J. Appl. Prob. **53,** 106–120 (2016)
*doi:*10.1017/jpr.2015.12
© *Applied Probability Trust* 2016

A COMPARATIVE ANALYSIS OF THE SUCCESSIVE LUMPING AND THE LATTICE PATH COUNTING ALGORITHMS

MICHAEL N. KATEHAKIS,* *Rutgers University*
LAURENS C. SMIT ** *** AND
FLOSKE M. SPIEKSMA,** **** *Leiden University*

Abstract

This paper provides a comparison of the successive lumping (SL) methodology developed in Katehakis *et al.* (2015) with the popular lattice path counting (Mohanty (1979)) in obtaining rate matrices for queueing models, satisfying the specific quasi birth and death structure as in Van Leeuwaarden *et al.* (2009) and Van Leeuwaarden and Winands (2006). The two methodologies are compared both in terms of applicability requirements and numerical complexity by analyzing their performance for the same classical queueing models considered in Van Leeuwaarden *et al.* (2009). The main findings are threefold. First, when both methods are applicable, the SL-based algorithms outperform the lattice path counting algorithm (LPCA). Second, there are important classes of problems (for example, models with (level) nonhomogenous rates or with finite state spaces) for which the SL methodology is applicable and for which the LPCA cannot be used. Third, another main advantage of SL algorithms over lattice path counting is that the former includes a method to compute the steady state distribution using this rate matrix.

Keywords: Steady state analysis; queueing; successive lumping

2010 Mathematics Subject Classification: Primary 60K25
Secondary 68M20

1. Introduction

Two-dimensional Markov chains arise as a natural way in which to model various real-life applications. In particular, many queueing models possess this structure and it is even possible that a more complex, higher-dimensional queueing model can be decomposed into various two-dimensional Markov processes. For various queueing models we refer the reader to [1]–[3], [5], [7], [9], [25], [27], [31], and [34]. Other areas in which these processes will arise outside queueing are, for example, inventory models [17], reliability [14], [15], and pricing models. In this paper we are particularly interested in a comparison of the new successive lumping (SL) methodology developed in [19] with the popular lattice path counting [22] in obtaining rate matrices for queueing models, as in [29] and [30]. The two methodologies are compared both in terms of applicability requirements and numerical complexity by analyzing their performance for the same classical queueing models considered in [30]. In all these models, the objective is to calculate the steady state distribution of a pertinent quasi birth and death (QBD) process (i.e. a

Received 22 December 2014; revision received 17 February 2015.

* Postal address: Department of Management Science and Information Systems, Rutgers University, 100 Rockafeller Road, Piscataway, NJ 08854, USA. Email address: mnk@rutgers.edu
** Postal address: Mathematisch Instituut, Universiteit Leiden, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands.
*** Email address: lsmit@math.leidenuniv.nl
**** Email address: spieksma@math.leidenuniv.nl

two-dimensional Markov chain with a transition generator matrix Q that contains nonzero rates only for transitions to the 'left' and to the 'right' in every state) that describes the evolution of the state of the system in time.

The main method that is used to analyze QBD processes is based on expressing the stationary probabilities of states of one level in terms of those of its previous levels. This is done with the aid of a rate matrix R, which is the basis of the matrix-geometric solution introduced by Neuts [24]. For general level-independent QBD processes, it is known that R satisfies a matrix-quadratic equation. Algorithms for solving this equation were given in [20] and [24]. A current state-of-the-art software implementing quadratically convergent algorithms with a number of speed-up features is described in [4]. A general algorithm for the level-independent case can be found in [6] and a discussion of the quasi skip-free case in [21].

There are various methods that make use of a special structure of the transition rate matrix Q in order to provide efficient computation procedures for the rate matrix R. Such a procedure is available in the case in which the 'down matrix' of Q is a product of a row and a column vector. For other procedures that explicitly calculate a rate matrix we refer the reader to [23] and [28]. In recent studies ([29], [30]) the authors have used lattice path counting methods to directly compute the rate matrix for certain QBD processes that arise in queueing models. For example, a priority queue model has been analyzed by this method, but also with other techniques; see, for example, [11] and the references therein. The idea of counting the number of paths on a lattice ([10], [22]) has been used in many fields of applied probability; see [26].

A new alternative method to compute the rate matrix for certain QBD processes is based on the SL procedure introduced in [16]. It was employed in [19] to obtain explicit solutions for 'rate sets' for large classes of quasi skip-free (QSF) processes, the so-called down entrance state (DES) and restart entrance state processes. The SL approach differs from the previous mentioned works by its distinct method of derivation and its applicability to models with infinite state spaces and models that are outside the QSF framework. However, it should be noted that algorithms given in [6], [11], and [20] can be used on other, more general (in terms of down-transitions) processes. The advantages of using the SL approach are described in [19]. Although the nature of a path counting based method and the SL-based method are very different, a comparison can be performed, since they both rely on the absence of certain kinds of transition. Herein, we compare the method introduced in [30] with the one based on the SL approach of [19].

The main contribution of this paper is to provide a clear comparison between SL-based methods and the lattice path counting algorithm (LPCA), introduced in [30], in computational complexity and applicability. First, it is shown that the SL methodology yields algorithms that are faster than the counting algorithm. Second, we show that SL-based procedures are applicable to many of the queueing models discussed in previous papers, and even to models with finite state spaces or with nonhomogenous transition rate structures and to models with a QSF structure; see [19]. However, there seem to exist some artificial queueing models that do not possess the SL property, for which an LPCA is applicable. Finally, in this paper we continue the work of [19], and we specialize its results to homogenous QBD processes in order to make the comparison of SL-based methods and the LPCA possible.

