Steady-state analysis of large scale systems : the successive lumping method

Smit, L.C.

Steady State Analysis of Large-Scale Systems

The Successive Lumping Method

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof. mr. C. J. J. M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op woensdag 25 mei 2016
klokke 15.00 uur

door

Laurens Christiaan Smit
geboren te Leiden
in 1986

Promotor:

Prof. dr. W. Th. F. den Hollander (Universiteit Leiden)

Copromotor:

Dr. F. M. Spieksma (Universiteit Leiden)

Promotiecommissie:

Prof. dr. A. W. van der Vaart (Universiteit Leiden, voorzitter)
Prof. dr. J. J. Meulman (Universiteit Leiden, secretaris)
Prof. dr. M. N. Katehakis (Rutgers University, Newark & New Brunswick, USA)
Prof. dr. R. J. Boucherie (Universiteit Twente)
Dr. S. Kapodistria (TU Eindhoven)

Contents

CHAPTER 1

Introduction

1.1 Motivation

In this thesis we will consider large systems (networks) with random aspects to their behaviour. As the state space of such systems become large, their behaviour gets very hard to analyse, both via mathematical theory and computational algorithms and via computer simulation. This is an instance of the so-called big state space challenge or the 'curse of dimensionality'. This phenomenon is known to confound one's ability to draw valid inference for problems that are important in many areas of science, e.g. computer science, economics, building systems and statistics.

This thesis is concerned with the analysis of a specific type of these networks. More precisely, in this thesis we are interested in networks that can be modelled as Markov processes and in computing the stationary distribution associated with it. When this stationary distribution is known, many quantities of interest can be derived. For example this distribution can be useful to compute the average waiting time in a queueing system, or the percentage of time that there is no inventory in stock in an inventory system.

Our interest in large-scale Markov processes originates from discussions an inventory model with random lead times as a follow up of the (Q, r)-replenishment model that we studied in [S1]. This model has a very specific structure that makes it possible to decompose the state space and to compute the stationary distribution on a sequence of nested subsets of the space instead of considering the entire state space. When expanding our scope, we noticed that many more models possess the same structure natural way. Traditionally, model size reduction for Markov processes has been achieved by some variant of a static (one step) lumping (or aggregation) of states, for example in [39]. Other ways to reduce the state space are in general either not exact or require some approximation method to estimate the stationary distribution of the model under consideration.
In this thesis we will propose a new method to explicitly compute the stationary distribution for a large class of Markov processes in a successive way. We specialize it to a specific type of Markov processes and derive many properties induced by this structure.

In this introduction we will give a short overview of the work done in the thesis. Since we will consider many models from queueing theory, we start by briefly introducing some basic properties and notation used in this field of research. Next, we will summarize the successive lumping method, one of the most important concepts introduced in this thesis. This overview is necessary in order to comprehensibly describe the main contributions of this thesis, in Section 1.4. Since most of the other chapters contain separate introductions that cover literature references, we will restrict the literature review in this introduction to some of the most closely related references that are important to mention, when describing the contributions and possible future research directions in Section 1.5.

1.2 Queueing theory

Many models that serve as examples in this thesis, have their origin in queueing theory. Therefore we will introduce some notation and concepts related to queueing in more detail. Queueing theory is the mathematical study of waiting lines. Queueing applications are usually of a stochastic nature and can often be modelled as large scale (infinite) Markov processes. Customers arrive according to some distribution and join a waiting line. One (or more) servers provide service to the customers, also according to a certain distribution. When modelled correctly, queue lengths and waiting time can be predicted.

Single queueing systems are usually denoted according to the notation introduced by Kendall and are of the form $A/S/n$. Herein, A describes the time between arrivals to the queue, S the distribution of the service time and n the number of servers in the system. The most basic example is the $M/M/1$ queue: in this model, customers arrive independently (Markovian or memoryless, hence the M) to a single server, and are then served according to an exponential distribution (the second M).

The most straightforward practical application that comes to mind is a supermarket model where customers choose a line and are served by a cashier. For other applications of queueing systems one can think of call centres and of models arising in computer science. In the latter case the customers are *jobs* that need to be processed by one or more servers.
Two large classes of classes of queueing systems are the $M/G/n$ and $G/M/n$ types (where the G stands for general). A lot of research has been done to analyse these type of processes and it remains an active area, since very general systems are considered. The queueing applications in this thesis mainly focus on special cases of these two types of queueing models, by restricting the general arrival distribution or service times in some way. For example, we will review models in which customers sequentially arrive after a (either fixed or non-fixed) number of exponentially distributed inter-arrival phases, or are served in a number of phases. These phase type distributions can be very useful as an approximation method for general inter-arrival and service time distributions and they are widely used. For example, a model where both the arrival process and the service time occurs in phases has been analysed in [24].

1.3 A brief overview of Successive Lumping

To clarify the relation of this thesis to other literature and to appreciate its contributions, we briefly discuss the successive lumping approach below. Many results presented in this thesis are in some way (closely) related to this procedure: both as applications and extensions.

To explain the concept we will introduce some notation and concepts. Consider a Markov process $X(t)$ on state space $\mathcal{X}$. We assume that the process is irreducible and ergodic. We have designed successive lumping as a straightforward algorithm to compute the stationary distribution of $X(t)$ in an efficient and exact way. To be able to perform successive lumping on $X(t)$ we first partition the state space $\mathcal{X}$ into N *levels*, named $L_1, \ldots, L_N$, where N is either finite or infinite. We refer to this partition by the letter $\mathcal{L}$.

The main idea is to compute the relative stationary distribution of the states belonging to the first level, and use this result to compute the relative stationary distribution of the states in the union of the second level and the first level, where the latter is 'lumped' to a single state. We repeat this process for levels $3, 4, \ldots, N$ until all states have been considered. By combining these results we can compute the stationary distribution of the entire system.

This procedure can not be applied for an arbitrary level partition $\mathcal{L}$ and heavily depends on the existence of so-called *entrance states*. A state x is an entrance state of a set S, if and only if all one-step transitions from any state $y \in \mathcal{X} \backslash S$ to set S are to state x. In words, the only trajectories to 'enter' a certain set of states pass through the entrance state of that set. It is intuitively clear that when a set S has an entrance state, its stationary distribution is then independent of the system dynamics in its complement $\mathcal{X} \backslash S$.

The applicability of the successive lumping procedure can be stated as follows. Consider the sets $S_n = \cup_{i=1}^{n} L_i$, with $n = 1, 2, \ldots, N$. If each set S_n has an entrance state, then the stationary distribution of $X(t)$ can be computed via successive lumping with respect to partition $\mathcal{L}$. The precise details, formulas with proofs and characteristics of this procedure are described in Chapter 2.

1.4 Contributions

The successive lumping procedure by itself is only a part of the contributions of this thesis. Many of the results in this work are related to it and they increase the numerical speed and applicability of the procedure in some way. Moreover, in this thesis we investigate many side paths, that provide insights useful outside the scope of successive lumping, as for example the inverse of a matrix in Chapter 5. In this section we discuss the topics of this thesis per subject, in Section 1.6 we provide a brief overview per chapter.

1.4.1 Computation

Most of the existing methods designed to compute the stationary distribution of a Markov process are either not exact or do not use a successive approach. The benefit of using a recursive/successive procedure is, that it can be straightforwardly extended to infinite dimensional Markov processes. The successive lumping approach can lead to big computational benefits and it appears to be one of the most efficient computational procedures available to-date to compute the stationary distribution for large classes of problems. In Chapter 2 we introduce the explanation of the successive lumping procedure. It is thoroughly described in this chapter, proven and clarified with examples. For ease of exposition, the procedure in that chapter is described in discrete time. However, we will show that the successive lumping procedure is also applicable to continuous time Markov processes and to semi-Markov processes. Besides describing the successive lumpable procedure, the possibility of having *multiple* successive lumpable structures within one process is considered.

A specific direction in the literature concerning the stationary analysis of Markov processes comprises the matrix analytic and matrix geometric related algorithms, introduced in [80] and continued in [70]. These algorithms consider Markov processes, the state space of which satisfies a Quasi Skip Free structure (QSF): for a certain level partition, transitions in one direction are restricted to one step, i.e. levels can not be skipped. In line with these well-known methods our aim is to compute a *rate matrix set*, used to express the stationary distribution of a level in terms of the stationary distributions of lower (or higher) levels. We will show in Chapter 3 how successive lumping can be used to compute a rate matrix set explicitly, something that is a hard problem in general. We derive solutions for two different types of these QSF processes that are successively lumpable.
As far as we know, hardly any algorithms are known that find an exact representation of the rate matrix set, and most indirect iterative solution procedures require the computation of the solution to a quadratic matrix equation. A subset of QSF processes are the well studied QBD processes, in which transitions are only allowed one level up and down from any state. We will show in Chapter 3 how our derived results simplify for QBD processes. In their book, Latouche and Ramaswami ([70, pp. 197]) identify a special transition structure of the transitions in a QBD process, that allows them to compute the rate matrix explicitly. The Markov processes satisfying this structure can be deformed to a Markov process containing entrance states, as we explain in Chapter 7: we show that we can introduce 'artificial' instantaneous states without, that induce this specific structure. Since the successive lumping approach to construct the rate matrix has no extra restrictions for infinite state spaces, all models that are analysed following the lines in [70] fit the framework of successive lumping.

