,i) to $(J+k,i)$ and end by going from state $(J+k,i)$ to (J,i). Moreover, the j-th term of the inner summation accumulates the likelihood of all the paths from $(J+k,i)$ to $(J+k,i)$ containing exactly j transitions from a node to one of its children, under taboo of node J. Indeed, the probability of such a path equals

$$\left(\prod_{s=1}^{2j+1} b_{i,i}^{n_s}\right)\left(\prod_{t=1}^{j} u_{i,i}^{(m_t)} d_{i,i}^{(m_t)}\right),$$

for some $n_s \geq 0$ and $m_t \in \{1,\ldots,d\}$. By summing over all n_s and m_t and taking into account that the number of transitions from a node to one of its children must dominate the number from a node to its parent, this amounts to $C_{j,2}\left(\sum_{l=1}^{d} u_{i,i}^{(l)} d_{i,i}^{(l)}\right)^j/(1-b_{i,i})^{2j+1} = C_{j,2}\xi^j/(1-b_{i,i})$.

3.2. Superdiagonals of V

To recover the $(s+1)$-th diagonal of the G_k matrices, it suffices to compute the first $s+1$ diagonals of V. Similar to the previous section we define $\Delta(V,s)$ as

$$\Delta(V,s) = \begin{bmatrix} v_{1,1} & \cdots & v_{1,s} & & \\ & \ddots & \ddots & \ddots & \\ & & v_{m-s+1,m-s+1} & \ddots & v_{m-s+1,m} \\ & & & \ddots & \vdots \\ & & & & v_{m,m} \end{bmatrix}.$$

Notice, $\Delta(V,s)$ and $(I-\Delta(V,s))^{-1}$ are completely determined by the elements $v_{i,i+k}$, for $k=0,\ldots,s-1$ and the first s diagonals of $(I-\Delta(V,s))^{-1}$ coincide with those of $(I-V)^{-1}$.

Recalling the probabilistic interpretation of the matrix V, the entry $\{(I-V)^{-1}\}_{i,i+s}$ represents the probability that starting from state (J,i) at time $t_1 = 0$, with $J \neq \emptyset$, the chain eventually visits state $(J,i+s)$ under taboo of its parent node, just prior to making a transition to its parent node at time t' (or to the root node $\emptyset$). Node J is visited v times between time $t_1 = 0$ and t' for some $v \geq 1$ and we denote the time of the r-th visit as t_r. As with the argument presented to establish Eqn. (5), we can distinguish between the case where $N_{t_r} - N_{t_{r-1}} < s$ for all $r = 2,\ldots,v$ or $N_{t_r} - N_{t_{r-1}} = s$ for some $r \geq 2$. Similarly to

Eqn. (5), these observations lead to

$$\{(I-V)^{-1}\}_{i,i+s} = \{(I-\Delta(V,s))^{-1}\}_{i,i+s} + (1-v_{i,i})^{-1} v_{i,i+s}(1-v_{i+s,i+s})^{-1}, \qquad (14)$$

so that in light of Eqn. (12) and its relationship to G_k, it follows that

$$g^{(j)}_{i,i+s} = \{(I-\Delta(V,s))^{-1}D_j\}_{i,i+s} + (1-v_{i,i})^{-1} v_{i,i+s}(1-v_{i+s,i+s})^{-1} d^{(j)}_{i+s,i+s}. \qquad (15)$$

We obtain $v_{i,i+s}$ from Equation (12) as

$$\begin{aligned} v_{i,i+s} &= b_{i,i+s} + \sum_{j=1}^{d}\sum_{k=0}^{s} u^{(j)}_{i,i+k} g^{(j)}_{i+k,i+s} \qquad (16)\\ &= \frac{b_{i,i+s} + \sum_{j=1}^{d}\left(\sum_{k=1}^{s} u^{(j)}_{i,i+k} g^{(j)}_{i+k,i+s} + u^{(j)}_{i,i}\{(I-\Delta(V,s))^{-1}D_j\}_{i,i+s}\right)}{1-\sum_{j=1}^{d} u^{(j)}_{i,i}(1-v_{i,i})^{-1}(1-v_{i+s,i+s})^{-1} d^{(j)}_{i+s,i+s}}. \end{aligned}$$

In this manner we can compute the G matrices one diagonal at a time, resulting in the following $O(m^3 d)$ algorithm to compute the G_k matrices from $v_{1,1},\dots,v_{m,m}$:

- Start by computing the main diagonal of $(I-V)^{-1}$ and G_j for $j=1$ to d (in $O(md)$ time). At this point, the first diagonal of the matrices $(I-V)^{-1}$ and G_j, for $j=1,\dots,d$, has been computed.