This paper has the following structure. In Section 2 we first define the notation for the QBD processes that we will use throughout this paper. In Section 2 we summarize the results of [19] for the DES processes as they apply to QBD processes with a DES and the resulting *quasi birth and death down entrance state algorithm* (QDESA). In Section 2 the QDESA procedure is specialized depending on the structure of the transition rate Q, applicable to the models

https://doi.org/10.1017/jpr.2015.12 Published online by Cambridge University Press

under investigation in this paper. Then, in Section 3 the introduced procedures are clarified by applying them to two specific queueing examples. In Section 4 we review the LPCA. In Section 5 we compare the procedures in speed (computational complexity). In Section 6 we discuss the type of model for which each procedure can be applied. We conclude with some models that further illustrate these comparisons.

2. Preliminary results

2.1. Successive lumping in QBD processes

In what follows we consider an ergodic QBD process $X(t)$ with states in a finite or countable set $\mathcal{X}$. The states (after relabeling) will be written as tuples (m, i), where in the state description the first entry $m = 0, 1, \dots, M$ represents the 'level' of the state and the second entry $i = 0, 1, 2, \dots, \ell_m$ represents the 'stage' of the state (m, i). The integers ℓ_m and M are given constants and they represent respectively the number of stages $(\ell_m + 1)$ and the highest level (M); these scalars can be infinite. Let Q denote the transition generator matrix. The process $X(t)$ is referred to as a 'level QBD' process if the only transitions allowed are to a state that is within the same level or to a level one step above or below, i.e. Q has the form:

$$Q = \begin{bmatrix} W^0 & U^0 & 0 & \cdots & 0 & 0 \\ D^1 & W^1 & U^1 & \ddots & 0 & 0 \\ 0 & D^2 & W^2 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & W^{M-1} & U^{M-1} \\ 0 & 0 & 0 & \cdots & D^M & W^M \end{bmatrix}.$$

The matrices W, D, and U represent 'within a level', 'down one level', and 'up one level' transitions, respectively. The sub-matrices W^m above are of dimension $(\ell_m + 1) \times (\ell_m + 1)$, the sub-matrices D^m are of dimension $(\ell_m + 1) \times (\ell_{m-1} + 1)$, and the submatrices U^m are of dimension $(\ell_m + 1) \times (\ell_{m+1} + 1)$. Furthermore, we will use the notation $\mathcal{L}_n = \{(n, i),$ $i = 0, 1, \dots, \ell\}$ for the level sets $(n = 0, 1, \dots, M)$.

Let π denote the steady state distribution, i.e. the solution of $\pi Q = 0$ and $\pi 1 = 1$. We denote by π^n the sub-vector of π formed by the stationary probabilities of the states of level n i.e. $\pi^n = [\pi(n, 0), \dots, \pi(n, \ell)]$.

In the context of this paper we will assume that every matrix D^m has only one nonzero column (that for this section we will assume be the first column). The underlying QBD process is therefore successively lumpable (a DES process) with respect to the partition $\{\mathcal{L}_n\}_{n\geq 0}$ of the state space $\mathcal{X}$; see [16] for lumping and [19] for a proof that $X(t)$ is lumpable with respect to this partition. In addition we will assume that $\ell_m = \ell$ for all m (i.e. the level size is independent of the level) and note that this condition is not necessary for the DES procedure to be applicable, but is necessary for the lattice path counting procedure, which will be discussed in Section 4. Below we will repeat the important definitions from [19], specialized for a QBD process.

In a QBD process we define the matrix $\widetilde{U}^m$ of size $(\ell + 1) \times (\ell + 1)$ as

$$\widetilde{U}^m = U^m \psi_m^\top \delta_m,$$

where ψ_m is a rowvector of size $\ell + 1$ identically equal to 1 and δ_m is a vector of the same size identically equal to 0 with a 1 on its first entry. Furthermore, we define

$$B^m = W^m + \widetilde{U}^m. \tag{1}$$

https://doi.org/10.1017/jpr.2015.12 Published online by Cambridge University Press

For a QBD process, we will call a matrix set $\{\mathcal{R}_m\}_m$ that satisfies the following equation *a rate matrix set*:

$$\pi^m = \pi^{m-1}\mathcal{R}_m \quad \text{for } m = 1, \dots, M_2. \tag{2}$$

In [19] it was shown that the matrix B^m is invertible. A simplification of [17, Theorem 2] for the special case of a QBD process implies that the matrix set $\mathcal{R}_0 := \{R_m\}_m$ defined by

$$R_m = -U^{m-1}(B^m)^{-1} \tag{3}$$

is a rate matrix set for Q when D^m has a single nonzero column.

Remark 1. Note that (2) and (3) imply that the following recursive relation holds for all $\nu = 0, \dots, m-1$:

$$\pi^m = \pi^\nu \prod_{k=\nu+1}^{m} R_k.$$

It is easy to see that the above defined π^m and R_m satisfy [20, Equation (12.2)]. The matrices R_m are solutions to [20, Equation (12.11)] but without the explicit procedure of (3) to compute them.

To obtain the steady state distribution $\pi = [\pi^0, \pi^1, \dots]$, we need only to compute π^0, which by [19, Theorem 3] is given by

$$\pi^0 = \delta_0[S_0^{M_2}\delta_0 - B^0]^{-1}, \tag{4}$$

where

$$S_0^{M_2} = \psi_0^\top + \sum_{m=1}^{M_2}\prod_{k=1}^{m} R_k \psi_m^\top. \tag{5}$$

The procedure to calculate the steady state distribution π when there is a DES in every level that is based on (3)–(5) is referred to as the QDESA.

