

# Chapter 2

## A Compressed Cyclic Reduction for QBD processes with Low-Rank Upper and Lower Transitions

Dario A. Bini, Paola Favati, and Beatrice Meini

### Introduction

A quasi-birth-and-death (QBD) process [10] is a Markov chain associated with a probability transition matrix

$$P = \begin{bmatrix} B_0 & B_1 & & 0 \\ B_{-1} & A_0 & A_1 & \\ & A_{-1} & A_0 & A_1 \\ & & A_{-1} & A_0 & \ddots \\ 0 & & & \ddots & \ddots \end{bmatrix}, \quad (2.1)$$

where  $B_0$ ,  $B_1$ , and  $A_i$ ,  $i = -1, 0, 1$ , are  $m \times m$  matrices,  $m$  being the phase space dimension. In the numerical solution of QBD processes, a crucial step is the computation of the minimal nonnegative solution  $G$  of the quadratic matrix equation

$$X = A_{-1} + A_0X + A_1X^2. \quad (2.2)$$

To this end, many numerical methods, with different properties, have been proposed in recent years (see, for instance, [2, 4–6]). Most of these algorithms are designed to deal with the general case where the block coefficients  $A_{-1}$ ,  $A_0$ , and  $A_1$  have no special structure.

---

D.A. Bini (✉) • B. Meini  
Dipartimento di Matematica, Università di Pisa, Italy  
e-mail: [bini@dm.unipi.it](mailto:bini@dm.unipi.it); [meini@dm.unipi.it](mailto:meini@dm.unipi.it)

P. Favati  
Istituto di Informatica e Telematica, CNR, Pisa, Italy  
e-mail: [paola.favati@iit.cnr.it](mailto:paola.favati@iit.cnr.it)

However, there are important applications where the block coefficients exhibit a structure that can be exploited to efficiently compute  $G$ . For instance, if  $A_{-1}$  has only one nonzero column, then  $G$  also has only one nonzero column, which can be computed using an explicit formula. We refer the reader to Neuts [12], to Sect. 10.4 of Latouche and Ramaswami [10], and to a few articles [7, 11, 15, 16] for some examples of queues where  $A_{-1}$  or  $A_1$  has rank one, and this property is used to provide a simple expression for  $G$  and for the steady state vector.

More recently, some interest has been demonstrated in specific cases where the blocks  $A_{-1}$  and/or  $A_1$  have many zero columns and rows, respectively. The interest in these cases is motivated by QBD processes with restricted transitions to higher (or lower) levels encountered in certain applications [8, 13, 14]. In particular, in [14] Pérez and Van Houdt exploit these properties of the matrix  $A_{-1}$  or  $A_1$  to formulate the QBD process in terms of an M/G/1- or GI/M/1-type Markov chain, where the block matrices have a size equal to the number of nonzero columns of  $A_{-1}$  or nonzero rows of  $A_1$ . In [8] Grassman and Tavakoli show how the structure of  $A_{-1}$  is used to reduce the computational cost of certain fixed-point iterations for computing  $G$ .

In this chapter, we consider the more general case where the matrices  $A_{-1}$  and  $A_1$  have small rank with respect to their size. This assumption is in particular satisfied in the case of restricted transitions to higher and lower levels. We exploit these rank properties to improve the efficiency of known algorithms for the computation of  $G$ . More specifically, we consider the cyclic reduction (CR) algorithm [2, 3] and show that if the sum of the ranks of  $A_{-1}$  and  $A_1$  is equal to  $r < m$ , then the CR step can be implemented with  $O(r^3)$  arithmetic operations (ops), instead of the  $O(m^3)$  ops required in the general case. This fact leads to a dramatic acceleration in the CPU time when  $r$  is much smaller than  $m$ . The same acceleration can be obtained for the logarithmic reduction (LR) algorithm of Latouche and Ramaswami [10]. The new algorithms keep the nice properties of numerical stability of the original algorithms because they avoid numerical cancellation. In fact, assuming that the low-rank decomposition of  $A_{-1}$  and  $A_1$  is formed by nonnegative matrices, we prove that the nonnegativity of the matrices involved in the algorithm is preserved and that all the operations consist of multiplications of nonnegative matrices and inversions of M-matrices. In the case where the low-rank decomposition is not given by nonnegative matrices, the algorithm can still be applied and keeps the same computational cost and convergence properties, but we cannot guarantee its numerical stability.

We consider also the case where only one matrix, among  $A_{-1}$  and  $A_1$ , has a small rank. We provide a version of CR where the number of ops required by the iteration is substantially smaller, even though it is still of the same order  $O(m^3)$ .

Our algorithms have been compared to the existing algorithms available in the literature. The many numerical experiments that we have performed show that they behave much better than the available methods. The larger the size of the matrix with respect to the rank, the larger the gain with respect to the customary algorithms.

The chapter is organized as follows. In the section “Cyclic Reduction for QBD processes” the customary CR algorithm is recalled together with its convergence and

applicability properties. The new algorithm for the case of low-rank downward and upward transitions is presented in the section “Case of Low-Rank Downward and Upward Transitions”; its numerical stability and computational complexity are also discussed. In the section “Case of Low-Rank Downward or Upward Transitions” we present an algorithm for the case where only one transition between the downward and upward transitions has low rank. Finally, in the section “Numerical Experiments” some numerical experiments are reported, showing the effectiveness of the proposed algorithms in terms of computational cost.

For the count of the arithmetic operations we use the following classical results: the LU factorization of a  $p \times p$  matrix  $A$  costs  $\frac{2}{3}p^3$  ops, the solution of  $q$  linear systems  $AX = B$ , given the LU factorization of the nonsingular matrix  $A$ , costs  $2p^2q$  ops, the inversion of  $A$  costs  $2p^3$  ops, and the multiplication of a  $p \times q$  matrix by a  $q \times s$  matrix costs  $2pqs$  ops.

Throughout the chapter we assume that the matrix  $P$  is irreducible, the matrix  $A_{-1} + A_0 + A_1$  is irreducible, and the QBD process is not null recurrent.

## Cyclic Reduction for QBD processes

The CR algorithm provides an effective method for computing the matrix  $G$ . It consists in generating a sequence of matrices  $A_i^{(k)}$ ,  $i = -1, 0, 1$ , and  $\hat{A}_0^{(k)}$  according to the following equations [2, 3]:

$$\begin{aligned} A_1^{(k+1)} &= A_1^{(k)}(I - A_0^{(k)})^{-1}A_1^{(k)}, \\ A_0^{(k+1)} &= A_0^{(k)} + A_1^{(k)}(I - A_0^{(k)})^{-1}A_{-1}^{(k)} + A_{-1}^{(k)}(I - A_0^{(k)})^{-1}A_1^{(k)}, \\ A_{-1}^{(k+1)} &= A_{-1}^{(k)}(I - A_0^{(k)})^{-1}A_{-1}^{(k)}, \\ \hat{A}_0^{(k+1)} &= \hat{A}_0^{(k)} + A_1^{(k)}(I - A_0^{(k)})^{-1}A_{-1}^{(k)}, \end{aligned} \quad (2.3)$$

with  $A_1^{(0)} = A_1$ ,  $A_0^{(0)} = A_0$ ,  $A_{-1}^{(0)} = A_{-1}$ ,  $\hat{A}_0^{(0)} = A_0$ , for  $k \geq 0$ , where we assume that  $I - A_0^{(k)}$  is nonsingular.