- Next, for $s=1$ to $m-1$:
  - Compute the $(s+1)$-th diagonal of $(I-\Delta(V,s))^{-1}$ (containing the $(i,i+s)$-th entries) from the first s diagonals of the matrices $(I-V)^{-1}$ and V in $O(m^2)$ time, using

$$\begin{aligned} \{(I-\Delta(V,s))^{-1}\}_{i,i+s} &= \frac{\sum_{k=0}^{s-1}\{(I-\Delta(V,s))^{-1}\}_{i,i+k}\{\Delta(V,s)\}_{i+k,i+s}}{1-v_{i+s,i+s}}\\ &= \frac{\sum_{k=1}^{s-1}\{(I-V)^{-1}\}_{i,i+k} v_{i+k,i+s}}{1-v_{i+s,i+s}}. \end{aligned}$$

  - Compute the $(s+1)$-th diagonal of $(I-\Delta(V,s))^{-1}D_j$, for $j=1,\dots,d$ (containing the $(i,i+s)$-th entries) from the first $s+1$ diagonals of the matrix $(I-\Delta(V,s))^{-1}$ in $O(m^2 d)$ time.
  - Obtain $v_{i,i+s}$, for $i=1$ to $m-s$, using relation (16) in $O(m^2 d)$ time.
  - Finally, using relations (14) and (15), compute the elements $\{(I-V)^{-1}\}_{i,i+s}$ and $g^{(j)}_{i,i+s}$ from the $(s+1)$-th diagonal of $(I-\Delta(V,s))^{-1}$ and $(I-\Delta(V,s))^{-1}D_j$, for $j=1$ to d, in $O(md)$ time, that is, the $(s+1)$-th diagonal of $(I-V)^{-1}$ and G_j.

This results in an overall time complexity of $O(m^3 d)$ and memory complexity of $O(m^2 d)$. All of the (general purpose) iterative algorithms (see Bini et al. (2003)) to determine V require at least the same amount of memory and time *per iteration*, while the only algorithm with quadratic convergence—being the Newton iteration—requires the solution of a linear system with m^2 equations and m^2 unknowns *per iteration.*

4. Conclusion

In this paper we considered the highly structured subclass of triangular M/G/1-type and tree-like QBD Markov chains and provided an efficient direct solution of its key matrix equation. The realized time and memory complexity for the class of M/G/1-type Markov chains is $\mathrm{O}(m^3 N)$ and $\mathrm{O}(m^2 N)$, respectively, while for tree-like QBD Markov chains a time and memory complexity of $\mathrm{O}(m^3 d)$ and $\mathrm{O}(m^2 d)$, respectively, was achieved. These complexities are, in both cases, comparable to the best possible complexity of a *single iteration* when applying a general purpose iterative algorithm for solving such matrix equations. We also gave explicit expressions for the diagonal entries of the matrix solutions along with their probabilistic interpretations.

References

Adan, I.J.B.F., J. van der Wal. 1998. Combining make to order and make to stock. *OR Spektrum* **20** 73–81.

Adan, I.J.B.F., G. Weiss. 2006. Analysis of a simple Markovian re-entrant line with infinite supply of work under the LBFS policy. *Queueing Syst.* **54** 169–183.

Bini, D.A., G. Latouche, B. Meini. 2003. Solving nonlinear matrix equations arising in tree-like stochastic processes. *Linear Algebra Appl.* **366** 39–64.

Bini, D.A., G. Latouche, B. Meini. 2005. *Numerical methods for structured Markov chains*. Oxford University Press.