2.2. Solution procedures for specific QBD processes

Unless otherwise stated, in the remainder of this paper we will consider homogenous level processes. Note that for these processes $B^m = B = W + \widetilde{U}$ (defined in (1)) for all m. Depending on the structure of the matrix B we define two subclasses, of decreasing generality, of the QDESA procedure. First, we identify homogenous QBD processes with a DES where the matrix B is of countable dimension and has the following form:

$$B = \begin{bmatrix} -b_0^d - b_0^u & b_0^u & 0 & 0 & 0 & \cdots \\ b_1^d + b_1^z & -b_1^w & b_1^u & 0 & 0 & \cdots \\ b_2^z & b_2^d & -b_2^w & b_2^u & 0 & \ddots \\ b_3^z & 0 & b_3^d & -b_3^w & b_3^u & \ddots \\ b_4^z & 0 & 0 & b_4^d & -b_4^w & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{6}$$

where $b_i^w = b_i^z + b_i^d + b_i^u$, and these elements b_i^a are nonzero for $a \in \{w, z, d, u\}$. The procedure to find the steady state distribution of these processes will be referred to as the QDESA$^+$.

https://doi.org/10.1017/jpr.2015.12 Published online by Cambridge University Press

Second, we consider homogenous QBD processes with a DES, where the matrix B has the structure of (6) and is *element homogenous*, i.e.

$$b_i^a = b^a \quad \text{for all } i = 0, 1, \dots \quad \text{and} \quad a \in \{z, d, w, u\}.$$

In this case the procedure to find the steady state distribution π will be called the QDESA^{++}.

In [18] we presented a fast $\mathcal{O}(\ell^2)$ algorithm to compute the inverse of matrix B of (6) when it is element homogenous and thus used in the QDESA^{++}. In that same paper we described a procedure with the same complexity to compute the inverse of B when it has the structure of (6) and it is not required to be element homogenous. An alternative method of computation with the same complexity is given in [13, p. 62], but only if $\ell < \infty$ and B is element homogenous.

Remark 2. One can determine which solution method is applicable by inspection of the matrix Q. If W^m has a birth and death structure, the QDESA$^+$ is applicable, and when both W and $\widetilde{U}$ have a homogenous birth and death structure, the QDESA^{++} is applicable.

When W has another structure than the ones described above, it might still have a sparse form. In that case it might be beneficial to use other fast matrix inversion algorithms, such as in [12] and [33].

In the remainder of this paper references to the QDESA include the special cases QDESA$^+$ and QDESA^{++} and it is assumed that the most efficient form of the QDESA is always applied.

3. Applications: classic queueing models

In this section we will discuss two classical queueing models and analyze how the procedures above can be used to compute the steady state distribution. The priority queue will be discussed in detail, and the longest queue more briefly. To avoid confusion we will use when necessary the notation A^{P} and A^{L} to distinguish a matrix A associated with the priority model of Section 3.1, or the longest queue of Section 3.2, respectively.

3.1. The priority queue

In the priority queue model customers arrive according to two independent Poisson processes with rate λ_i for queue i, $i = 1, 2$. There is a single server that serves at exponential rate μ, independently of the arrival processes. The server serves customers at queue 2 only when queue 1 is empty, preemptions are allowed, and server switches are instantaneous. Under these assumptions the state of the system can be summarized by a tuple (n, j), where n (respectively j) is the number of customers in queue 2 (respectively in queue 1).

It is easy to see that Q is the transition rate matrix of a DES process, in fact a homogenous level QBD process with $M = \infty$; the level sets $\mathcal{L}_n$ and their entrance states $(n, 0)$ are illustrated in Figure 1.

Since there is no maximum for the number of customers in queue 1 the sub-matrices D, W, and U have infinite dimension ($\ell = \infty$) and the representation below, where $d = (\lambda_1 + \lambda_2 + \mu)$.

https://doi.org/10.1017/jpr.2015.12 Published online by Cambridge University Press

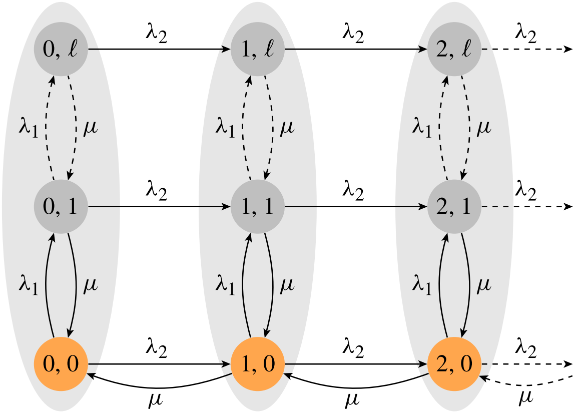

FIGURE 1: Transition diagram of the priority queue model.

Note that W_0 is obtained from W by replacing d in its $(0,0)$ position by $(\lambda_1+\lambda_2)$, since in state $(0,0)$ there are no customers in service. We have

$$D = \begin{bmatrix} \mu & 0 & \cdots \\ 0 & 0 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \qquad U = U^0 = \begin{bmatrix} \lambda_2 & 0 & 0 & \cdots \\ 0 & \lambda_2 & 0 & \ddots \\ 0 & 0 & \lambda_2 & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix},$$

$$W = \begin{bmatrix} -d & \lambda_1 & 0 & \cdots \\ \mu & -d & \lambda_1 & \ddots \\ 0 & \mu & -d & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Note that in this model, we have $U^0 = U = \lambda_2 I$; thus, $R^{\mathrm{P}} = R_1^{\mathrm{P}} := -\lambda_2 B^{-1}$, where

$$B^{\mathrm{P}} = \begin{bmatrix} -(\lambda_1+\mu) & \lambda_1 & 0 & 0 & \cdots \\ \lambda_2+\mu & -d & \lambda_1 & 0 & \cdots \\ \lambda_2 & \mu & -d & \lambda_1 & \ddots \\ \lambda_2 & 0 & \mu & -d & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

It is clear that matrix B^{P} has the required structure to use the QDESA^{++}. Thus, the priority queue model can be solved easily using this method.

3.2. Longest queue

In a longest queue model, see [35], two types of customer arrive according to independent Poisson streams, each with rate λ and form two queues according to their type. There is a single exponential server with rate $\mu > 2\lambda$ that severs customers from the longest queue (i.e. the one

https://doi.org/10.1017/jpr.2015.12 Published online by Cambridge University Press

having the most customers), where ties are resolved with equal probabilities for each queue; server queue switches are instantaneous.