An approximation of  $G$  is provided by  $(I - \hat{A}_0^{(k)})^{-1}A_{-1}$ , for a sufficiently large value of  $k$ , since  $G = \lim_{k \rightarrow \infty} (I - \hat{A}_0^{(k)})^{-1}A_{-1}$ , according to the following convergence and applicability properties [2, Theorems 7.5, 7.6].

**Theorem 2.1.** *If the QBD processes is positive recurrent, then  $\det(I - A_0^{(k)}) \neq 0$  and  $\det(I - \hat{A}_0^{(k)}) \neq 0$ , so that the CR can be carried out with no breakdown; the matrices  $I - A_0^{(k)}$  and  $I - \hat{A}_0^{(k)}$  are (nonsingular)  $M$ -matrices,  $A_i^{(k)}$ ,  $i = -1, 0, 1$ , are nonnegative, and  $A_{-1}^{(k)} + A_0^{(k)} + A_1^{(k)}$  is stochastic for  $k \geq 0$ . Moreover, the following limits exist:*

$$\begin{aligned}
\lim_k A_0^{(k)} &= A_0^{(\infty)}, \quad \lim_k \widehat{A}_0^{(k)} = \widehat{A}_0^{(\infty)}, \\
\lim_k A_{-1}^{(k)} &= (I - A_0^{(\infty)})eg^T, \quad \lim_k A_1^{(k)} = 0, \\
\lim_k (I - \widehat{A}_0^{(k)})^{-1}A_{-1} &= G,
\end{aligned} \tag{2.4}$$

where  $\widehat{A}_0^{(\infty)}$  is the minimal nonnegative solution of

$$X = A_0 + A_1(I - X)^{-1}A_{-1}, \tag{2.5}$$

$e = (1, \dots, 1)^T$ ,  $g \geq 0$  is such that  $g^T G = g^T$ ,  $g^T e = 1$ , and all the sequences in equations (2.4) quadratically converge to their limits.

In the general case each step of the CR algorithm requires the solution of  $2m$  linear systems of size  $m$  and four matrix multiplications of order  $m$ , with an overall cost of  $\frac{38}{3}m^3$  ops. Due to the quadratic convergence, few steps are generally sufficient to reach the desired accuracy for the computation of  $G$ .

If the QBD process is transient, a similar applicability and convergence result can be given. We refer the reader to [2] for more details.

## Case of Low-Rank Downward and Upward Transitions

Now we consider the case where both downward and upward transitions have low rank, i.e., both the matrices  $A_{-1}$  and  $A_1$  have low rank. Denote by  $U_i$  and  $V_i$ ,  $i = -1, 1$ , matrices of size  $m \times r_i$  and  $r_i \times m$ , respectively, such that  $A_i = U_i V_i$ ,  $i = -1, 1$ , and set  $r = r_{-1} + r_1$ , where we assume that  $r$  is much smaller than  $m$ .

We show that the matrices  $A_i^{(k)}$ ,  $i = -1, 0, 1$ , generated at the  $k$ th step of the CR can be expressed in terms of the matrices  $U_i$  and  $V_i$  and in terms of small size matrices that depend on the step  $k$ .

To this end, define the following two matrices of size  $m \times r$  and  $r \times m$ , respectively:

$$U = [U_{-1} \mid U_1], \quad V = \begin{bmatrix} V_{-1} \\ V_1 \end{bmatrix}. \tag{2.6}$$

The following result holds:

**Theorem 2.2.** Assume that  $A_i = U_i V_i$ ,  $i = -1, 1$ , where  $U_i$  and  $V_i$ ,  $i = -1, 1$ , are matrices of size  $m \times r_i$  and  $r_i \times m$ , respectively. Let  $U$  and  $V$  be the matrices defined in (2.6). Then the sequences of matrices generated by cyclic reduction verify the following relations:

$$\begin{aligned}
A_1^{(k)} &= U H_1^{(k)} V, \\
A_0^{(k)} &= A_0 + U H_0^{(k)} V,
\end{aligned}$$

$$\begin{aligned} A_{-1}^{(k)} &= UH_{-1}^{(k)}V, \\ \widehat{A}_0^{(k)} &= A_0 + U\widehat{H}_0^{(k)}V, \end{aligned} \quad (2.7)$$

where the  $r \times r$  matrices  $H_i^{(k)}$ ,  $i = -1, 0, 1$ , and  $\widehat{H}_0^{(k)}$  are recursively defined, for  $k \geq 0$ , by

$$\begin{aligned} H_1^{(k+1)} &= H_1^{(k)}Q^{(k)}H_1^{(k)}, \\ H_0^{(k+1)} &= H_0^{(k)} + H_1^{(k)}Q^{(k)}H_{-1}^{(k)} + H_{-1}^{(k)}Q^{(k)}H_1^{(k)}, \\ H_{-1}^{(k+1)} &= H_{-1}^{(k)}Q^{(k)}H_{-1}^{(k)}, \\ \widehat{H}_0^{(k+1)} &= \widehat{H}_0^{(k)} + H_1^{(k)}Q^{(k)}H_{-1}^{(k)}, \end{aligned} \quad (2.8)$$

where

$$Q^{(k)} = (I - Q^{(0)}H_0^{(k)})^{-1}Q^{(0)}, \quad (2.9)$$

with  $Q^{(0)} = V(I - A_0)^{-1}U$ ,  $H_0^{(0)} = \widehat{H}_0^{(0)} = 0$ , and

$$H_{-1}^{(0)} = \begin{bmatrix} I_{r-1} & 0 \\ 0 & 0 \end{bmatrix}, \quad H_1^{(0)} = \begin{bmatrix} 0 & 0 \\ 0 & I_{r_1} \end{bmatrix}.$$