1.4.2 Applicability

One of the major insights that can be drawn from this thesis is that although the successive lumping appears limited, it is actually a very strong tool. Every (Markov) model *always*

contains several partitions of the state space that satisfy the requirements necessary to use successive lumping. For example, a trivial partition in which one set consists of a single state and the remaining states are contained in another set, satisfies the entrance state property and the process is thus successively lumpable with respect to this partition. In general, there will be other partitions that satisfy these requirements, but they might be hard to identify. The challenge lies in finding a partition that provides a large computational benefit when applying the method. It is important to mention, that in this dissertation we will not discuss how to find the partition over which the Markov process is successively lumpable: we assume that this partition is given, or, like in many applications, the model will provide it in a natural way.

Chapter 3 further deals with several applications of successively lumpable QSF processes. Many of these applications are in queueing: for example, we can determine an exact solution (the probability distribution of the number of customers in the system) for a queueing system where customers are served according to an Erlang process (i.e. in a fixed number of phases). We will also provide solutions to notoriously hard problems in inventory management and reliability theory. The usage of successive lumping in these models allows us to calculate the stationary distribution and to establish various performance measures efficiently.

In some Markov processes we can identify a partition that is close to satisfying the entrance state requirement for each set. We have constructed some additional algorithms and steps to relax this successive lumping requirement. For example, we can use thinning (see Chapter 7) or make use of exit states instead of entrance states (Chapter 6), if they are present.

Many models provide a partition with entrance states when a state space description is chosen in a natural way. In some models this partition is less straightforward to find. For example, consider the routing model in Chapter 8. In this system, customers (jobs) arrive independently in a two server model, where the two servers serve at different rates. The jobs join the queue in which they expect to have minimum amount of expected delay. We show how to identify the levels, and we provide a procedure to identify the entrance states of the level sets, given that the ratio of the service rates is known. This model is a perfect example of a system that is in general hard to analyse, but for which a non-trivial partition of the state space containing entrance states exists. Using successive lumping, we derive some numerical results that give insight in the behaviour of this queueing model.

When comparing successive lumping to other procedures that intend to find the stationary distribution of a Markov process (we have done a comparison in detail in Chapter 4) some features stand out. Many algorithms do require that the rates are homogeneous, as for example a lattice path counting algorithm (cf. [107]), or the compensation approach. (cf [14]). One major contribution that successive lumping brings to the various existing methods is, that the rates can be heterogeneous and level independent. In addition, when considering other explicit algorithms more closely, it turns out that most of them, require entrance states as well. In many procedures this feature is hidden, since these algorithms do not use the same level partition, or do not even use a level partition at all.

When the rates have a homogeneous structure, the transition structure within levels can be such that there exists a product form relation between the stationary distribution of the levels. In specific cases the states within the levels may even obey such a product form solution as well. In Chapter 6 we go into more detail on how to find the factor associated with this product form. Also, we will study the relation between entrance states and 'exit states' (their straightforward counterpart), and how exit states in some cases can help in constructing the product form factor.

1.4.3 Numerics

Interestingly enough, although the successive lumping procedure does not require homogeneous rates, it performs very fast and efficiently. It only requires a single matrix inversion per step, and this matrix has the size of the number of states in the level. In many models this matrix is of a very specific type (a birth-and-death structure) as we will show in Chapter 5. In that chapter we will derive a very efficient method to compute the inverse. Many algorithms exist to compute the inverse of a matrix, however the matrix that we consider is so specific, that we were able to find a faster method, that even readily extends to infinite sized matrices. Also, we take a closer look at the eigenvalues of the system; we derive a procedure that can find these eigenvalues from another matrix that is much easier to analyse. This matrix-structure also appears when considering some linear systems of equations, and is not restricted to the stationary analysis of a Markov process.

1.5 Future directions

There are many open directions to investigate as a follow up for the work presented in this thesis. First of all, it would be very interesting to see if there is a certain procedure or heuristic approach that could help to identify the partition to which a certain Markov process is successively lumpable. As has been argued before, the model under consideration can provide a first step to identify this structure. Many models do possess entrance states corresponding to levels of substantial size, but it is hard to locate them, especially when the states are completely unsorted. Without knowledge of the model, it is in general hard to construct an algorithm that identifies such a partition, in particular to find one that leads to substantial computational benefits. For these benefits, both the number of levels and the size of the levels need to be relatively big. One could think of procedures that identify entrance states of sets, by systematically looking at certain combinations of states. This is time consuming, but could be a feasible approach when the sets are not very large. To include the concept of successively constructing large sets with entrance states is undoubtedly the hardest part, and an inspiring research direction.

In Chapter 7 we have provided a variety of extension possibilities for the successive lumping procedure. Mostly, these extensions contain the relaxation of the entrance state requirement.

We suspect that it is very hard in general to find another way to work around this condition; as soon as a process does not return to a (artificial) state with probability one, it will become necessary to include the dynamics of the system outside the set under consideration. In some models, the 'thinning' method introduced in that chapter, (distributing transitions from a specific state in separate processes) could provide to be a powerful tool. This method would be even more powerful if an extension would be developed to perform this technique in different states simultaneously instead of sequentially. It would be interesting to consider some models that contain single transitions that violate the successive lumping property with respect to a given partition. These models could have their roots in any application area.

Another very interesting direction in which the successive lumping could be useful is to apply it in the theory of Markov decision processes. In these processes, we identify two variants of successive lumpability. The first variant is when the process is successively lumpable with respect to the same partition for *every* possible policy. The other variant is when the process is successively lumpable with respect to different partitions for different strategies. We can construct a successive lumping based algorithm to compute the discounted reward efficiently per policy. In the first variant, one could think of some sort of combination of these type of algorithms, that find the average reward per set and find the optimal policy per set. The exact details need to be figured out, but this approach appears to be a fruitful way to at least find a well performing heuristic for the optimal policy of the entire decision process.
In the second variant the identified partition does not contain fixed entrance states per level for all strategies. Therefore computing the optimal policy will benefit less from the identified structure. Although it is still possible to compute the discounted reward per policy efficiently, there are no unique defined partition and corresponding levels anymore. When there exists levels that are present in all partitions, we might be able to find the optimal strategy for the states belonging to this set. The details for applying successive lumping in Markov decision processes still need to be figured out, and could be a possible follow-up direction to the work done in this thesis.

Some models in the literature contain a partition that justifies the use of the successive lumping approach. We would like to perform a more detailed numerical analysis on some of these models, such as we did for the shortest expected delay routing model considered in Chapter 8. Before, the stationary analysis of these models might not have been straightforward to do. As a first step, it would be necessary to identify these models in applications, as in power grid networks, queueing and inventory models and in reliability. We are very confident that many applications can be found in the mentioned areas.

1.6 Overview of the thesis

To conclude this introduction, we summarise the content of the thesis by providing how it is organized. In Chapter 2, we introduce successive lumpable Markov processes. Also in this chapter we give some applications and introduce multiple successive lumpability. This

chapter appeared as [S2].
In Chapter 3, we apply the method of the previous chapter to QSF processes, and use it to construct the rate matrices, thus fitting the framework of the matrix geometric approach. We also provide bounds for the stationary distribution. The content of this chapter appeared as [S3].
In Chapter 4 we compare the successive lumping method with lattice path counting algorithms, with respect to both speed and applicability. This chapter will appear as [S4].
Chapter 5 handles the inverse of a specific matrix that arises in a natural way, when the successive lumping method is applied. It provides algorithms for several cases to compute this inverse and also studies the eigenvalues of this matrix. This Chapter has been submitted as [S5].
In Chapter 6, we consider a specific type of QSF processes, in which the stationary distribution has a product form for the levels. We apply this to queues where the customers arrive according to a Coxian inter-arrival distribution. This chapter has been submitted as [S6].
In Chapter 7, we will discuss some possible extensions to lumping. We will relax the entrance state requirement and show how considering the inverse of the rate matrix can lead to a computational benefit. A part of this chapter will be submitted as [S7].
In Chapter 8 we consider a specific model that has its roots in computer science. This model, where jobs are routed to the server where they will receive the least delay before service, has a hidden successive lumping structure. We trace this structure, and perform a numerical analysis for the stationary distribution.The work in this chapter will be submitted as [S8].