To obtain meaningful results for this model, we will use the following state space description that is easy to work with. At each point of time let the state be specified by a tuple (n, j), where j denotes the difference between the two queue lengths and n denotes the length of the shortest queue. A more natural state space description is discussed in Section 6.2.

It is easy to deduce that this is a DES process, in fact a homogenous level QBD process, with $M = \infty$ with level sets $\mathcal{L}_n$ as described in Section 2 and entrance states $(n, 1)$ for level n, where matrices D, U, and W are as given below with $d = 2\lambda + \mu$. We note that W_0 is obtained from W by replacing d in its $(0, 0)$ position by $(\lambda_1 + \lambda_2)$, since in state $(0, 0)$ there are no customers in service. We have

$$D = \begin{bmatrix} 0 & \mu & 0 & \cdots \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \qquad U = \begin{bmatrix} 0 & 0 & 0 & \cdots \\ \lambda & 0 & 0 & \ddots \\ 0 & \lambda & 0 & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix},$$

$$W = \begin{bmatrix} -d & 2\lambda & 0 & \cdots \\ \mu & -d & \lambda & \ddots \\ 0 & \mu & -d & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Since $U^0 = U$, the rate matrices R_1 and R for this model are equal, i.e. $R_1^{\mathrm{L}} = R^{\mathrm{L}}$, as in the previous models, and the matrix B in this model has the following form:

$$B^{\mathrm{L}} = \begin{bmatrix} -d & 2\lambda & 0 & 0 & \cdots \\ \mu & -(\mu + \lambda) & \lambda & 0 & \cdots \\ 0 & \mu + \lambda & -d & \lambda & \ddots \\ 0 & \lambda & \mu & -d & \ddots \\ 0 & \lambda & 0 & \mu & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Note that the matrix B^{L} has a structure similar (but not identical) to that of B defined in (6); its structure from the second column is identical to that of B, but an extra column has been added to the left. This can be easily resolved with a suitable modification of the QDESA^{++}.

Remark 3. The feedback queue, the third model that is discussed in [30], fits the QDESA framework as well; its analysis is analogous to the analysis of the priority queue.

4. Lattice path counting

A different approach to compute the steady state distribution π for a class of Markov processes that includes the queueing models described before, is the LPCA of [29]; see also [30]. In this section we will repeat the LPCA in the notation used in this paper.

https://doi.org/10.1017/jpr.2015.12 Published online by Cambridge University Press

Throughout this paper we use a labeling of states that is consistent with our notation introduced in [16] and [19]. In [30] a similar tuple notation was used, but the meaning of the first and the second element is reversed. For example, in the priority queue model of Section 3.1 we denote a system with two queues with n customers in queue 2 and i in queue 1 as (n, i). This same (n, i) in [30] denoted a system with two queues with n customers in queue 1 and i customers in queue 2.

Recall that we used the *level* (first coordinate) sets $\mathcal{L}_n = \{(n, i),\ i = 1, \dots, \ell\}$, where $n = 0, 1, \dots$ to define a partition with respect to which the studied processes are 'level QBD' processes. A 'stage QBD' process can be defined analogously; one can rearrange the states of $\mathcal{X}$ in the order of stages (second coordinate), i.e. as

$$(0, 1), \dots, (M, 1), (0, 2), \dots, (M, 2), \dots, (0, \ell), \dots, (M, \ell).$$

In this case we define the stage sets to be $\mathcal{K}_i = \{(n, i),\ n = 0, 1, \dots\}$. Transitions are allowed one stage up and one stage down to preserve the QBD property in the direction of stages. Using a stage partition, we obtain the following representation of the transition generator matrix, which will be denoted by $\widehat{Q}$ to indicate that a stage partition is used:

$$\widehat{Q} = \begin{bmatrix} B_1 & B_0 & 0 & \cdots \\ A_2 & A_1 & A_0 & \ddots \\ 0 & A_2 & A_1 & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix},$$

where the dimension of the above sub-matrices is $M \times M$.

The matrix $\widehat{Q}$ in this paper is the same as the matrix Q of [30], subject to appropriate relabeling of states, as is mentioned above. Note that in [30] the notation M is used for our ℓ above and their corresponding ℓ is infinite.

Following the approach introduced in [30], a process $X(t)$ is called lattice path countable (LPC) if the following three conditions hold:

(i) when $j > 1$, the only transitions allowed from state (n, j) are to states $(n + e_1, j + e_2) \in \mathcal{X}$, where $e_1 \in \{0, 1\}$ and $e_2 \in \{-1, 0, 1\}$;

(ii) when $j > 1$, the transition rate $\widehat{Q}((n, j), (n + e_1, j + e_2))$ is a function of the jump size and direction only, i.e.

$$\widehat{Q}((n, j), (n + e_1, j + e_2)) = \hat{q}(e_1, e_2);$$

(iii) the process is a stage QBD process, where ℓ is infinite and M is finite or infinite.

In the previous section we described a rate matrix R that provides a relationship between the steady state distributions of the different levels. A similar recursion can be defined for the steady state vectors π_i for stages $i > 0$: $\pi_{i+1} = \pi_i \widehat{R}$, where $\widehat{R}$ is the minimal nonnegative solution to the matrix quadratic equation $A_0 + \widehat{R} A_1 + \widehat{R}^2 A_2 = 0$.

We have denoted the rate matrix constructed with LPC as $\widehat{R}$ to distinguish it from the matrix R used in (3).