Bini, D.A., B. Meini, S. Steffé, B. Van Houdt. 2006a. Structured Markov chains solver: algorithms. *SMCtools Workshop*. ACM Press, Pisa, Italy.

Bini, D.A., B. Meini, S. Steffé, B. Van Houdt. 2006b. Structured Markov chains solver: software tools. *SMCtools Workshop*. ACM Press, Pisa, Italy.

Cohen, J.W. 1987. A two-queue, one-server model with priority for the longer queue. *Queueing Systems* **2** 261–283.

De Bruijn, N. G. 1981. *Asymptotic methods in analysis*. 3rd ed. Dover Publications Inc., New York.

Flatto, L. 1989. The longer queue model. *Probability in the Engineering and Informational Sciences* **3** 537–559.

Grassmann, W.K. 1993. The use of eigenvalues for finding equilibrium probabilities of certain markovian two-dimensional queueing problems. *INFORMS J. Comput* **15** 412–421.

Grassmann, W.K. 1994. Finding equilibrium probabilities of QBD processes by spectral methods when eigenvalues vanish. *Linear Algebra and its Applications* 207–223.

He, Q.M. 2000. Classification of Markov chains of M/G/1 type with a tree structure and its applications to queueing models. *Operations Research Letters* **26** 67–80.

Jaiswal, N.K. 1968. *Priority Queues*. Mathematics in Science and Engineering, Vol. 50, Academic Press, New York.

Latouche, G., V. Ramaswami. 1999. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM, Philadelphia.

Liu, D., Y. Zhao. 1996. Determination of explicit solution for a general class of markov processes. *Matrix-Analytic Methods in Stochastic Models, (S. R. Chakravarthy and A. S. Alfa (Editors))*. Marcel-Dekker, Inc., New York, 1343–357.

Neuts, M.F. 1981. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. The Johns Hopkins University Press.

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

Pitman, Jim. 1998. Enumerations of trees and forests related to branching processes and random walks. *Microsurveys in discrete probability (Princeton, NJ, 1997)*, *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, vol. 41. Amer. Math. Soc., Providence, RI, 163–180.

Riska, A., E. Smirni. 2002. Mamsolver: A matrix analytic methods tool. *TOOLS '02: Proceedings of the 12th International Conference on Computer Performance Evaluation, Modelling Techniques and Tools*. Springer-Verlag, London, UK, 205–211.

Takács, L. 1962. A generalization of the ballot problem and its application in the theory of queues. *J. Amer. Statist. Assoc.* **57** 327–337.

Takine, T., B. Sengupta, R.W. Yeung. 1995. A generalization of the matrix M/G/1 paradigm for Markov chains with a tree structure. *Stochastic Models* **11** 411–421.

Van Leeuwaarden, J.S.H., M.S. Squillante, E.M.M. Winands. 2009. Quasi-birth-and-death processes, lattice path counting, and hypergeometric functions. *To appear in Journal of Applied Probability* .

Van Leeuwaarden, J.S.H., E.M.M. Winands. 2006. Quasi-birth-and-death processes with an explicit rate matrix. *Stoch. Models* **22** 77–98.

Van Velthoven, J., B. Van Houdt, C. Blondia. 2006a. The impact of buffer finiteness on the loss rate in a priority queueing system. *Lecture Notes in Computer Science (LNCS 4054): Formal methods and stochastic models for performance evaluation, EPEW 2006*. Budapest, Hungary, 211–225.

Van Velthoven, J., B. Van Houdt, C. Blondia. 2006b. Transient analysis of tree-like processes and its application to random access systems. *ACM Sigmetrics Performance Evaluation Review* **34** 181–190.

Yeung, R.W., A.S. Alfa. 1999. The quasi-birth-death type Markov chain with a tree structure. *Stochastic Models* **15** 639–659.

Yeung, R.W., B. Sengupta. 1994. Matrix product-form solutions for Markov chains with a tree structure. *Adv. in Appl. Probab.* **26** 965–987.

Zhang, Z.G., N. Tian. 2004. An analysis of queueing systems with multi-task servers. *European Journal of Operational Research* **156** 375–389.