A list of self-references is given at the beginning of the Bibliography. For ease of exposition, the notation might slightly differ per chapter.

CHAPTER 2

Successive lumping

This chapter appeared as: *A Successive Lumping Procedure for a Class of Markov Chains*, cf. [S2].

2.1 Introduction to Chapter 2

In this chapter we identify a class of (discrete time) Markov chains that we call successively lumpable, for which it is shown that the stationary probabilities can be obtained by successively computing the stationary probabilities of a propitiously constructed sequence of Markov chains. Each of the latter chains has a, typically much, smaller state space and a successive method of solution becomes possible with significant computational improvements. Lumping of states was first discussed in [63]. Methods and benefits of aggregation/disaggregation are thoroughly described in [92], [77] and [116]. In our construction the new key idea is to identify conditions (cf., Definition 2.2) on the transition matrix of the Markov chain under which it is *successively lumpable*. A necessary condition for a chain to be successively lumpable is the existence of "entrance states" cf., Definition 2.2. These states are called "input states" by [39] and they are a special case of the "mandatory states" which have been studied in [66] and [67].

This chapter is organized as follows. In Section 2.2.1 after some preliminaries we provide the basic framework for the first lumping stage. The successively lumpable class of Markov chains is defined in Section 2.2.2 and their main properties are given in Theorems 2.2 and 2.3. These theorems are the main results of this chapter. In Section 2.3 another class of Markov chains is introduced for which, using our results of Section 2.2, we construct a multiple successive lumping procedure. In Section 2.4, we discuss the ramifications of the work in Sections 2.2 and 2.3 to the case of semi-Markov processes and continuous time Markov processes.

Later in this dissertation, we will study applications of successively lumpable Markov chains to certain classical reliability/queueing problems. Versions of these reliability/queueing mod-

els have been studied before in [36], [43], [51], [61] as well as in [60], [87], [117] and references therein.

2.2 Successively lumpable Markov chains

Let $X(t)$ denote an irreducible and positive recurrent Markov chain on a finite or countable state space $\mathcal{X}$. Clearly $\mathcal{X}$ can be partitioned into a (possibly infinite) sequence of mutually *exclusive* and *exhaustive* sets $\mathcal{D} := \{D_0, D_1, \ldots, D_M\}$, with $M \leq \infty$, $\cup_{m=0}^{M} D_m = \mathcal{X}$, and $D_m \cap D_{m'} = \emptyset$, when $m \neq m'$. For notational convenience, the elements of each set D_m will be denoted (relabelled) as $\{(m,1),(m,2),\ldots,(m,\ell_m)\}$, for some fixed constants $\ell_m \leq \infty$. The transition matrix of $X(t)$ will be denoted by $\underline{\underline{\mathrm{P}}} = [p(m,j \,|\, m',j')]$, where its $((m',j'),(m,j))-$element is

$$p(m,j \,|\, m',j') = \mathbf{Pr}[X(t+1) = (m,j) \,|\, X(t) = (m',j')].$$

In the sequel we will denote the stationary probabilities for state (m,j) by $\pi(m,j) = \lim_{t\to\infty} \mathbf{Pr}[X(t) = (m,j)]$. These probabilities exist, because the Markov chain $X(t)$ is irreducible and positive recurrent. We will use the notation

$$\underline{\pi} = \big(\pi(0,1),\ldots,\pi(0,\ell_0),\pi(1,1),\ldots,\pi(1,\ell_1),\ldots,\pi(M,1),\ldots,\pi(M,\ell_M)\big).$$

In the sequel, to avoid trivial cases we assume that $M \geq 2$, i.e., the partition $\mathcal{D}$ has at least two subsets. Note also, that we will use the symbol $\underline{\underline{\mathrm{A}}}$ to denote a matrix where $a(i,j)$ will denote its (i,j)-th element and $\underline{a}(i)$ (respectively $\underline{a}'(j)$) will denote its i-th row (respectively j-th column) vector.

2.2.1 Definitions and proofs for the first lumping stage

We start with the definition of the **entrance state** of a subset D_m of a partition $\mathcal{D}$ of the state space $\mathcal{X}$.

Definition 2.1. A subset D_m of $\mathcal{D}$ has an **entrance state** $(m, \boldsymbol{\varepsilon}_m(\mathcal{D})) \in D_m$ if and only if

$$p(m,j \,|\, m',j') = 0, \text{ for all } m' \neq m \text{ with } j \neq \boldsymbol{\varepsilon}_m(\mathcal{D}), \text{ and all } j' \in D_{m'}.$$

Remark 2.1. i) Note that from the positive recurrence assumption it follows that if $D_m \in \mathcal{D}$ has an entrance state, there exists some $(m',j') \in D_{m'}$ with $m' \neq m$ such that

$$p(m, \boldsymbol{\varepsilon}_m(\mathcal{D}) \,|\, m',j') > 0.$$

ii) An entrance state of a set D_m is the *only* state via which the set D_m can be entered by the chain $X(t)$ from a state in $\mathcal{X}\backslash D_m$, where given two sets A and B, $A\backslash B$ denotes the elements of A that do not belong to B.

iii) Note also that in the familiar one dimensional notation for the states, a subset D of $\mathcal{X}$ has an **entrance state** $\boldsymbol{\varepsilon} \in D$ if

$$p(j|j') = 0 \text{ for all } j \neq \boldsymbol{\varepsilon},\, j \in D \text{ and all } j' \notin D.$$

Given a partition $\mathcal{D}$ with an entrance state $(0, \boldsymbol{\varepsilon}_0(\mathcal{D})) \in D_0$ we construct the following Markov chains.

a) A Markov chain $Z_0(t)$ on state space D_0 with transition matrix $\underline{\underline{\mathrm{U}}}_{D_0}$ which elements are as follows:

$$u_{D_0}(0, j \,|\, 0, i) = \begin{cases} p(0, \boldsymbol{\varepsilon}_0(\mathcal{D})) \,|\, 0, i) + \displaystyle\sum_{(k,j') \notin D_0} p(k, j' \,|\, 0, i), & \text{if } j = \boldsymbol{\varepsilon}_0(\mathcal{D}), \\ p(0, j \,|\, 0, i), & \text{otherwise.} \end{cases} \tag{2.1}$$

b) A Markov chain $X_1(t)$ with state space $\mathcal{X}_1 = \{(1,0)\} \cup D_1 \cup \ldots \cup D_M$ and transition matrix $\underline{\underline{\mathrm{P}}}_1$ where its $((k,j),(k',j'))$-th element is defined by Eq. (2.2) below if $(k,j) = (k',j') = (1,0)$ and by Eq. (2.3), otherwise.

$$p_1(1,0 \,|\, 1,0) = \sum_{(0,i'),\,(0,i) \in D_0} p(0, i' \,|\, 0, i) \upsilon_{D_0}(0,i), \tag{2.2}$$

$$p_1(k', j' \,|\, k, j) = \begin{cases} \displaystyle\sum_{(0,i) \in D_0} p(k', j' \,|\, 0, i) \upsilon_{D_0}(0,i), & \text{if } (k,j) = (1,0), \\ \displaystyle\sum_{(0,i) \in D_0} p(0, i \,|\, k, j), & \text{if } (k',j') = (1,0), \\ p(k', j' \,|\, k, j), & \text{otherwise.} \end{cases} \tag{2.3}$$

It is easy to see that both chains $Z_0(t)$ and $X_1(t)$ are irreducible and positive recurrent, because $X(t)$ has these properties as well. The steady state probabilities of Markov chain $Z_0(t)$ will be denoted by $\upsilon_{D_0}(0,i) = \lim_{t\to\infty} \mathbf{Pr}[Z_0(t) = (0,i)]$. The vector of the steady state probabilities of the Markov chain $X_1(t)$, will be denoted by:

$$\underline{\pi}_1 = \big(\pi_1(1,0); \pi_1(1,1), \ldots, \pi_1(1,\ell_1), \ldots, \pi_1(M,1), \ldots, \pi_1(M,\ell_M)\big).$$

Note that in the above construction of the new process $X_1(t)$, we have introduced an artificial state we denote as $(1,0)$. This state $(1,0)$ essentially represents the "lumped states" of the set D_0 of the initial process $X(t)$; we have used a semicolon in the above notation for $\underline{\pi}_1$ to emphasize this fact.