In Figure 2 we illustrate a simplification of a transition diagram of a process that is a QBD process with respect both to the levels and to the stages. The LPCA can be applied with respect to the stages.

https://doi.org/10.1017/jpr.2015.12 Published online by Cambridge University Press

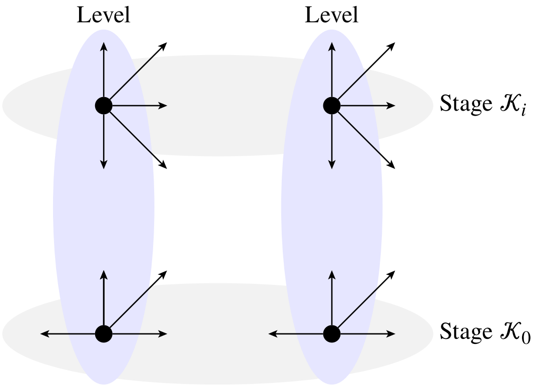

FIGURE 2: Levels and stages.

Furthermore, it is known (see, for example, [20]) that the elements $\hat{r}(n \mid m)$ of the matrix $\widehat{R} = [\hat{r}(n \mid m)]$ represent the expected taboo sojourn time in $(n, i+1)$ before the first return to stage i, given that the process starts in (m, i) multiplied by the sojourn time in stage i for any $i \geq 1$. Since the LPC assumption above does not allow transitions in the downward direction and has a homogenous structure by point (ii) above, the rate matrix is upper-triangular and has the following form:

$$\widehat{R} = \begin{bmatrix} \hat{r}_0 & \hat{r}_1 & \hat{r}_2 & \cdots \\ 0 & \hat{r}_0 & \hat{r}_1 & \cdots \\ 0 & 0 & \hat{r}_0 & \cdots \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix}.$$

In Theorem 1 below we provide an explicit expression for the elements of $\widehat{R}$. It is the main result of [30] and uses the following expressions:

$$\begin{aligned} P_h(s,u,m) &= \phi\langle 1,-1\rangle^s \phi\langle 1,0\rangle^t \phi\langle 1,1\rangle^u \phi\langle 0,1\rangle^{m-u} \phi\langle 0,-1\rangle^{m+1-s}, \\ L_h(s,u,m) &= \frac{1}{m+1}\binom{2m}{m}\binom{m+1}{s}\binom{m}{u}\binom{2m+t}{t}, \\ G_h &= \sum_{s=0}^{h}\sum_{u=0}^{h-s}\sum_{m=\max(u,s-1)}^{\infty} L_h(s,u,m)P_h(s,u,m), \qquad (7) \\ \kappa_h &= \frac{\phi\langle 1,0\rangle\kappa_{h-1} + \phi\langle 0,1\rangle \sum_{j=0}^{h-1} G_{h-j}\kappa_j + \phi\langle 1,1\rangle \sum_{j=0}^{h-1} G_{h-j-1}\kappa_j}{1-\phi\langle 0,1\rangle G_0}, \end{aligned}$$

where $\rho_0 = 1$ and $\rho_{-1} = 0$, and $\phi(e_1, e_2)$ denotes the transition probability from state (n, j) to state $(n+e_1, j+e_2)$.

Theorem 1. *The upper diagonal elements $\hat{r}_h$ of $\widehat{R}$ can be expressed as*

$$\hat{r}_h = 2\frac{\phi\langle 0,1\rangle\kappa_h + \phi\langle 1,1\rangle\kappa_{h-1}}{1+\sqrt{1-4\phi\langle 0,1\rangle\phi\langle 0,-1\rangle}}. \qquad (8)$$

The LPCA is using the calculation of (8), utilizing a new computation of G_h in (7) above based on hypergeometric functions; see [30, Equations (26) and (27)].

https://doi.org/10.1017/jpr.2015.12 Published online by Cambridge University Press

5. Comparative analysis

In this section we will compare the efficiency of the LPCA and the QDESA described in the previous section. To make a fair comparison between these algorithms we will compare their complexities in Section 5.1 for transition rate matrices on which they can *both* be applied. In Section 6 we discuss classes of model for which a version of the QDESA is applicable while the LPCA is not. We will also distinguish structures for which the LPCA can be used efficiently, but for which the QDESA is not readily applicable.

It is important to note that the LPCA is based on the existence of a 'homogeneous portion' of stages, i.e. transition rates are both stage and level independent, as is described in [30, Section 5] and summarized in the previous section. The nonhomogeneous part of the state space is considered to be (part of) stage $\mathcal{K}_0$. From the nonhomogeneous part we may deduce that the QDESA might not be applicable; the entrance state property might be violated. Exit states might still be present; for the formal definition of an exit state we refer the reader to [8]. In this paper we have described how an entrance state and an exit state are related and how the choice of levels can be adjusted to transform an exit state into an entrance state. However, no applications are known for which such a complex structure in $\mathcal{K}_0$ is necessary, and for which the QDESA is no longer applicable.

When a process has such a structure that the QDESA applies (with respect to the levels) *and* the LPCA applies (with respect to the stages), we note that B (where $R = UB^{-1}$) has to have the structure of (6), up to a permutation of the columns, due to the fact that the process is a QBD process in the stage direction; see Remark 2. Furthermore, it is easy to see that this homogeneous structured process implies that matrix B has an element-homogenous structure, since the elements are independent on the stages. Summarizing the above, we state the following proposition.

Proposition 1. *Suppose that the following hold:*

(i) *the LPCA is applicable to a QBD process with respect to the stages;*

(ii) *the set $\bigcup_{k=0}^{n} \mathcal{L}_k$ has an entrance state or the set $\bigcup_{k=n}^{M} \mathcal{L}_k$ has an exit state.*

Then the QDESA^{++} can be applied with respect to the level partition.

A result of this proposition is that for a fair computational comparison between the algorithms it suffices to compare the LPCA with the QDESA^{++}.

5.1. Computational complexity of the procedures

By (3) we know that the computational complexity of the QDESA^{++} is determined by the complexity of calculating the elements of the matrix R with dimension $\ell \times \ell$. Since U is a sparse matrix in this case, the computationally heavy step is to invert matrix B. For the LPCA the computational complexity is determined by the complexity of calculating the elements of matrix $\widehat{R}$. Recall that $\widehat{R}$ has dimension $M \times M$.