*Proof.* We prove the result by induction on  $k$ . For  $k = 0$ , Eqs. (2.7) hold by construction. Assume that (2.7)–(2.9) hold for a fixed  $k \geq 1$ , and prove them for  $k + 1$ . Consider  $A_1^{(k+1)}$ . From (2.3) it follows that  $A_1^{(k+1)} = A_1^{(k)}(I - A_0^{(k)})^{-1}A_1^{(k)}$ . By inductive hypothesis, one has  $A_1^{(k)} = UH_1^{(k)}V$ ; therefore  $A_1^{(k+1)} = UH_1^{(k+1)}V$ , where

$$H_1^{(k+1)} = H_1^{(k)}V(I - A_0^{(k)})^{-1}UH_1^{(k)}. \quad (2.10)$$

Since  $I - A_0^{(k)} = I - A_0 - UH_0^{(k)}V$ , by applying the Sherman–Woodbury–Morrison formula [9], we have

$$(I - A_0^{(k)})^{-1} = (I - A_0)^{-1} + (I - A_0)^{-1}U(T^{(k)})^{-1}H_0^{(k)}V(I - A_0)^{-1},$$

with  $T^{(k)} = I - H_0^{(k)}V(I - A_0)^{-1}U$ . The latter matrix is invertible since both the matrices  $I - A_0^{(k)}$  and  $I - A_0$  are invertible by Theorem 2.1. Observe that

$$V(I - A_0^{(k)})^{-1}U = Q^{(0)} + Q^{(0)}(I - H_0^{(k)}Q^{(0)})^{-1}H_0^{(k)}Q^{(0)} = (I - Q^{(0)}H_0^{(k)})^{-1}Q^{(0)}.$$

Hence  $V(I - A_0^{(k)})^{-1}U = Q^{(k)}$ , so that from (2.10) one finds that  $H_1^{(k+1)} = H_1^{(k)}Q^{(k)}H_1^{(k)}$ . We proceed similarly, for the remaining matrix sequences.  $\square$

According to the preceding theorem, the matrices  $A_{-1}^{(k)}$  and  $A_1^{(k)}$  have rank at most  $r$  and can be expressed by means of the  $r \times r$  matrices  $H_i^{(k)}$ ,  $i = -1, 1$ . Moreover, the matrices  $A_0^{(k)}$  and  $\widehat{A}_0^{(k)}$  are at most a rank  $r$  correction of the original matrix  $A_0$ . These properties allow one to carry out CR relying on (2.8) and (2.9) with a reduced computational cost.

Observe that  $G = \lim_k X^{(k)}$ , where  $X^{(k)}$  is the solution of the linear system  $(I - \widehat{A}_0^{(k)})X = A_{-1}$ . For the structure of the matrix  $\widehat{A}_0^{(k)}$ , by applying the Sherman–Woodbury–Morrison formula, we find that

$$(I - \widehat{A}_0^{(k)})^{-1}A_{-1} = (I - A_0)^{-1}A_{-1} + (I - A_0)^{-1}U(\widehat{T}^{(k)})^{-1}\widehat{H}_0^{(k)}V(I - A_0)^{-1}A_{-1}, \quad (2.11)$$

with  $\widehat{T}^{(k)} = I - \widehat{H}_0^{(k)}V(I - A_0)^{-1}U$ . From (2.11), since  $A_{-1} = U_{-1}V_{-1}$ , it follows that  $G$  has at most rank  $r_{-1}$ , and  $G = U_G V_{-1}$ , where  $U_G = \lim_{k \rightarrow \infty} U_G^{(k)}$  and

$$\begin{aligned} U_G^{(k)} &= (I - A_0)^{-1}U_{-1} + (I - A_0)^{-1}U(\widehat{T}^{(k)})^{-1}\widehat{H}_0^{(k)}V(I - A_0)^{-1}U_{-1} \\ &= (I + (I - A_0)^{-1}U(\widehat{T}^{(k)})^{-1}\widehat{H}_0^{(k)}V)(I - A_0)^{-1}U_{-1}. \end{aligned}$$

If the matrices  $U$  and  $V$  are nonnegative, then the sequences  $H_i^{(k)}$  are nonnegative as well, and their computation is numerically stable since it involves additions of nonnegative matrices and inversions of M-matrices, as stated by the following theorem.

**Theorem 2.3.** *Assume that the assumptions of Theorem 2.2 hold. If  $U \geq 0$  and  $V \geq 0$ , then  $Q^{(0)} \geq 0$ , and the sequences  $\{H_i^{(k)}\}_k$ ,  $i = -1, 0, 1$ , and  $\{\widehat{H}_0^{(k)}\}_k$  are such that  $\widehat{H}_0^{(k)} \geq 0$ ,  $H_i^{(k)} \geq 0$ ,  $i = -1, 0, 1$ ; moreover, the matrix  $I - Q^{(0)}H_0^{(k)}$  is a nonsingular M-matrix for any  $k \geq 1$ .*

*Proof.* The matrix  $Q^{(0)}$  is nonnegative since  $U, V \geq 0$  and  $I - A_0$  is a nonsingular M-matrix. To prove the remaining part of the theorem, we proceed by induction on  $k$ . For  $k = 1$ , one has  $\widehat{H}_0^{(1)} \geq 0$ ,  $H_i^{(1)} \geq 0$ ,  $i = -1, 0, 1$ , by construction since  $Q^{(0)} \geq 0$ . To show that  $S = I - Q^{(0)}H_0^{(1)}$  is a nonsingular M-matrix; consider the  $2 \times 2$  block matrix

$$B = \begin{bmatrix} I - A_0 & -UH_0^{(1)} \\ -V & I \end{bmatrix},$$

and observe that, since  $Q^{(0)} = V(I - A_0)^{-1}U$ , then  $S$  is the Schur complement of  $I - A_0$  in  $B$ . We show that  $B$  is a nonsingular M-matrix, therefore, since the Schur complement in a nonsingular M-matrix is a nonsingular M-matrix [1], also  $S$  is a nonsingular M-matrix. The matrix  $B$  is a Z-matrix since  $I - A_0$  is an M-matrix and  $UH_0^{(1)} \geq 0$ ,  $V \geq 0$ . Therefore, to show that  $B$  is an M-matrix, it is sufficient

to find a positive vector  $w$  such that  $Bw > 0$  [1]. For Theorem 2.1 the matrix  $I - A_0^{(1)}$  is a nonsingular M-matrix; therefore, there exists a positive vector  $r$  such that  $(I - A_0^{(1)})r = s > 0$  [1]. Since  $I - A_0^{(1)} = I - A_0 - UH_0^{(1)}V$  in view of (2.7), one finds that

$$B \begin{bmatrix} r \\ Vr + \varepsilon e \end{bmatrix} = \begin{bmatrix} s - \varepsilon UH_0^{(1)}e \\ \varepsilon e \end{bmatrix}, \quad (2.12)$$

where  $e$  is the vector of all ones. Since  $s > 0$ , we may find  $\varepsilon > 0$  such that  $s - \varepsilon UH_0^{(1)}e > 0$ . The vector  $w = \begin{bmatrix} r \\ Vr + \varepsilon e \end{bmatrix}$  is positive and, with this choice of  $\varepsilon$ , the right-hand side in (2.12) is positive; therefore,  $B$  is a nonsingular M-matrix. Assume that the properties hold for a fixed  $k \geq 1$ . Since  $I - Q^{(0)}H_0^{(k)}$  is a nonsingular M-matrix, its inverse is nonnegative; therefore,  $Q^{(k)} \geq 0$  and, from (2.8),  $\hat{H}_0^{(k+1)} \geq 0$ ,  $H_i^{(k+1)} \geq 0$ ,  $i = -1, 0, 1$ . To show that  $S = I - Q^{(0)}H_0^{(k+1)}$  is a nonsingular M-matrix, we proceed as in the case  $k = 1$  by observing that  $S$  is the Schur complement of  $I - A_0$  in the matrix