We will use the notation $\underline{\underline{\mathrm{U}}}_{D_0} = [\underline{u'}_{D_0}(0,1), \ldots, \underline{u'}_{D_0}(0,\ell_0)]$, where $\underline{u'}_{D_0}(0,j)$ denotes the j-th column of the transition matrix $\underline{\underline{\mathrm{U}}}_{D_0}$. Similarly,

$$\underline{\underline{\mathrm{P}}} = [\underline{p'}(0,1), \ldots, \underline{p'}(0,\ell_0), \ldots, \underline{p'}(M,1), \ \ldots, \underline{p'}(M,\ell_M)].$$

It is well known that $\underline{\pi}$ is the solution to the following system of equations: $\underline{\pi}\ \underline{\underline{\mathrm{P}}} = \underline{\pi}$ and $\underline{\pi}\ \underline{1}' = 1$. Here $\underline{1}$ will always denote a vector of ones of the same dimension as $\underline{\pi}$.

We will next state and prove the following proposition and theorem.

Proposition 2.1. *If D_0 has an entrance state $(0, \varepsilon_0(\mathcal{D}))$, then the following is true for all $(0,i) \in D_0$:*

$$\upsilon_{D_0}(0,i) = \frac{\pi(0,i)}{\sum_{(0,j)\in D_0} \pi(0,j)}. \tag{2.4}$$

Proof. Let $\underline{\upsilon}_{D_0} = \big(\upsilon_{D_0}(0,1), \ldots, \upsilon_{D_0}(0,\ell_0)\big)$. It is clear that for $\underline{\upsilon}_{D_0}$, defined by Eq. (2.4), the statement $\underline{\upsilon}_{D_0}\ \underline{1}' = 1$ holds. To prove that this choice of $\underline{\upsilon}_{D_0}$ is the solution, we will show that it also satisfies:

$$\underline{\upsilon}_{D_0}\ \underline{\underline{\mathrm{U}}}_{D_0} = \underline{\upsilon}_{D_0}. \tag{2.5}$$

By uniqueness of solutions to Eq. (2.5) (with $\underline{\upsilon}_{D_0}\ \underline{1}' = 1$) it then follows that $\underline{\upsilon}_{D_0}$ is indeed the steady state vector. To show that Eq. (2.5) holds, we distinguish two cases: the entrance state $(0, \varepsilon_0(\mathcal{D}))$ or any of the other states. For this, we will use the following straightforward derivation:

$$\begin{aligned}
\sum_{(0,j)\in D_0} p(0,\varepsilon_0(\mathcal{D}) \mid 0,j)\pi(0,j) =& \pi(0,\varepsilon_0(\mathcal{D})) - \sum_{(k,i')\notin D_0} p(0,\varepsilon_0(\mathcal{D}) \mid k,i')\pi(k,i') \\
=& \pi(0,\varepsilon_0(\mathcal{D})) - \sum_{(k,i')\notin D_0} \big(1 - \sum_{(k',i'')\notin D_0} p(k',i'' \mid k,i')\big)\pi(k,i') \\
=& \pi(0,\varepsilon_0(\mathcal{D})) - \sum_{(k,i')\notin D_0} \pi(k,i') \\
&+ \sum_{(k',i''),(k,i')\notin D_0} p(k',i'' \mid k,i')\pi(k,i') \\
=& \pi(0,\varepsilon_0(\mathcal{D})) - \sum_{(k,i')\notin D_0} \pi(k,i') \\
&+ \sum_{(k',i'')\notin D_0} \big(\pi(k',i'') - \sum_{(0,j)\in D_0} p(k',i'' \mid 0,j)\pi(0,j)\big) \\
=& \pi(0,\varepsilon_0(\mathcal{D})) - \sum_{(0,j)\in D_0} \sum_{(k,i')\notin D_0} p(k,i' \mid 0,j)\pi(0,j).
\end{aligned}$$

Now, using this equality we obtain for $(0,i) = (0,\varepsilon_0(\mathcal{D}))$:

$$\begin{aligned}
\underline{\upsilon}_{D_0}\,\underline{u'}_{D_0}(0,\varepsilon_0(\mathcal{D})) &= \sum_{(0,j)\in D_0} \upsilon_{D_0}(0,j) u_{D_0}(0,\varepsilon_0(\mathcal{D})\,|\,0,j) \\
&= \frac{\sum_{(0,j)\in D_0}\pi(0,j)\left(p(0,\varepsilon_0(\mathcal{D})\,|\,0,j) + \sum_{(k,i')\notin D_0} p(k,i'\,|\,0,i)\right)}{\sum_{(0,i')\in D_0}\pi(0,i')} \\
&= \frac{\pi(0,\varepsilon_0(\mathcal{D})) - \sum_{(0,j)\in D_0}\pi(0,j)\sum_{(k,i')\notin D_0}\left(p(k,i'\,|\,0,j) - p(k,i'\,|\,0,j)\right)}{\sum_{(0,i')\in D_0}\pi(0,i')} \\
&= \frac{\pi(0,\varepsilon_0(\mathcal{D}))}{\sum_{(0,j)\in D_0}\pi(0,j)} = \upsilon_{D_0}(\epsilon_0(\mathcal{D})).
\end{aligned}$$

Similarly, for $(0,i) \neq (0,\varepsilon_0(\mathcal{D}))$:

$$\begin{aligned}
\underline{\upsilon}_{D_0}\,\underline{u'}_{D_0}(0,i) &= \sum_{(0,j)\in D_0} \upsilon_{D_0}(0,j) u_{D_0}(0,i\,|\,0,j) \\
&= \frac{1}{\sum_{(0,i')\in D_0}\pi(0,i')} \sum_{(0,j)\in D_0} \pi(0,j) p(0,i\,|\,0,j) \\
&= \frac{\pi(0,i)}{\sum_{(0,i')\in D_0}\pi(0,i')} \\
&= \upsilon_{D_0}(0,i).
\end{aligned}$$

Thus, $\underline{\upsilon}_{D_0}\,\underline{u'}_{D_0}(0,i) = \upsilon_{D_0}(0,i)$ for all $(0,i) \in D_0$ and the proof is complete. □

For the chain $X_1(t)$ we have the following main result concerning the steady state distribution.

Theorem 2.1. *If D_0 has an entrance state $(0,\varepsilon_0(\mathcal{D}))$, then the following are true regarding the Markov chains $X(t)$ and $X_1(t)$.*

i) If $(k,j) \neq (1,0)$, then

$$\pi_1(k,j) = \pi(k,j),$$

ii) If $(k,j) = (1,0)$, then

$$\pi_1(k,j) = \sum_{(0,i)\in D_0} \pi(0,i).$$

Proof. We need to show that the above choice of $\underline{\pi}_1$ satisfies the steady state equations of the $X_1(t)$ process, i.e., it is the unique solution of the linear system

$$\underline{\pi}_1\,\underline{\underline{\mathrm{P}}}_1 = \underline{\pi}_1,$$

together with $\underline{\pi}_1\ \underline{1}' = 1$. The latter equality is easy to see. Next, for each state $(k,j) \neq (1,0)$ we have:

$$\begin{aligned}
\underline{\pi}_1 \underline{p}_1'(k,j) &= \sum_{(k',j')\in\mathcal{X}_1} p_1(k,j \,|\, k',j')\pi_1(k',j') \\
&= \sum_{(k',j')\in\mathcal{X}_1\setminus\{(1,0)\}} p(k,j \,|\, k',j')\pi(k',j') \\
&\quad + \sum_{(0,i)\in D_0} p(k,j \,|\, 0,i)\upsilon_{D_0}(0,i) \sum_{(0,i')\in D_0} \pi(0,i') \\
&= \sum_{(k',j')\in\mathcal{X}_1\setminus\{(1,0)\}} \pi(k',j')p(k,j \,|\, k',j') + \sum_{(0,i)\in D_0} p(k,j \,|\, 0,i)\pi(0,i) \\
&= \pi(k,j) = \pi_1(k,j),
\end{aligned}$$

and thus $\pi_1(k,j) = \pi(k,j)$ satisfies the steady state equations of the $X_1(t)$ process for all $(k,j) \neq (1,0)$.

Finally, for $(k,j) = (1,0)$, we have:

$$\begin{aligned}
\underline{\pi}_1\, \underline{p}_1'(1,0) &= \sum_{(k',j')\in\mathcal{X}_1} p_1(1,0 \,|\, k',j')\pi_1(k',j') \\
&= \sum_{(k',j')\in\mathcal{X}_1\setminus\{(1,0)\}} \sum_{(0,i)\in D_0} p(0,i \,|\, k',j')\pi(k',j') \\
&\quad + \sum_{(0,i),(0,i')\in D_0} p(0,i \,|\, 0,i')\upsilon_{D_0}(0,i') \sum_{(0,i'')\in D_0} \pi(0,i'') \\
&= \sum_{(0,i)\in D_0} \sum_{(k',j')\in\mathcal{X}_1\setminus\{(1,0)\}} p(0,i \,|\, k',j')\pi(k',j') + \sum_{(0,i),(0,i')\in D_0} p(0,i \,|\, 0,i')\pi(0,i') \\
&= \sum_{(0,i)\in D_0} \sum_{(k',j')\in\mathcal{X}} p(0,i \,|\, k',j')\pi(k',j') \\
&= \sum_{(0,i)\in D_0} \pi(0,i) = \pi_1(1,0).
\end{aligned}$$

So indeed the choice of $\pi_1(1,0) = \sum_{(0,i)\in D_0} \pi(0,i)$ satisfies the steady state equation of $X_1(t)$, that corresponds to state $(1,0)$.