The general result on complexity is summarized in Theorem 2. To compare the complexities of the QDESA to that of the LPCA, we take $\ell = M$; for example, this is the case in the priority queue model when the queues have the same (finite or truncated) capacity. In the following complexity analysis we assume that arithmetic operations with individual elements have complexity $\mathcal{O}(1)$.

https://doi.org/10.1017/jpr.2015.12 Published online by Cambridge University Press

Theorem 2. *When the steady state distribution of a QBD process can be found both by using the LPCA and the QDESA, the following hold:*

(i) *using the LPCA, the computation of the stage-rate matrix $\widehat{R}$ has complexity $\mathcal{O}(M^4)$;*

(ii) *using the QDESA^{++}, the computation of the level-rate matrix R has complexity $\mathcal{O}(\ell^2)$.*

Proof of Theorem 2(i). We assign the complexity of $\mathcal{O}(h)$ to the computation of the term $\sum_{m=\max(u,s-1)}^{\infty} L_h(s,u,m)P_h(s,u,m)$ that involves hypergeometric functions; see [30, Equations (26) and (27)], noting that $s+u+t=h$. The *correct* complexity of the above computation is actually higher, but this lower bound is easy to establish when counting conservatively. From (7) we see that in order to calculate G_h we need approximately $(h^2/2)\mathcal{O}(h)=\mathcal{O}(h^3)$ iterations (a double summation). The computation of matrix $\widehat{R}$ (of size $M \times M$) requires the computation of all its M different nonzero elements $\hat{r}_0,\dots,\hat{r}_{M-1}$ and each of these computations is of complexity $\mathcal{O}(h^3)$. The complexity of the computation of rate matrix $\widehat{R}$ is $\sum_{h=0}^{M-1}\mathcal{O}(h^3)=\mathcal{O}(M^4)$. □

Proof of Theorem 2(ii). We will establish the complexity for the QDESA^{++}. The procedure for the computations of the elements of the first row and first column of C uses a single computation per element of $\mathcal{O}(1)$. For the remaining elements a linear expression has to be solved, having a complexity of $\mathcal{O}(1)$ per element as well. Thus, the total complexity of computing C is $\mathcal{O}(\ell^2)$, the number of elements of B^{-1}. The matrices U have a sparse form (at most three nonzero elements per row), induced by the fact that the LPCA is applicable by assumption. Since $R=UB^{-1}$, the complexity of computing R is $\mathcal{O}(\ell^2)$: both the complexity of the matrix multiplication UB^{-1} and of the calculation of B^{-1} have this complexity. □

Remark 4. For some special cases, for example the priority queue, the complexity of the LPCA is lower because of the absence of transitions from (n,j) to $(n+e_1, j+e_2)$ with $(e_1,e_2)\in\{\langle -1,1\rangle,\langle 1,1\rangle\}$ for all (n,j). In this special case the complexity of the LPCA is $\mathcal{O}(M^2)$ because in the computation of G_h, both $s=0$ and $u=0$ and the summation in (7) is only over m, i.e. the complexities of the LPCA and the QDESA are the same in this case.

Remark 5. When there is no additional structure on matrix B, both the QDESA^{+} and the QDESA^{++} cannot be used, so we need a general matrix inversion to compute B^{-1} of dimension ℓ by ℓ that is in complexity less than $\mathcal{O}(\ell^{2.379})$ (see [32]) when ℓ is finite. When U is a nonsparse matrix this provides a solution procedure with total complexity $\mathcal{O}(l^3)$ for the QDESA.

6. The applicability of the QDESA to more general models

In this section we will determine the differences in applicability between the QDESA and the LPCA, and present these differences with examples. We will consider variations of the queues in Sections 3.1 and 3.2 that can be solved with the QDESA but not with the LPCA.

One of the main advantages of the QDESA over the LPCA is that the QDESA not only provides a method to find the rate matrix, but the algorithm includes a way to find the steady state distribution using this rate matrix. Since the LPCA does not require any restrictions on the nonhomogenous part $\mathcal{K}_0$, the structure on this set can be very complex and a direct technique to perform this step is absent and not trivial to include. Therefore, the QDESA can be viewed as a more complete solution procedure. For that reason we will not discuss models that have a complicated structure on $\mathcal{K}_0$; even though it is possible to find the rate matrix for such a model with the LPCA, but perhaps not with the QDESA; within the LPCA no procedure is provided to find the steady state distribution.

https://doi.org/10.1017/jpr.2015.12 Published online by Cambridge University Press

There are four important classes of model for which (an extension of) the QDESA is applicable and for which the LPCA cannot be used at all. The first class involves element-nonhomogenous DES processes. In this case there is no homogeneous tail on which the LPCA is applicable. The second class involves processes with a finite number of stages ℓ, as described in Section 2; in the LPC case there is analysis only for the case in which the number of stages ℓ is infinite. The third class involves DES processes with ‘down’ transitions to the entrance state in a level L_{m-1} from more than one state in level L_m for some m. The fourth and most general class involves all DES processes, i.e. Markov chains with transitions from an arbitrary state (n, j) to states $(n+e_1, j+e_2) \in \mathcal{X}$, where $e_1 \in \{0, 1, \ldots\}$ and $e_2 \in \{\ldots, -1, 0, 1, \ldots\}$, under the condition of a single entrance state in the ‘down’ direction; see [19].

Conversely, there are processes for which the LPCA is applicable, but the QDESA is not. Such processes will contain transitions that destroy the DES property with respect to the level partition. For example, transitions from a state $(n, 1)$ to $(n-2, 1)$ are allowed in an LPC process, but are not allowed in a DES process, when $(n, 1)$ is the entrance state for every level $\mathcal{L}_n$. However, by relabeling and changing the levels we can construct a DES process in many cases.