$$B^{(k+1)} = \begin{bmatrix} I - A_0 & -UH_0^{(k+1)} \\ -V & I \end{bmatrix}.$$

The latter matrix is a nonsingular M-matrix since it is a Z-matrix and, for a suitable  $\varepsilon > 0$ , one has  $B^{(k+1)} \begin{bmatrix} r \\ Vr + \varepsilon e \end{bmatrix} > 0$ , where  $r > 0$  is such that  $(I - A_0^{(k+1)})r > 0$ . Such a positive vector  $r$  exists since  $I - A_0^{(k+1)}$  is a nonsingular M-matrix for Theorem 2.1.  $\square$

Algorithm 1 reports a pseudocode that implements CR by exploiting the low-rank properties of the matrices. In the code, we use the Matlab notation where  $A(:, 1:s)$  is the matrix formed by the first  $s$  columns of the matrix  $A$ . The algorithm should stop the iterative process if the norm of  $A_{-1}^{(k)}$  or  $A_1^{(k)}$  is sufficiently small, but in practice, since the computation of the norm of  $H_i^{(k)}$ ,  $i = -1, 1$ , is less expensive, we stop the algorithm if  $\|H_{-1}^{(k)}\|_1 < \varepsilon$  or  $\|H_1^{(k)}\|_1 < \varepsilon$  for a fixed tolerance  $\varepsilon$ . On output, the algorithm provides an approximation to the  $m \times r_{-1}$  matrix  $U_G$  such that  $G = U_G V_{-1}$ .

To apply formulas (2.8) and (2.9), we must first compute the matrix  $Q^{(0)}$  by solving  $r$  linear systems with an  $m \times m$  matrix and by computing a multiplication between an  $r \times m$  matrix and an  $m \times r$  matrix, with an overall cost of  $\frac{2}{3}m^3 + 2m^2r + 2mr^2$  arithmetic operations. At each step the computation of  $Q^{(k)}$  requires one multiplication between two  $r \times r$  matrices and the solution of  $r$  linear systems of size  $r$ , with a cost of  $\frac{14}{3}r^3$  ops; the computation of  $H_i^{(k)}$  and  $\hat{H}_0^{(k)}$  requires six matrix multiplications of size  $r$ , with a cost of  $12r^3$ . Thus the overall arithmetic cost of the  $k$ th step is  $\frac{50}{3}r^3$  ops. The computational cost of recovering  $U_G$  amounts to  $\frac{14}{3}r^3 + 2r^2r_{-1} + 2mrr_{-1}$ .

**Algorithm 1** Low-Rank CR

---

```

Set  $k = 0$ 
Set  $H_0^{(0)} = \widehat{H}_0^{(0)} = 0_{r \times r}$ ,  $H_{-1}^{(0)} = \begin{bmatrix} I_{r-1} & 0 \\ 0 & 0 \end{bmatrix}$ ,  $H_1^{(0)} = \begin{bmatrix} 0 & 0 \\ 0 & I_{r_1} \end{bmatrix}$ 
Compute  $W = (I - A_0)^{-1}U$  by solving the linear system  $(I - A_0)X = U$ 
Compute  $Q^{(0)} = VW$ 
while  $\min\{\|H_{-1}^{(k)}\|_1, \|H_1^{(k)}\|_1\} \geq \varepsilon$  do
    Compute  $Y_1 = H_1^{(k)}Q^{(k)}$  and  $Y_{-1} = H_{-1}^{(k)}Q^{(k)}$ 
    Compute  $H_1^{(k+1)} = Y_1H_1^{(k)}$  and  $H_{-1}^{(k+1)} = Y_{-1}H_{-1}^{(k)}$ 
    Compute  $Z = Y_1H_{-1}^{(k)}$ 
    Compute  $H_0^{(k+1)} = H_0^{(k)} + Z + Y_{-1}H_1^{(k)}$  and  $\widehat{H}_0^{(k+1)} = \widehat{H}_0^{(k)} + Z$ 
    Compute  $F = Q^{(0)}H_0^{(k)}$ 
    Compute  $Q_0^{(k+1)}$  by solving the linear system  $(I - F)X = Q^{(0)}$ .
    Set  $k = k + 1$ 
end while
Compute  $F = \widehat{H}_0^{(k)}Q^{(0)}$ 
Solve the linear system  $(I - F)X = \widehat{H}_0^{(k)}$ .
return  $U_G = W(:, 1 : r_{-1}) + WXQ^{(0)}(:, 1 : r_{-1})$ 

```

---

To sum up, the algorithm consists of a preprocessing stage that costs  $\frac{2}{3}m^3$  ops, an iterative stage where each iteration costs  $\frac{50}{3}r^3$  ops, and a postprocessing stage that costs  $\frac{14}{3}r^3 + 2r^2r_{-1} + 2mrr_{-1}$ . It is important to point out that the cost of the iterative part is independent of the size  $m$  of the blocks.

Due to the interplay between CR and LR [2], a similar analysis can be carried out for the LR algorithm.

## Case of Low-Rank Downward or Upward Transitions

We consider now the case where only one matrix between  $A_{-1}$  and  $A_1$  has low rank. More specifically, assume that  $A_{-1}$  has low rank, that is,  $A_{-1} = U_{-1}V_{-1}$ , where  $U_{-1}$  and  $V_{-1}$  are  $m \times r_{-1}$  and  $r_{-1} \times m$  matrices with  $r_{-1}$  much smaller than  $m$ .

Also in this case the CR algorithm can be carried out with a computational cost lower than the cost of the general case. This improvement relies on the following properties:

- The matrix  $A_{-1}^{(k)}$  generated at the  $k$ th step of CR can be expressed in terms of the matrices  $U_{-1}$  and  $V_{-1}$  and in terms of small size matrices that depend on step  $k$ .
- The matrices  $A_0^{(k)}$  and  $\widehat{A}_0^{(k)}$  are corrections of rank at most  $2r_{-1}$  and  $r_{-1}$ , respectively, of the original matrix  $A_0$ .

More precisely, the following result provides the recursive equations for  $A_0^{(k)}$ ,  $\widehat{A}_0^{(k)}$ ,  $A_{-1}^{(k)}$ , while the equation for  $A_1^{(k)}$  is left unchanged.