The proof is now complete. □

In section 2.2.2 below we provide conditions under which it is possible to successively (or sequentially) use the lumping procedure of Section 2.2.1 over the sets $D_1,\ D_2, \ldots, D_m$. In section 2.2.3 we present the algorithm and a computational example.

2.2.2 Definitions and proofs for successive lumping

We start with the following extended notation and definitions. For a Markov chain $X(t)$, with state space $\mathcal{X}$, transition matrix $\underline{\underline{\mathrm{P}}}$ and a partition $\mathcal{D} = \{D_0, \ldots, D_M\}$, we define $\Delta_0 = D_0$, $\Delta_m = \{(m,0)\} \cup D_m$, with $m = 1, 2, \ldots M$, where $(m,0)$ is an artificial state, representing the lumped states: $\bigcup_{k=0}^{m-1} D_k$.

We further define the partitions $\mathcal{D}_m = \{\Delta_m, D_{m+1}, \ldots, D_M\}$ and the state spaces: $\mathcal{X}_m = \Delta_m \cup D_{m+1} \cup \ldots \cup D_M$, for $m = 0, \ldots, M$.

For notational consistency, we will use the notation: $X_0(t) = X(t)$, $\mathcal{X}_0 = \mathcal{X}$, $\mathcal{D}_0 = \mathcal{D}$, $\underline{\underline{\mathrm{P}}}_0 = \underline{\underline{\mathrm{P}}}$, and $\underline{\pi}_0 = \underline{\pi}$. Furthermore without loss of generality we will denote the entrance state of Δ_m in $\mathcal{D}_m$ with $(m, \epsilon_m(\mathcal{D}_m))$, if it exists.

We next state the following definition.

Definition 2.2. A Markov chain $X(t)$ is called **successively lumpable** with respect to partition $\mathcal{D} = \{D_0, \ldots, D_M\}$ if and only if the set $D_0 \cup D_1 \cup \ldots \cup D_i$ has an entrance state for all $i = 0, \ldots, M$.

The above definition means that there exists only one state in $\{D_0 \cup \ldots \cup D_{m'}\}$ that can be entered from a state in D_m when $m > m' > 0$. Note also that the definition implies that transitions out of states in $D_{m'}$ can only lead to states in D_m with $m \geq m' \geq 0$ or to the entrance state of the set $\Delta_{m'-1}$.

In the sequel, the state (m, η_m) will denote an arbitrary but fixed state in Δ_m.

Given the partition $\mathcal{D}_m$ we successively construct the following Markov chains.

a) A Markov chain $Z_m(t)$ with state space Δ_m and transition matrix $\underline{\underline{\mathrm{U}}}_{\Delta_m}$ with

$$u_{\Delta_m}(m, j \,|\, m, i) = \begin{cases} p_m(m, j \,|\, m, i) + \displaystyle\sum_{(k,i') \notin \Delta_m} p_m(k, i' \,|\, m, i), & \text{if } (m,j) = (m, \eta_m), \\ p_m(m, j \,|\, m, i), & \text{otherwise.} \end{cases} \tag{2.6}$$

Suppose that the chains $Z_m(t)$ is irreducible and positive recurrent. Then its steady state probabilities will be denoted by $\upsilon_{\Delta_m}(m,i)$, i.e.,

$$\upsilon_{\Delta_m}(m,i) = \lim_{t \to \infty} \mathbf{Pr}[Z_m(t) = (m,i)].$$

b) A Markov chain $X_{m+1}(t)$ with state space $\mathcal{X}_{m+1} = \Delta_{m+1} \cup D_{m+2} \cup \ldots \cup D_M$ and transition matrix $\underline{\underline{\mathrm{P}}}_{m+1}$, with elements $((k,j),(k',j'))$ defined by Eq. (2.7) below if $(k,j) = (k',j') = (m+1, 0)$ and by Eq. (2.8) otherwise.

$$p_{m+1}(m+1,0\,|\,m+1,0)=\sum_{(m,i'),\,(m,i)\in\Delta_m} p_m(m,i'\,|\,m,i)\upsilon_{\Delta_m}(m,i), \qquad (2.7)$$

$$p_{m+1}(k',j'\,|\,k,j)=\begin{cases}\displaystyle\sum_{(m,i)\in\Delta_m} p_m(k',j'\,|\,m,i)\upsilon_{\Delta_m}(m,i), & \text{if } (k,j)=(m+1,0),\\ \displaystyle\sum_{(m,i)\in\Delta_m} p_m(m,i\,|\,k,j), & \text{if } (k',j')=(m+1,0),\\ p_m(k',j'\,|\,k,j), & \text{otherwise.}\end{cases} \qquad (2.8)$$

Note that in order to compute $p_{m+1}(\cdot\,|\,\cdot)$ we first need to compute $\upsilon_{\Delta_m}(\cdot)$. The vector of the steady state probabilities of the chain X_{m+1} will be denoted by:

$$\underline{\pi}_{m+1}=\big(\pi_{m+1}(m,0);\pi_{m+1}(m,1),...,\pi_{m+1}(m,\ell_m),...,\pi_{m+1}(M,1),...,\pi_{m+1}(M,\ell_M)\big).$$

We will use the notation:

$$\underline{\underline{\mathrm{U}}}_{\Delta_m}=[\underline{u}'_{\Delta_m}(m,1),\ldots,\underline{u}'_{\Delta_m}(m,\ell_m)],$$

and

$$\underline{\underline{\mathrm{P}}}_m=[\underline{p}'_m(m,0);\underline{p}'_m(m,1),\ldots,\underline{p}'_m(m,\ell_m),\ldots,\underline{p}'_m(M,1),\ldots,\underline{p}'_m(M,\ell_M)].$$

Remark 2.2. Every Markov chain is successively lumpable with respect to a partition $\mathcal{D}=\{D_0,D_1\}$ when $D_0=\{(0,\boldsymbol{\varepsilon}_0(\mathcal{D}))\}$ is any single state and D_1 contains the remaining states.

We can now state and prove the following proposition regarding successively lumpable Markov chains.

Proposition 2.2. *If Markov chain $X_0(t)$ is successively lumpable with respect to the partition $\mathcal{D}_0$, then $X_m(t)$ is successively lumpable with respect to partition $\mathcal{D}_m$, for all $m=1,\ldots,M$.*

Proof. To complete a proof with induction we need to show that if $X_m(t)$ is successively lumpable with respect to partition $\mathcal{D}_m$, then $X_{m+1}(t)$ is successively lumpable with respect to partition $\mathcal{D}_{m+1}$.

For $m=0,$ Definition 2.2 holds by assumption on $\underline{\underline{\mathrm{P}}}_0(=\underline{\underline{\mathrm{P}}})$. We assume the induction holds for $k=0,\ldots,m$ and we show it holds for $m+1$. We have defined $\underline{\underline{\mathrm{P}}}_{m+1}$ in Eq. (2.7)-(2.8).

To prove that $X_{m+1}(t)$ is successively lumpable with respect to $\mathcal{D}_{m+1}$ we first show that Δ_{m+1} has an entrance state $(m+1, \epsilon_{m+1}(\mathcal{D}_{m+1}))$.

By induction we know that $\Delta_m \cup D_{m+1}$ has an entrance state in $X_m(t)$: either $(m, \epsilon_m(\mathcal{D}_m))$ or $(m+1, i_1)$, a state in D_{m+1}. Furthermore, we know by Eq. (2.8) that for $i \neq 0$, with $k > m+1$:

$$p_{m+1}(m+1, i \,|\, k, j) = p_m(m+1, i \,|\, k, j), \tag{2.9}$$

and for $i = 0$:

$$p_{m+1}(m+1, 0 \,|\, k, j) = \sum_{(m,i') \in \Delta_m} p_m(m, i' \,|\, k, j). \tag{2.10}$$

Now, if $(m, \epsilon_m(\mathcal{D}_m))$ is the entrance state of $\Delta_m \cup D_{m+1}$ in $X_m(t)$ we get by Eq. (2.9) that $p_{m+1}(m+1, i \,|\, k, j) = 0$ for all $i > 0, k > m+1$ and thus that $(m+1, 0)$ is the entrance state of Δ_{m+1} in $X_{m+1}(t)$.