In Table 1 we identify the difference in applicability between the two procedures. We note that the transitions within the heterogenous stage $\mathcal{K}_0$ are not restricted, i.e. matrices B_0 and B_1 are possibly nonsparse matrices in the LPCA procedure. We compare this with the restrictions that are imposed by the QDESA.

6.1. The priority queue with batch arrivals

Consider the priority queue model where two types of customer arrive in batches according to independent Poisson processes with rate λ_i for queue i, $i = 1, 2$. Upon arrival the size Z_i of a batch of type i becomes known. For each fixed i the Z_i are independent and identically distributed random variables that follow a known discrete distribution $\mathbb{P}(Z_i = z) = p_i(z)$.

There is a single server that serves at exponential rate μ, independent of the arrival processes. The server serves customers at queue 2 only when queue 1 is empty, preemptions are allowed and switches are instantaneous. Under these assumptions the state of the system can be summarized by a tuple (n, j), where n (respectively j) is the number of customers in queue 2 (respectively in

TABLE 1: Restrictions for the applicability of the LPCA and the QDESA.

Stage $\mathcal{K}_0$, the nonhomogeneous portion	
LPCA	QDESA
Within this stage all transitions allowed.	QSF structure should be obeyed.
Transitions leaving $\mathcal{K}_0$ allowed only to $\mathcal{K}_1$.	Transitions are allowed to all higher stages.
Elements nonhomogeneous.	Elements nonhomogeneous.
Solution procedure on $\mathcal{K}_0$ not included in algorithm.	Solution procedure included for all levels.
Stage $\mathcal{K}_i$ from the homogeneous portion	
LPCA	QDESA
Nearest neighbor structure within levels.	All transitions allowed within levels.
Nearest neighbor to ‘NE’, ‘E’, ‘SE’.	All transitions allowed to higher levels.
Elements *homogeneous*.	Elements nonhomogeneous.
No transitions to ‘NW’, ‘W’, ‘SW’ allowed.	Transitions to ‘W’ allowed to *entrance* state.
Number of stages must be infinite.	Number of stages can be finite or infinite.

queue 1). Because we assume that there is no maximum for the number of customers in queue 1 the sub-matrices of Q have infinite dimension. It is easy to see that Q is the transition rate matrix of a successively lumpable process with respect to the levels with $M_1 = 0$, $M_2 = \infty$, and the following within- and up-matrices, where $d = (\lambda_1 + \lambda_2 + \mu)$:

$$W = \begin{bmatrix} -d & \lambda_1 p_1(1) & \lambda_1 p_1(2) & \cdots \\ \mu & -d & \lambda_1 p_1(1) & \ddots \\ 0 & \mu & -d & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \qquad U^{nk} = \begin{bmatrix} \lambda_2 p_2(k) & 0 & 0 & \cdots \\ 0 & \lambda_2 p_2(k) & 0 & \ddots \\ 0 & 0 & \lambda_2 p_2(k) & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

The matrix W^0 has its $(1, 1)$ element equal to $-(\lambda_1 + \lambda_2)$ and all its other elements are the same as those of W. The matrix D is the same as that of the process described in Section 3.1. This model can be solved using the QDESA, but the LPCA is not applicable.

6.2. Longest queue model with nonhomogeneous arrival rates

We will extend the model discussed in Section 3.2 in such a way that now two types of customer arrive, according to independent Poisson streams, with rate λ_1 and λ_2. There is a single exponential server with rate $\mu > \lambda_1 + \lambda_2$. Note that the fact that the arrivals have a different rate implies that the state space description used in Section 3.2 does not induce a Markov chain. Therefore, we now let the state be specified by a tuple (n, j), where j denotes the number of customers in queue 1 and n the number of customers in queue 2. The buffers are of size M and ℓ, respectively, and can be either finite of infinite. The transition diagram is displayed in Figure 3 and the level partition is highlighted by the shaded background. It is easy

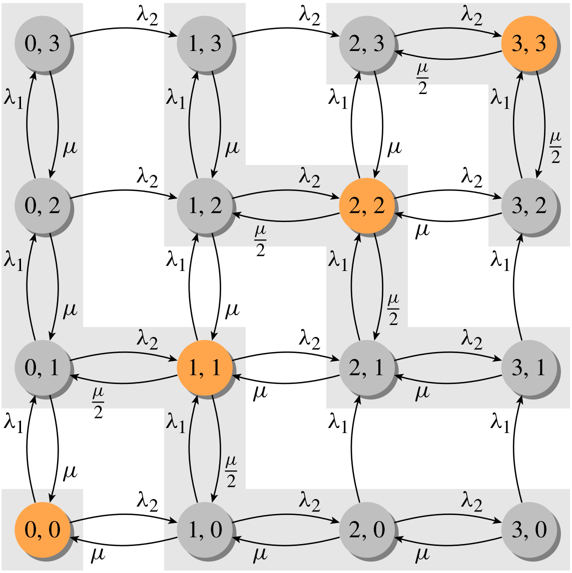

FIGURE 3: Longest queue model.

https://doi.org/10.1017/jpr.2015.12 Published online by Cambridge University Press

to deduce that this is a DES process, where the level sets $\mathcal{L}$ are formally described as

$$\mathcal{L}_m = \bigcup_{n=m}^{M} \{(n, m-1)\} \cup \bigcup_{i=m}^{\ell} \{(m-1, \ell)\} \cup \{(m, m)\}.$$

State (m, m) is the entrance states for the set $\bigcup_{k=0}^{m} \mathcal{L}_k$. With these different arrival rates, the LPCA cannot be used, while the QDESA$^+$ can be used. Note that the rate matrix R_m depends on the level m.

Acknowledgement

This research was partially supported by the National Science Foundation (grant no. CMMI-14-50743).

References

[1] Adan, I., Economou, A. and Kapodistria, S. (2009). Synchronized reneging in queueing systems with vacations. *Queueing Systems* **62,** 1–33.

[2] Adan, I. J. B. F., Kapodistria, S. and van Leeuwaarden, J. S. H. (2013). Erlang arrivals joining the shorter queue. *Queueing Systems* **74,** 273–302.