**Theorem 2.4.** Let  $A_{-1} = U_{-1}V_{-1}$ . Then the sequences of matrices  $A_{-1}^{(k)}$ ,  $A_0^{(k)}$ , and  $\hat{A}_0^{(k)}$  generated by the CR verify the following relations:

$$\begin{aligned} A_{-1}^{(k)} &= U_{-1}K_{-1}^{(k)}V_{-1}, \\ A_0^{(k)} &= A_0 + U_{-1}W^{(k)} + Z^{(k)}V_{-1}, \\ \hat{A}_0^{(k)} &= A_0 + Z^{(k)}V_{-1}, \end{aligned} \quad (2.13)$$

where the matrices  $K_{-1}^{(k)}$  of size  $r_{-1} \times r_{-1}$ ,  $Z^{(k)}$  of size  $m \times r_{-1}$ , and  $W^{(k)}$  of size  $r_{-1} \times m$  are recursively defined, for  $k \geq 0$ , by

$$\begin{aligned} K_{-1}^{(k+1)} &= K_{-1}^{(k)}V_{-1}(I - A_0^{(k)})^{-1}U_{-1}K_{-1}^{(k)} \\ Z^{(k+1)} &= Z^{(k)} + A_1^{(k)}(I - A_0^{(k)})^{-1}U_{-1}K_{-1}^{(k)} \\ W^{(k+1)} &= W^{(k)} + K_{-1}^{(k)}V_{-1}(I - A_0^{(k)})^{-1}A_1^{(k)} \end{aligned} \quad (2.14)$$

with  $K_{-1}^{(0)} = I_{r_{-1}}$  and  $Z^{(0)} = W^{(0)} = 0$ .

*Proof.* We prove the result by induction on  $k$ . For  $k = 0$ , Eqs. (2.13) hold by construction. Assume that (2.13) and (2.14) hold for a fixed  $k \geq 1$ , and we prove the result for  $k + 1$ . Consider first  $A_{-1}^{(k+1)}$ . From (2.3) one has  $A_{-1}^{(k+1)} = A_{-1}^{(k)}(I - A_0^{(k)})^{-1}A_{-1}^{(k)}$ . By inductive hypothesis,  $A_{-1}^{(k)} = U_{-1}K_{-1}^{(k)}V_{-1}$ ; therefore,  $A_{-1}^{(k+1)} = U_{-1}K_{-1}^{(k+1)}V_{-1}$ , where  $K_{-1}^{(k+1)} = K_{-1}^{(k)}V_{-1}(I - A_0^{(k)})^{-1}U_{-1}K_{-1}^{(k)}$ .

Consider now  $A_0^{(k+1)}$ . From (2.3) one has  $A_0^{(k+1)} = A_0^{(k)} + A_1^{(k)}(I - A_0^{(k)})^{-1}A_{-1}^{(k)} + A_{-1}^{(k)}(I - A_0^{(k)})^{-1}A_1^{(k)}$ . By inductive hypothesis,  $A_0^{(k)} = A_0 + U_{-1}W^{(k)} + Z^{(k)}V_{-1}$ ; therefore,

$$\begin{aligned} A_0^{(k+1)} &= A_0 + U_{-1}(W^{(k)} + K_{-1}^{(k)}V_{-1}(I - A_0^{(k)})^{-1}A_1^{(k)}) \\ &\quad + (Z^{(k)} + A_1^{(k)}(I - A_0^{(k)})^{-1}U_{-1}K_{-1}^{(k)})V_{-1}, \end{aligned}$$

that is,  $A_0^{(k+1)} = A_0 + U_{-1}W^{(k+1)} + Z^{(k+1)}V_{-1}$ . We proceed similarly for the remaining matrix sequences.  $\square$

Consider the matrices

$$U^{(k)} = [U_{-1} \mid Z^{(k)}], \quad V^{(k)} = \begin{bmatrix} W^{(k)} \\ V_{-1} \end{bmatrix}$$

of size  $m \times 2r_{-1}$  and  $2r_{-1} \times m$ , respectively. Then we can write

$$A_0^{(k)} = A_0 + U^{(k)}V^{(k)},$$

---

**Algorithm 2** Downward Low-Rank CR
 

---

```

Set  $k = 0$ 
Set  $K_{-1}^{(0)} = I_{r-1}$ ,  $Z^{(0)} = 0_{m \times r-1}$ ,  $W^{(0)} = 0_{r-1 \times m}$ 
Compute  $B = (I - A_0)^{-1}$ ,  $N_{-1} = BU_{-1}$  and  $Q_{-1} = V_{-1}N_{-1}$ 
while  $\min\{\|K_{-1}^{(k)}\|_1, \|A_1^{(k)}\|_1\} \geq \varepsilon$  do
  Compute  $F = U_{-1}K_{-1}^{(k)}$ ,  $G = (I - A_0^{(k)})^{-1}F$ ,  $M = K_{-1}^{(k)}V_{-1}$ 
  Compute  $K_{-1}^{(k+1)} = MG$ 
  Compute  $L = (I - A_0^{(k)})^{-1}A_1^{(k)}$  and  $A_1^{(k+1)} = A_1^{(k)}L$ 
  Compute  $Z^{(k+1)} = Z^{(k)} + A_1^{(k)}G$  and  $W^{(k+1)} = W^{(k)} + ML$ 
  Compute  $N = BZ^{(k+1)}$ 
  Compute  $R_{11} = W^{(k+1)}N_{-1}$ ,  $R_{12} = W^{(k+1)}N$  and  $R_{22} = V_{-1}N$ 
  Set  $R = \begin{bmatrix} R_{11} & R_{12} \\ Q_{-1} & R_{22} \end{bmatrix}$ ,  $V^{(k+1)} = \begin{bmatrix} W^{(k+1)} \\ V_{-1} \end{bmatrix}$ 
  Compute  $S = (I - R)^{-1}V^{(k+1)}$  by solving the linear system  $(I - R)X = V^{(k+1)}$ 
  Compute  $(I - A_0^{(k+1)})^{-1} = B + [N_{-1} \mid N]SB$ 
  Set  $k = k + 1$ 
end while
Compute  $F = (I - R_{22})^{-1}Q_{-1}$  by solving the linear system  $(I - R_{22})X = Q_{-1}$ 
return  $U_G = N_{-1} + NF$ 

```

---

and from the Sherman–Woodbury–Morrison formula we have

$$(I - A_0^{(k)})^{-1} = B + BU^{(k)}(I - V^{(k)}BU^{(k)})^{-1}V^{(k)}B, \quad (2.15)$$

where  $B = (I - A_0)^{-1}$ . Moreover, also in this case, the desired solution  $G$  is given by  $G = \lim_k X^{(k)}$ , where  $X^{(k)}$  is the solution of the linear system  $(I - \hat{A}_0^{(k)})X = A_{-1}$ . It can be expressed as  $G = U_G V_{-1}$ , where  $U_G = \lim_{k \rightarrow \infty} U_G^{(k)}$  and

$$\begin{aligned} U_G^{(k)} &= (I - A_0)^{-1}U_{-1} + (I - A_0)^{-1}Z^{(k)}(\hat{T}^{(k)})^{-1}V_{-1}(I - A_0)^{-1}U_{-1} \\ &= (I + (I - A_0)^{-1}Z^{(k)}(\hat{T}^{(k)})^{-1}V_{-1})(I - A_0)^{-1}U_{-1}, \end{aligned} \quad (2.16)$$

with  $\hat{T}^{(k)} = I - V_{-1}(I - A_0)^{-1}Z^{(k)}$ .