If $(m+1, i_1)$ is the entrance state of $\Delta_m \cup D_{m+1}$ in $X_m(t)$, we know by Eq. (2.9) that $p_{m+1}(m+1, i \,|\, k, j) = 0$ for all i except i_1 and by Eq.(2.10) that $p_{m+1}(m+1, 0 \,|\, k, j) = 0$. Thus $(m+1, i_1)$ is the entrance state of Δ_{m+1} in $X_{m+1}(t)$.

With a similar argument we can prove that $\Delta_{m+1} \cup \ldots \cup D_i$ has an entrance state in $X_{m+1}(t)$ for all i. Thus $X_{m+1}(t)$ is successively lumpable with respect to $\mathcal{D}_{m+1}$ when $X_m(t)$ is successively lumpable with respect to $\mathcal{D}_m$. □

Remark 2.3. Because of Proposition 2.2 we know that Δ_m has an entrance state in $X_m(t)$ for all $m \leq M$. In the construction of $Z_m(t)$, (m, η_m) was chosen arbitrarily. From now on we choose $(m, \eta_m) = (m, \epsilon_m(\mathcal{D}_m))$. Then $Z_m(t)$ is irreducible and positive recurrent as can be easily seen from a graphical representation.

We can now state the following.

Theorem 2.2. *Under the assumption of Proposition 2.2 the following are true:*

i)

$$\upsilon_{\Delta_m}(m, i) = \frac{\pi_m(m, i)}{\sum_{(m,i') \in \Delta_m} \pi_m(m, i')}. \tag{2.11}$$

ii)

$$\pi_{m+1}(k, j) = \begin{cases} \displaystyle\sum_{(m,i') \in \Delta_m} \pi_m(m, i'), & \text{if } (k, j) = (m+1, 0), \\ \pi_m(k, j), & \text{otherwise.} \end{cases} \tag{2.12}$$

Proof. The proof is easy to complete by induction using a similar derivation as in Proposition 2.1 and Theorem 2.1, combined with the induction result of Proposition 2.2. □

The previous results imply that the following theorem holds.

Theorem 2.3. *If $X_0(t)$ is successively lumpable with $|\mathcal{X}_0| < \infty$ the following is true:*

$$\pi_0(m,j) = \upsilon_{\Delta_m}(m,j) \prod_{k=m+1}^{M} \upsilon_{\Delta_k}(k,0),\ \forall (m,j) \in \mathcal{X}_0.$$

Proof. The proof follows by induction on decreasing values of $n = M, M-1, \ldots, 0$ for fixed M; note that $|\mathcal{X}_0| < \infty$ implies that M is finite.

For $n = M$, we need to show that

$$\pi_0(M,j) = \upsilon_{\Delta_M}(M,j), \forall (M,j) \in D_M.$$

Indeed, by Theorem 2.2, we have $\upsilon_{\Delta_M}(M,j) = \pi_M(M,j)/1$, where the denominator is 1 because Δ_M contains all states of $\mathcal{X}_M$. Since $j \neq 0$, (i.e. (M,j) has never been lumped by our lumping procedure) by using Theorem 2.2 repeatedly we obtain $\pi_M(M,j) = \pi_{M-1}(M,j) = \ldots = \pi_0(M,j)$, and the proof is complete for $n = M$.

We next show that the claim is true for $n = M-1$, assuming it is true for $n = M$. So we will show that:

$$\pi_0(M-1,j) = \upsilon_{\Delta_{M-1}}(M-1,j) \prod_{k=M}^{M} \upsilon_{\Delta_k}(k,0).$$

The right hand side of the above is

$$\begin{aligned}
\upsilon_{\Delta_{M-1}}(M-1,j)\upsilon_{\Delta_M}(M,0) &= \frac{\pi_{M-1}(M-1,j)}{\sum_{(M-1,j')\in\Delta_{M-1}} \pi_{M-1}(M-1,j')}\, \upsilon_{\Delta_M}(M,0) \\
&= \frac{\pi_{M-1}(M-1,j)}{\sum_{(M-1,j')\in\Delta_{M-1}} \pi_{M-1}(M-1,j')} \frac{\pi_M(M,0)}{\sum_{(M,j')\in\Delta_M} \pi_M(M,j')} \\
&= \pi_{M-1}(M-1,j),
\end{aligned}$$

where the first two equalities follow from Theorem 2.2, Eq. (2.11). The last equality uses Eq. (2.12) and the fact that $\sum_{(M,\ell)\in\Delta_M} \pi_M(M,\ell) = 1$, as before. The proof for $n = M-1$ is complete when we observe that $\pi_{M-1}(M-1,j) = \pi_{M-2}(M-1,j) = \ldots = \pi_0(M-1,j)$ since $j \neq 0$, as in the case when $n = M$.

The induction step from n to $n-1$ is easy to complete using similar algebra with albeit more cluttered equations. □

2.2.3 The algorithm and an example

Using the construction and results of Theorem 2.3 of the previous section, we can now state an algorithm for computing the stationary probability vector $\underline{\pi}$, of a successively lumpable Markov chain with respect to partition $\mathcal{D}$ as below.

Algorithm 2.1. *Sucessive Lumping*

1 Construct $\underline{\underline{\mathrm{U}}}_{D_0}$, cf., Eq. (2.1).
2 Calculate $\underline{v}_{D_0}$.
3 Lump D_0 to $(1,0)$ and let $\Delta_1 = \{(1,0)\} \cup D_1$.
Set $m = 1$.
While $m \leq M$
4.1 Construct $\underline{\underline{\mathrm{U}}}_{\Delta_m}$ cf., Eq. (2.6).
4.2 Calculate $\underline{u'}_{\Delta_m}$.
4.3 Lump Δ_m to $(m+1,0)$ and let $\Delta_{m+1} = (m+1,0) \cup D_m$.
$m = m + 1$
End
5 Calculate $\underline{\pi}$, cf., Theorem 2.3.

We next clarify the previous results with a small example.

Example 1. For clarity we will number the state space according to the notation introduced in Section 2.2, so we take:
$\mathcal{X} = \{(0,1),(0,2),(1,1),(1,2),(2,1),(2,2),(3,1),(3,2),(3,3)\}$,
with a partition $\mathcal{D} = \{D_0, \ldots, D_3\}$ where $D_0 = \{(0,1),(0,2)\}, D_1 = \{(1,1),(1,2)\}$, $D_2 = \{(2,1),(2,2)\}$ $D_3 = \{(3,1),(3,2),(3,3)\}$ and transition matrix $\underline{\underline{\mathrm{P}}}$:

$$\underline{\underline{\mathrm{P}}} = \left[\begin{array}{c|ccccccccc} & (0,1) & (0,2) & (1,1) & (1,2) & (2,1) & (2,2) & (3,1) & (3,2) & (3,3) \\ \hline (0,1) & 0 & 1/3 & 5/9 & 0 & 0 & 0 & 0 & 1/9 & 0 \\ (0,2) & 0 & 0 & 1/3 & 2/3 & 0 & 0 & 0 & 0 & 0 \\ (1,1) & 0 & 0 & 0 & 1/6 & 2/3 & 0 & 1/6 & 0 & 0 \\ (1,2) & 0 & 0 & 0 & 0 & 1/6 & 3/4 & 0 & 0 & 1/12 \\ (2,1) & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ (2,2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ (3,1) & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 \\ (3,2) & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 \\ (3,3) & 1/2 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 \end{array}\right]$$

The transition diagram of the corresponding Markov chain $X(t)$ is given in Figure 2.1. It is easy to see that $X(t)$ is successively lumpable with respect to the partition $\mathcal{D}$. The first steps of the algorithm are:

1 $\underline{\underline{\mathrm{U}}}_{\Delta_0} = \begin{bmatrix} 2/3 & 1/3 \\ 1 & 0 \end{bmatrix}$.

2 $\underline{v}_{\Delta_0} = [3/4, 1/4]$.

Next, we continue with $D_1 \neq \emptyset$, and note that for every positive $p(k', j' \mid k, j)$ with $(k', j') \in D_1$ we have $(k, j) \in D_0 \cup D_1$.

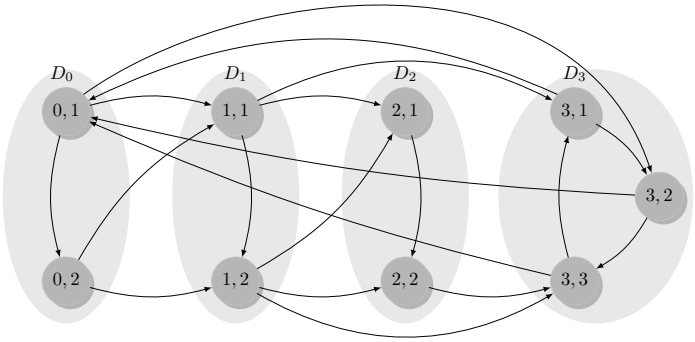


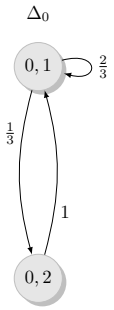
Figure 2.1: Transition diagram of a successively lumpable Markov chain $X(t)$, arrows represent possible transitions under $\underline{\underline{\mathrm{P}}}$.