[3] Adan, I. J. B. F., Boxma, O. J., Kapodistria, S. and Kulkarni, V. G. (2013). The shorter queue polling model. *Ann. Operat. Res.* 1–34. Published online. DOI:10.1007/s10479-013-1495-0.

[4] Bini, D. A., Meini, B., Steffé, S. and Van Houdt, B. (2006). Structured Markov chains solver: software tools. In *SMCTOOLS* (Pisa, Italy, October 2006), ACM, New York, 10pp.

[5] Böhm, W., Krinik, A. and Mohanty, S. G. (1997). The combinatorics of birth–death processes and applications to queues. *Queueing Systems Theory Appl.* **26,** 255–267.

[6] Bright, L. and Taylor, P. G. (1995). Calculating the equilibrium distribution in level dependent quasi-birth-and-death processes. *Commun. Statist. Stoch. Models* **11,** 497–525.

[7] Eisenblätter, A. *et al.* (2003). Modelling feasible network configurations for UMTS. In *Telecommunications Network Design and Management*. Springer, New York, pp. 1–23.

[8] Ertiningsih, D., Katehakis, M., Smit, L. and Spieksma, F. (2016). QSF processes with level product form stationary distributions. To appear in *Naval Res. Logistics.*

[9] Etessami, K., Wojtczak, D. and Yannakakis, M. (2010). Quasi-birth–death processes, tree-like QBDs, probabilistic 1-counter automata, and pushdown systems. *Performance Evaluation* **67,** 837–857.

[10] Flajolet, P. and Guillemin, F. (2000). The formal theory of birth-and-death processes, lattice path combinatorics and continued fractions. *Adv. Appl. Prob.* **32,** 750–778.

[11] Gillent, F. and Latouche, G. (1983). Semi-explicit solutions for M/PH/1-like queuing systems. *Europ. J. Operat. Res.* **13,** 151–160.

[12] Hager, W. W. (1989). Updating the inverse of a matrix. *SIAM Rev.* **31,** 221–239.

[13] Heinig, G. and Rost, K. (1984). *Algebraic Methods for Toeplitz-Like Matrices and Operators*. Birkhäuser, Basel.

[14] Katehakis, M. N. and Derman, C. (1989). On the maintenance of systems composed of highly reliable components. *Manag. Sci.* **35,** 551–560.

[15] Katehakis, M. N. and Melolidakis, C. (1988). Dynamic repair allocation for a k-out-of-n system maintained by distinguishable repairmen. *Prob. Eng. Inf. Sci.* **2,** 51–62.

[16] Katehakis, M. N. and Smit, L. C. (2012). A successive lumping procedure for a class of Markov chains. *Prob. Eng. Inf. Sci.* **26,** 483–508.

[17] Katehakis, M. N. and Smit, L. C. (2012). On computing optimal (Q, r) replenishment policies under quantity discounts. *Ann. Operat. Res.* **200,** 279–298.

[18] Katehakis, M., Smit, L. and Spieksma, F. (2016). A solution to a countable system of equations arising in stochastic processes. Submitted.

[19] Katehakis, M. N., Smit, L. C. and Spieksma, F. M. (2015). DES and RES processes and their explicit solutions. *Prob. Eng. Inf. Sci.* **29,** 191–217.

[20] Latouche, G. and Ramaswami, V. (1999). *Introduction to Matrix Analytic Methods in Stochastic Modeling*. SIAM, Philadelphia, PA.

[21] Liu, D. and Zhao, Y. Q (1997). Determination of explicit solutions for a general class of Markov processes. In *Matrix-Analytic Methods in Stochastic Models* (Lecture Notes Pure Appl. Math. **183**), Dekker, New York, pp. 343–358.

[22] Mohanty, S. G. (1979). *Lattice Path Counting and Applications*. Academic Press, New York.

[23] Mohanty, S. G. and Panny, W. (1990). A discrete-time analogue of the M/M/1 queue and the transient solution: a geometric approach. *Sankhyā A* **52,** 364–370.

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

[25] Perros, H. G. (1994). *Queueing Networks with Blocking*. Oxford University Press.

[26] Spitzer, F. (2001). *Principles of Random Walk* (Graduate Texts Math. **34**), 2nd edn. Springer, New York.

[27] Ulukus, M. Y., Güllü, R. and Örmeci, L. (2011). Admission and termination control of a two class loss system. *Stoch. Models* **27,** 2–25.

[28] Van Houdt, B. and van Leeuwaarden, J. S. H. (2011). Triangular M/G/1-type and tree-like quasi-birth–death Markov chains. *INFORMS J. Comput.* **23,** 165–171.

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

[30] Van Leeuwaarden, J. S. H., Squillante, M. S. and Winands, E. M. M. (2009). Quasi-birth-and-death processes, lattice path counting, and hypergeometric functions. *J. Appl. Prob.* **46,** 507–520.

[31] Vlasiou, M., Zhang, J. and Zwart, B. (2014). Insensitivity of proportional fairness in critically loaded bandwidth sharing networks. Preprint. Available at http://arxiv.org/abs/1411.4841.

[32] Williams, V. V. (2012). Multiplying matrices faster than Coppersmith–Winograd. In *STOC '12—Proceedings of the 2012 ACM Symposium on Theory of Computing*, ACM, New York, pp. 887–898.

[33] Woodbury, M. A. (1950). Inverting modified matrices. *Memo. Rep.* **42,** Statistical Research Group, Princeton University.

[34] Zhao, Y. Q. and Grassmann, W. K. (1995). Queueing analysis of a jockeying model. *Operat. Res.* **43,** 520–529.

[35] Zheng, Y.-S. and Zipkin, P. (1990). A queueing model to analyze the value of centralized inventory information. *Operat. Res.* **38,** 296–307.

https://doi.org/10.1017/jpr.2015.12 Published online by Cambridge University Press