Algorithm 2 shows a pseudocode that implements CR by exploiting the low-rank properties of the matrices in the case of low-rank downward transitions. Also in this case, we have chosen as a stopping criterion the condition  $\|K_{-1}^{(k)}\|_1 < \varepsilon$  or  $\|A_1^{(k)}\|_1 < \varepsilon$  for a fixed tolerance  $\varepsilon$ . On output, the algorithm provides an approximation to the  $m \times r_{-1}$  matrix  $U_G$  such that  $G = U_G V_{-1}$ .

Computing the matrices  $B$ ,  $N_{-1}$ , and  $Q_{-1}$  requires an  $m \times m$  matrix inversion, a multiplication between an  $m \times m$  matrix, and an  $m \times r_{-1}$  matrix and a multiplication between an  $r_{-1} \times m$  matrix and an  $m \times r_{-1}$ , at an overall cost of  $2(m^3 + m^2 r_{-1} + m r_{-1}^2)$  arithmetic operations. At each step the updating of  $K_{-1}^{(k)}$  requires four matrix multiplications at a cost of  $2(m^2 r_{-1} + 3m r_{-1}^2)$  ops, the updating of  $A_1^{(k)}$  requires two

$m \times m$  matrices at a cost of  $4m^3$  ops, the updating of matrices  $Z^{(k)}$  and  $W^{(k)}$  requires two matrix multiplications at a cost of  $4m^2r_{-1}$  ops, and, finally, the updating of  $(I - A_0^{(k)})^{-1}$  according to formulas (2.15) requires four matrix multiplications at a cost of  $2(m^2r_{-1} + 3mr_{-1}^2)$  ops, the solution of  $m$  linear systems of size  $2r_{-1}$ , at a cost of  $\frac{16}{3}r_{-1}^3 + 8mr_{-1}^2$  ops, and two further matrix multiplications at a cost of  $8m^2r_{-1}$  ops. Thus the overall arithmetic cost of the  $k$ th step is  $4m^3 + 16m^2r_{-1} + 20mr_{-1}^2 + \frac{16}{3}r_{-1}^3$  ops. The computation of  $U_G$  according to formulas (2.16) requires the solution of  $r_{-1}$  linear systems of size  $r_{-1}$  at a cost of  $\frac{8}{3}r_{-1}^3$  ops and a matrix multiplication at a cost of  $2mr_{-1}^2$  ops.

Observe that, even in this case, the algorithm consists of a preprocessing stage, an iterative stage, and a postprocessing stage. However, unlike the case where both  $A_{-1}$  and  $A_1$  have low rank, each step of the iterative stage has a cost dependent on the size  $m$  of the blocks. On the other hand, the number of ops needed at each step of the iterative stage is smaller, by a fixed constant, w.r.t. the cost of the general CR.

If  $A_1$ , instead of  $A_{-1}$ , were of low rank, then we might apply a similar technique by switching the role of  $A_{-1}$  with that of  $A_1$ .

## Numerical Experiments

We report some numerical experiments that show the gain, in terms of computational time, of the proposed algorithms. We performed the experiments using Matlab on an Intel Xeon 2.80-GHz processor.

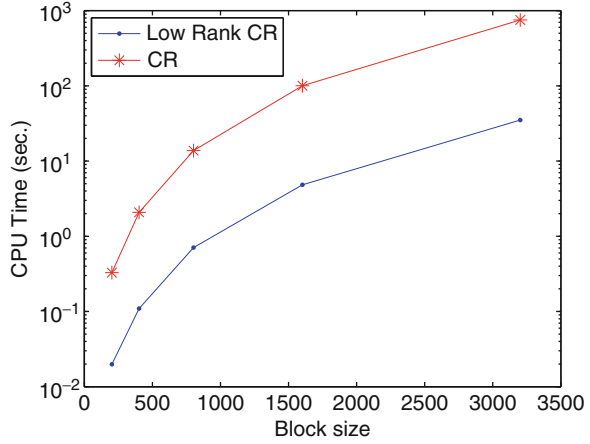
The first example is a QBD process where the matrices  $A_{-1}$  and  $A_1$  have rank 2 and 3, respectively. The entries are randomly generated in such a way that the matrices  $U_i$  and  $V_i$ ,  $i = -1, 1$ , are nonnegative. Figure 2.1 reports the CPU time (in seconds) needed by CR, exploiting and without exploiting the low-rank properties, for different values of  $m$  leaving unchanged the rank of  $A_{-1}$  and  $A_1$ . It is clear from the figure that our algorithm outperforms the general algorithm already for small values of block size  $m$ .

The second example is a PH/PH/1 queue, where  $A_{-1} = (t\alpha) \otimes S$ ,  $A_0 = T \otimes S + (t\alpha) \otimes (s\beta)$ ,  $A_1 = T \otimes (s\beta)$ , and

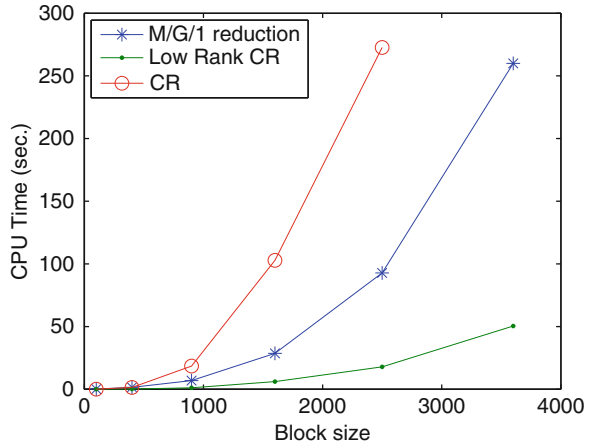
$$T = \begin{bmatrix} 0.5 & 0.4 & & \\ & 0.5 & \ddots & \\ & & \ddots & 0.4 \\ & & & 0.5 \end{bmatrix}, \quad S = \begin{bmatrix} 0.4 & 0.3 & & \\ & 0.4 & \ddots & \\ & & \ddots & 0.3 \\ & & & 0.4 \end{bmatrix}$$

are  $n \times n$  matrices,  $t = e - Te$ ,  $s = e - Se$ ,  $\alpha = \beta = (1, 0, \dots, 0)$ . In this case  $U_{-1} = t \otimes I_n$ ,  $V_{-1} = \alpha \otimes I_n$ ,  $U_1 = I_n \otimes s$ , and  $V_1 = I_n \otimes \beta$ . Therefore, the size of the blocks is  $m = n^2$ , while the blocks  $A_{-1}$  and  $A_1$  have rank  $n$ . With this choice of the vectors  $\alpha$  and  $\beta$ , the matrices  $A_{-1}$  and  $A_1$  have  $n$  nonzero columns.