3 Lump $\{(0,1),(0,2)\}$ to $(1,0)$ and let $\Delta_1 = \{(1,0),(1,1),(1,2)\}$.

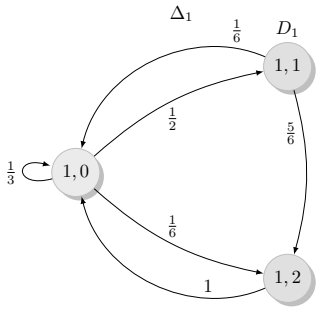
4.1 $\underline{\underline{\mathrm{U}}}_{\Delta_1} = \begin{bmatrix} 1/3 & 1/2 & 1/6 \\ 5/6 & 0 & 1/6 \\ 1 & 0 & 0 \end{bmatrix}$.

4.2 $\underline{v}_{\Delta_1} = [4/7, 2/7, 1/7]$.

Figure 2.2b illustrates the transition diagram of this $Z_1(t)$ chain.



(a) Transition diagram of $\underline{\underline{\mathrm{U}}}_{\Delta_0}$.



(b) Transition diagram of $\underline{\underline{\mathrm{U}}}_{\Delta_1}$

Figure 2.2: First iteration.

4.3 Lump $\{(1,0),(1,1),(1,2)\}$ to $(2,0)$ and let $\Delta_2 = \{(2,0),(2,1),(2,2)\}$. Note that since we know $\underline{\upsilon}_{\Delta_1}$ we can construct transition probabilities of a set Δ_2 without knowledge of $\underline{\pi}$ with the use of the previous states $\{(1,0),(1,1),(1,2)\}$.

4.1 $\underline{\underline{\mathrm{U}}}_{\Delta_2} = \begin{bmatrix} 19/28 & 3/14 & 3/28 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$.

4.2 $\underline{\upsilon}_{\Delta_2} = [28/43, 6/43, 9/43]$.

Next we look at subset D_3 and repeat the previous.

4.3 Lump $\{(2,0),(2,1),(2,2)\}$ to $(3,0)$ and let $\Delta_3 = \{(3,0),(3,1),(3,2),(3,3)\}$.

4.1 $\underline{\underline{\mathrm{U}}}_{\Delta_3} = \begin{bmatrix} 93/129 & 4/129 & 4/129 & 4/129 \\ 1/2 & 0 & 1/2 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 1/2 & 1/2 & 0 & 0 \end{bmatrix}$.

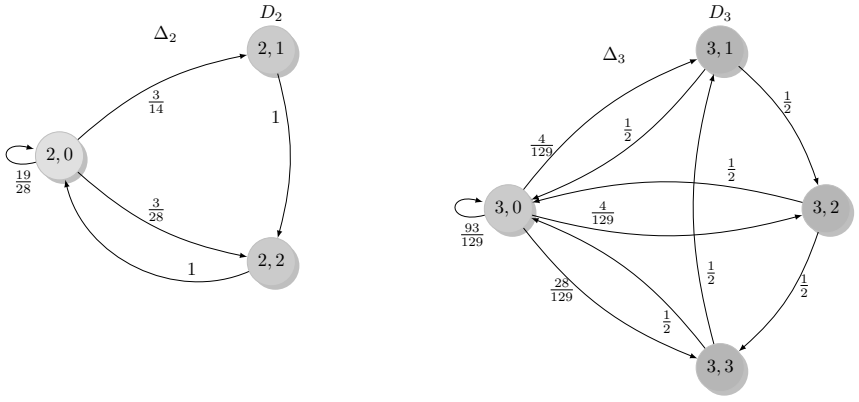
4.2 $\underline{\upsilon}_{\Delta_3} = [43/67, 142/1407, 104/1407, 248/1407]$.

We advance to step 5 and calculate $\underline{\pi}$:

$$
\begin{aligned}
\pi(0,1) &= \upsilon_{\Delta_3}(3,0)\upsilon_{\Delta_2}(2,0)\upsilon_{\Delta_1}(1,0)\upsilon_{\Delta_0}(0,1) = \frac{43}{67}\frac{28}{43}\frac{4}{7}\frac{3}{4} = \frac{12}{67} \\
\pi(0,2) &= \upsilon_{\Delta_3}(3,0)\upsilon_{\Delta_2}(2,0)\upsilon_{\Delta_1}(1,0)\upsilon_{\Delta_0}(0,2) = \frac{43}{67}\frac{28}{43}\frac{4}{7}\frac{1}{4} = \frac{4}{67} \\
\pi(1,1) &= \upsilon_{\Delta_3}(3,0)\upsilon_{\Delta_2}(2,0)\upsilon_{\Delta_1}(1,1) = \frac{43}{67}\frac{28}{43}\frac{2}{7} = \frac{8}{67} \\
\pi(1,2) &= \upsilon_{\Delta_3}(3,0)\upsilon_{\Delta_2}(2,0)\upsilon_{\Delta_1}(1,2) = \frac{43}{67}\frac{28}{43}\frac{1}{7} = \frac{4}{67} \\
\pi(2,1) &= \upsilon_{\Delta_3}(3,0)\upsilon_{\Delta_2}(2,1) = \frac{43}{67}\frac{6}{43} = \frac{6}{67} \\
\pi(2,2) &= \upsilon_{\Delta_3}(3,0)\upsilon_{\Delta_2}(2,2) = \frac{43}{67}\frac{9}{43} = \frac{9}{67} \\
\pi(3,1) &= \upsilon_{\Delta_3}(3,1) = \frac{142}{1407} \\
\pi(3,2) &= \upsilon_{\Delta_3}(3,2) = \frac{104}{1407} \\
\pi(3,3) &= \upsilon_{\Delta_3}(3,3) = \frac{248}{1407}.
\end{aligned}
$$

Remark 2.4.

i) To illustrate the fact that a Markov chain can be successively lumped with respect to different partitions, Figure 2.4 shows its transition diagram of the chain of example 1, where highlighted areas represent the sets of a different partition $\mathcal{D}'$ and it is easy to see that the chain is also successively lumpable with respect to partition $\mathcal{D}'$.



(a) Graphical representation of $\underline{\mathrm{U}}_{\Delta_2}$.

(b) Transition diagram of $\underline{\mathrm{U}}_{\Delta_3}$.

Figure 2.3: Second iteration.

ii) Figure 2.5 illustrates the transition diagram of a Markov chain. An arrow from a state (m, j) to a state (m', j') is present only if the corresponding transition probability $p(m', j' \mid m, j)$ is positive; where we ignore "loop" transitions with $p(m, j \mid m, j) > 0$ that do not play a role in determining successive lumpability. The Markov chain corresponding to this transition diagram is successively lumpable with respect to partition $\mathcal{D}$ that consists of the four highlighted sets of states. This picture is interesting since it is easy to see that "adding" any additional (non-loop) arrows (i.e., transition(s), with positive probability) will result in a transition diagram of a chain for which the successive lumpability property does not hold.

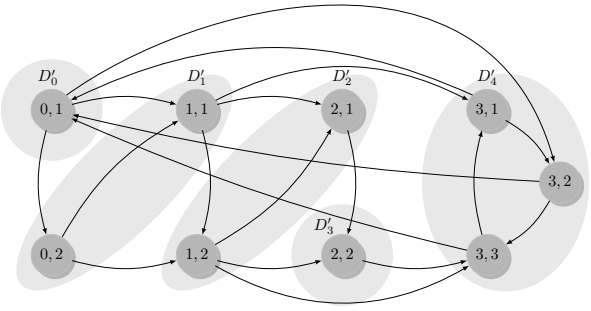


Figure 2.4: Transition diagram of $X(t)$, of Figure 1, with state space $\mathcal{X}$ partitioned by $\mathcal{D}' = \{D'_0, D'_1, D'_2, D'_3, D'_4\}$.

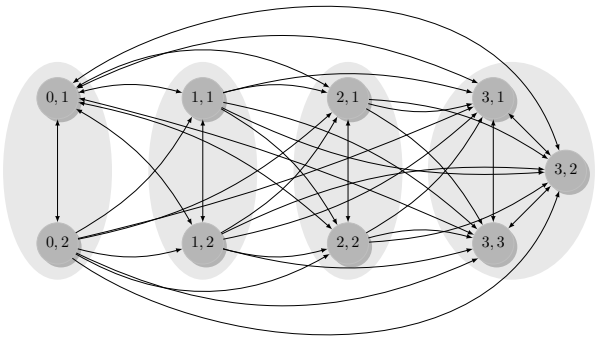


Figure 2.5: Transition diagram of a process with maximum number of positive probability transitions that is successively lumpable, cf., Remark 2.4.