**Fig. 2.1** CPU time for CR and low rank CR, for matrices with fixed rank



**Fig. 2.2** CPU time of CR, low-rank CR, and M/G/1 reduction for a PH/PH/1 queue



Since the matrix  $A_{-1}$  has  $n$  nonzero columns, we may apply the algorithm proposed in [14] for solving QBD processes with restricted transitions to lower levels. This algorithm, which we call an *M/G/1 reduction algorithm*, consists in solving the QBD process by solving an M/G/1-type Markov chain, with block matrices of size equal to the number  $s$  of nonzero columns of  $A_{-1}$ , followed by the solution of a Stein matrix equation at a cost of  $O((m-s)^3)$  ops. The algorithm of [14] can be applied also to the case where  $A_1$  has a few nonzero rows; in this case the QBD process is reduced to a GI/M/1-type Markov chain.

Figure 2.2 reports the CPU time needed by customary CR, by low-rank CR, and by the M/G/1 reduction algorithm of [14]. Observe that the algorithm of [14] provides an improvement with respect to the general CR and that it is our algorithm that has the minimum computational cost.

**Table 2.1** CPU time for low-rank CR and customary CR

$m$	Low-rank CR				CR
	Preproc.	CR	$G$	Total	
100	0	1.0e-02	0	1.0e-02	5.0e-02
400	6.0e-02	1.0e-02	6.0e-02	1.3e-01	2.3e+00
900	5.4e-01	1.0e-02	4.3e-01	9.8e+00	2.4e+01
1,600	2.6e+00	1.0e-02	2.0e+00	4.6e+00	1.1e+02
2,500	9.3e+00	1.0e-02	7.5e+00	1.7e+01	4.8e+02
3,600	2.7e+01	1.0e-02	2.2e+01	4.9e+01	1.3e+03

The third example consists of a QBD process where the coefficients are defined by

$$\begin{aligned}
 A_{-1} &= C_0 \otimes D_1, \\
 A_0 &= C_0 \otimes D_0 + C_1 \otimes D_1, \\
 A_1 &= C_1 \otimes D_0,
 \end{aligned}$$

where  $(C_0, C_1)$  and  $(D_0, D_1)$  define two Markov arrival processes (MAP's), i.e.,  $C_i, D_i, i = 0, 1$ , are nonnegative matrices such that  $C_0 + C_1$  and  $D_0 + D_1$  are stochastic. If  $C_i$  and  $D_i, i = 0, 1$ , are low-rank matrices, then also  $A_{-1}$  and  $A_1$  are low rank. More specifically, if

$$\begin{aligned}
 C_0 &= U_{C_0} V_{C_0}, \quad C_1 = U_{C_1} V_{C_1}, \\
 D_0 &= U_{D_0} V_{D_0}, \quad D_1 = U_{D_1} V_{D_1},
 \end{aligned}$$

where  $U_{C_0}, U_{C_1}, U_{D_0}, U_{D_1}$  are  $n \times h_1, n \times h_2, n \times h_3, n \times h_4$  matrices, respectively, then  $A_i = U_i V_i, i = -1, 1$ , where

$$\begin{aligned}
 U_{-1} &= U_{C_0} \otimes U_{D_1}, \quad V_{-1} = V_{C_0} \otimes V_{D_1} \\
 U_1 &= U_{C_1} \otimes U_{D_0}, \quad V_1 = V_{C_1} \otimes V_{D_0}.
 \end{aligned}$$

Therefore, the blocks  $A_i$  have size  $m = n^2$ , while  $A_{-1}$  and  $A_1$  have rank  $r_{-1} = h_1 h_4$  and  $r_1 = h_2 h_3$ , respectively. We set  $h_1 = 5, h_2 = 3, h_3 = 7$ , and  $h_4 = 4$  and tried several values of  $n$ . Table 2.1 reports, for different values of  $m = n^2$ , the CPU time of low-rank CR, where we have distinguished the time needed in the preprocessing stage, the time needed by CR, the time to recover  $U_G$ , and the total time; we report also the overall time needed by customary CR. In all the tests both low-rank CR and customary CR performed the same number of iterations and provided an approximation of  $G$  to the same accuracy, i.e., having a residual error around  $10^{-15}$ .

The low-rank CR is faster than general CR, and the major cost of low-rank CR is due to the pre- and postprocessing stages, that is, the computation of  $W$  and  $Q^{(0)}$  (preprocessing) and the computation of  $U_G$  (postprocessing) in Algorithm 1. The remaining computation has a negligible cost independent of the size  $m$ .

The fourth example is the overflow queueing system described in Example 5.3 of [14]. The queueing system consists of two queues, the first having a finite buffer of size  $C$  and the second having an infinite buffer. Customers arriving at the first queue are served on a first-come, first-served (FCFS) basis by a single server, and the customers that find the buffer full are sent to the second queue. The second queue receives only overflow arrivals from the first queue and serves them in FCFS order with a single server. The arrival process at the first queue is a MAP characterized by  $(m_a, D_0, D_1)$ , and the service time follows a PH distribution characterized by the parameters  $(m_1, \alpha, T)$ . The service time of the second queue follows a PH distribution with parameters  $(m_2, \beta, S)$ . The arrival process at the second queue can be represented by a MAP with parameters  $(m_0, C_0, C_1)$ , where  $m_0 = (C + 1)m_a m_1$  and

$$C_0 = \begin{bmatrix} D_0 \otimes I & D_1 \otimes I & 0 & \dots & \dots & 0 \\ I \otimes t\alpha & D_0 \oplus T & D_1 \otimes I & \ddots & \ddots & \vdots \\ 0 & I \otimes t\alpha & D_0 \oplus T & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & D_0 \oplus T & D_1 \otimes I \\ 0 & \dots & \dots & 0 & I \otimes t\alpha & D_0 \oplus T \end{bmatrix}, \quad C_1 = \begin{bmatrix} 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & D_1 \otimes I \end{bmatrix},$$

with  $t = -Te$ .

This queueing system can be described by a continuous-time QBD process, where the level represents the number of customers in the second queue. The blocks are  $A_{-1} = I_{m_0} \otimes s\beta$ ,  $A_0 = C_0 \otimes I_{m_2} + I_{m_0} \otimes S$ ,  $A_1 = C_1 \otimes I_{m_2}$ , where  $s = -Se$ , with size  $m = m_0 m_2$ . The number of nonzero rows in  $A_1$  is  $r = m_a m_1 m_2$ .

The matrices  $A_{-1}$  and  $A_1$  can be decomposed as the product of matrices of rank  $m_0$  and  $r$ , respectively, as

$$A_{-1} = (I_{m_0} \otimes s)(I_{m_0} \otimes \beta), \quad A_1 = \left( \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I \end{bmatrix} \otimes I_{m_2} \right) ([0 \dots 0 \ D_1 \otimes I] \otimes I_{m_2}).$$

The continuous-time QBD process is transformed into a discrete-time QBD process using standard uniformization.

We have chosen the same parameters as in [14], i.e., the arrival process at the first queue has an arrival rate and squared coefficient of variation (SCV) equal to five, while the service time has mean 1 and SCV equal to 2. Also for the second queue service times have SCV equal to two. Therefore, the first queue is heavily loaded and many customers are overflowed to the second queue. The load of the second queue is a parameter  $\rho_2$  that can vary and the buffer capacity  $C$ .

**Table 2.2** CPU time for low-rank CR and GI/M/1 reduction for overflow queueing system, with  $C = 20$ 

$\rho_2$	Low-rank CR				GI/M/1 reduction				
	Preproc.	CR	$G$	Total	Bandwidth	Construction	CR	Stein eq.	Total
0.2	0	3.0e-02	2.0e-02	5.0e-02	1,296	1.9e+00	1.9e+01	1.9e-01	2.1e+01
0.3	0	4.0e-02	2.0e-02	6.0e-02	881	9.9e-01	5.2e+01	2.0e-01	6.4e+00
0.4	1.0e-02	4.0e-02	1.0e-02	6.0e-02	671	6.1e-01	5.3e+00	1.9e-01	6.1e+00
0.5	1.0e-02	5.0e-02	1.0e-02	7.0e-02	544	4.3e-01	2.5e+00	1.9e-01	3.1e+00
0.6	1.0e-02	5.0e-02	1.0e-02	7.0e-02	459	3.2e-01	1.5e+00	2.0e-01	2.1e+00
0.7	1.0e-02	5.0e-02	1.0e-02	7.0e-02	399	2.7e-01	1.5e+00	2.0e-01	2.0e+00
0.8	1.0e-02	5.0e-02	1.0e-02	7.0e-02	352	2.2e-01	1.5e+00	1.9e-01	1.9e+00
0.9	1.0e-02	6.0e-02	1.0e-02	8.0e-02	316	1.9e-01	1.5e+00	1.9e-01	1.9e+00

We have set  $m_a = m_1 = m_2 = 2$  and  $C = 20$ . In Table 2.2 we report the CPU time needed by low-rank CR, and by the algorithm of [14] based on the reduction to a GI/M/1-type Markov chain for different values of  $\rho_2$ . For the latter algorithm we have reported the bandwidth of the GI/M/1-type Markov chain, the time needed to construct its blocks, the time needed by CR, the time to solve the Stein equation, and the total time. The higher computational time of the algorithm of [14] is mainly due to the large bandwidth of the GI/M/1-type Markov chain; in fact, as observed in [14], the bandwidth is larger when the load  $\rho_2$  is closer to zero.

**Acknowledgements** The authors wish to thank the anonymous referees, and Juan Pérez and Benny Van Houdt for providing the code to construct the matrices  $A_i$ ,  $i = -1, 0, 1$ , for the overflow queueing model example.

## References

1. Berman, A., Plemmons, R.J.: Nonnegative matrices in the mathematical sciences. In: Classics in Applied Mathematics, vol. 9. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (1994). Revised reprint of the 1979 original
2. Bini, D.A., Latouche, G., Meini, B.: In: Numerical Methods for Structured Markov Chains. Numerical Mathematics and Scientific Computation. Oxford University Press, New York (2005). Oxford Science Publications
3. Bini, D.A., Meini, B.: The cyclic reduction algorithm: From Poisson equation to stochastic processes and beyond. In memoriam of Gene H. Golub. Numer. Algorithms **51**(1), 23–60 (2009)
4. Bini, D.A., Meini, B., Steffé, S., Van Houdt, B.: Structured Markov chains solver: Algorithms. In: SMCTools '06: Proceeding from the 2006 workshop on Tools for solving structured Markov chains Article No. 13 ACM New York, NY, USA (2006)
5. Bini, D.A., Meini, B., Steffé, S., Van Houdt, B.: Structured Markov chains solver: Software tools. In: SMCTools '06: Proceeding from the 2006 workshop on Tools for solving structured Markov chains Article No. 14 Publisher: ACM New York, NY, USA (2006)

6. Bini, D.A., Meini, B., Steffé, S., Van Houdt, B.: Structured Markov chains solver: Tool extension. In: VALUETOOLS '09: Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools Article No. 20 Publisher: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, Belgium, Belgium (2009)
7. Carroll, J.L., Van de Liefvoort, A., Lipsky, L.: Solutions of M/G/1/N-type loops with extensions to M/G/1 and GI/M/1 queues. *Oper. Res.* **30**(3), 490–514 (1982)
8. Grassmann, W.K., Tavakoli, J.: Comparing some algorithms for solving QBD processes exhibiting special structures. *INFOR: Inf. Syst. Oper. Res.* **48**(3), 133–141 (2010)
9. Hogben, L. (ed.): In: Handbook of Linear Algebra. Discrete Mathematics and Its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton (2007). Associate editors: Richard Brualdi, Anne Greenbaum, and Roy Mathias
10. Latouche, G., Ramaswami, V.: In: Introduction to Matrix Analytic Methods in Stochastic Modeling. ASA-SIAM Series on Statistics and Applied Probability. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (1999)
11. Neuts, M.F.: Explicit steady-state solutions to some elementary queueing models. *Oper. Res.* **30**(3), 480–489 (1982)
12. Neuts, M.F.: Matrix-Geometric Solutions in Stochastic Models. Dover, New York (1994). An algorithmic approach, Corrected reprint of 1981 original
13. Pérez, J.F., Van Houdt, B.: The M/G/1-type Markov chain with restricted transitions and its application to queues with batch arrivals. *Probability in the Engineering and Informational Sciences* **25**(04), 487–517 (2011)
14. Pérez, J.F., Van Houdt, B.: Quasi-birth-and-death processes with restricted transitions and its applications. *Perform. Eval.* **68**(2), 126–141 (2011)
15. Ramaswami, V., Latouche, G.: A general class of Markov processes with explicit matrix-geometric solution. *OR Spektrum* **8**, 209–218 (1986)
16. Ramaswami, V., Lucantoni, D.: Algorithmic analysis of a dynamic priority queue. In: Applied Probability—Computer Science: The Interface, vol. II, pp. 157–204. Birkhäuser, Basel (1981)



Matrix-Analytic Methods in Stochastic Models

Latouche, G.; Ramaswami, V.; Sethuraman, J.; Sigman, K.; Squillante, M.S.; Yao, D. (Eds.)

2013, XIV, 258 p., Hardcover

ISBN: 978-1-4614-4908-9