2.3 Multiple successively lumpable Markov chains

The main result of Section 2.2 is that when a Markov chain is successively lumpable its steady state probability vector can be calculated using successive lumping. In Section 2.3.1 we will show that it is possible to have multiple lumpable structures in one Markov chain. In this case it is possible to calculate the steady state vector using multiple times successive lumping. We also establish a product form expression, for finite state spaces. We present the algorithm and an example in Section 2.3.2.

2.3.1 Definitions and proofs for multiple successive lumping

Let $X(t)$ be a Markov chain on a finite (or countable) state space $\mathcal{X}$ with transition matrix $\underline{\underline{P}}$. We will assume that the state space $\mathcal{X}$ is composed of $N \leq \infty$ mutually exclusive and exhaustive sets, i.e.

$$\mathcal{X} = \bigcup_{n=1}^{N} \mathcal{X}^n,$$

where each subset $\mathcal{X}^n$ can be partitioned into a (possibly infinite) sequence of

$$\mathcal{D}^n = \{D_0^n, \ldots, D_{M_n}^n\}.$$

Alternatively the partition

$$\mathcal{D} = \{D_0^1, \ldots, D_{M_1}^1, \ldots, D_0^N, \ldots, D_{M_N}^N\}$$

is a sequence of $N \leq \infty$ subpartitions of $\mathcal{X}$. For notational convenience, the elements of each set D_m^n will be relabelled to a triple-notation as $\{(n,m,1),(n,m,2),\ldots,(n,m,\ell_{(n,m)})\}$,

for given constants $\ell_{(n,m)} \leq \infty$. After this state relabeling, the transition matrix of $X(t)$ will be denoted by $\underline{\underline{\mathrm{P}}} = [p(n', m', j' \,|\, n, m, j)]$, where the $((n, m, j)\,, (n', m', j'))$ element is given by

$$p(n', m', j' \,|\, n, m, j) = \mathbf{Pr}[X(t+1) = (n', m', j') \,|\, X(t) = (n, m, j)].$$

In the sequel, to avoid trivial cases we assume that $N \geq 2$, i.e., the partition $\mathcal{D}$ has at least two subsets.

The definition of an **entrance state** in this triple index notation is as follows.

Definition 2.3. A subset D_m^n of $\mathcal{D}$ has an **entrance state** $(n, m, \boldsymbol{\varepsilon}_{n,m}(\mathcal{D})) \in D_m^n$ iff for all $n' \neq n$, $m' \neq m$ and $j \neq \boldsymbol{\varepsilon}_{n,m}(\mathcal{D})$,

$$p(n, m, j \,|\, n', m', j') = 0.$$

Given a Markov chain $X(t)$ with partition $\mathcal{D}$ for which every subset $\mathcal{X}^n$ has an entrance state $(n, m, \epsilon_{n,m}(\mathcal{D}))$, we next study the Markov chains $X^n(t)$ on state space $\mathcal{X}^n$. We state the following definition about their transition matrix $\underline{\underline{\mathrm{P}}}^n$.

Definition 2.4. We call $X^n(t)$ the $\boldsymbol{n^{th}}$**- component Markov chain** of $X(t)$ when the $(n, m', j' \,|\, n, m, j)$-element of $\underline{\underline{P}}^n$ is defined as follows:

a) If $(n, m', j') = (n, m', \boldsymbol{\varepsilon}_{n,m'}(\mathcal{D}))$, then

$$p^n(n, m', j' \,|\, n, m, j) = p(n, m', j' \,|\, n, m, j) + \sum_{(n', m'', j'') \notin \mathcal{X}^n} p(n', m'', j'' \,|\, n, m, j).$$

b) Otherwise,

$$p^n(n, m', j' \,|\, n, m, j) = p(n, m', j' \,|\, n, m, j).$$

We can now state the following.

Definition 2.5. A Markov chain $X(t)$ is called **multiple successively lumpable** with respect to partition $\mathcal{D} = \{\mathcal{D}^1, \ldots, \mathcal{D}^N\}$ if and only if the following conditions hold.

a) $D_0^n \cup D_1^n \cup \ldots \cup D_{M_n}^n$ has an entrance state in $X(t)$ for all $n = 1, \ldots, N$.

b) $D_0^n \cup D_1^n \cup \ldots \cup D_{m_n}^n$ has an entrance state in $X^n(t)$ for all $n = 1, \ldots, N$ and for all $m_n = 0, \ldots, M_n$.

Note that condition (a) makes the assertion that any state (n', m', j') in $D_{m'}^{n'}$ can not be entered from a state (n, m, j) in D_m^n, except when $(n', m', j') = (n', m', \boldsymbol{\varepsilon}_{n',m'}(\mathcal{D}))$. Condition (b) asserts that a state (n, m', j') in $D_{m'}^n$ can not be entered from a state (n, m, j) in D_m^n when $m' < m$, except when (n, m', j') is the entrance state of $D_0^n \cup \ldots \cup D_{m'}^n$ in $X^n(t)$.

We can now state the following lemma which shows that a multiple successively lumpable Markov chain is indeed several times successively lumpable.

Lemma 2.1. *When $X(t)$ is multiple successively lumpable with respect to $\mathcal{D}$, the n^{th}- component-Markov chain $X^n(t)$, is successively lumpable with respect to $\mathcal{D}^n$ for all $n \leq N$.*

Proof. Using their construction (cf. Definition 2.4 (a) and (b)) the transition probabilities $p^n(n, m', j' \,|\, n, m, j)$ can be shown to satisfy the conditions of Definition 2.5. □

We next introduce some notation, extending that of Section 2.2.

1. On the n^{th}- component-Markov chain $X^n(t)$ of $X(t)$ we define $\Delta_0^n = D_0^n$, $\Delta_1^n = \{(n,1,0)\} \cup D_1^n$, $\Delta_m^n = \{(n,m,0)\} \cup D_m^n$, where states $(n,m,0)$ are lumped states representing $\bigcup_{k=0}^{m-1} D_k^n$.

   For notational consistency we will use the notation: $X_0^n(t) = X^n(t)$, $\mathcal{X}_0^n = \mathcal{X}^n$, $\mathcal{D}_0^n = \mathcal{D}^n$, and $\underline{\underline{\mathrm{P}}}_0^n = \underline{\underline{\mathrm{P}}}$. Further, we consider $X_m^n(t)$ on $\mathcal{X}_m^n$ with transitions $\underline{\underline{\mathrm{P}}}_m^n$.

2. Analogously to the chains $Z_m(t)$ defined in Section 2.2, we define Markov chains $Z_m^n(t)$ with state space Δ_m^n and transition matrix $\underline{\underline{\mathrm{U}}}_{\Delta_n^m}^n$.

3. Also, we define:

$$\begin{aligned}
\pi(n,m,j) &= \lim_{t\to\infty} \mathbf{Pr}[X(t) = (n,m,j)], \\
\pi^n(n,m,j) &= \lim_{t\to\infty} \mathbf{Pr}[X^n(t) = (n,m,j)], \\
\upsilon_{\Delta_n^m}^n(j) &= \lim_{t\to\infty} \mathbf{Pr}[Z_m^n(t) = (j)].
\end{aligned}$$

4. Similarly, we define $\underline{\pi}$, $\underline{\pi}^n$, $\underline{\upsilon}_{\Delta_n^m}^n$ to be the corresponding probability vectors; with dimensions $\prod_{n=1}^N \prod_{m=0}^{M_n} \ell_m^n$, $\prod_{m=0}^{M_n} \ell_m^n$, $\ell_m^n + \delta(m)$ respectively, where in the last expression the term "$\delta(m)$" is equal to one if $m > 0$, and equal to zero when $m = 0$ (note that no artificial state in Δ_m^n is used when $m = 0$).

   The elements of $\underline{\underline{\mathrm{U}}}_{\Delta_n^m}^n$ and $\underline{\underline{\mathrm{P}}}_{m+1}^n$ are akin to the elements of $\underline{\underline{\mathrm{U}}}_{\Delta_m}$ and $\underline{\underline{\mathrm{P}}}_{m+1}$, cf., Eq. (2.6)-(2.8).

5. Finally, we define a chain $Y(t)$ with state space $\mathcal{E} = \{1, \ldots, N\}$ and transition matrix $\underline{\underline{\mathrm{Q}}}$ with its (n, n') element being equal to:

$$q(n' \,|\, n) = \sum_{(n',m',j') \in \mathcal{X}^{n'}} \sum_{(n,m,j) \in \mathcal{X}^n} \pi^n(n,m,j) p(n',m',j' \,|\, n,m,j). \qquad (2.13)$$

Note that the chain $Y(t)$ can be viewed as a chain between the different "lumped" successively lumpable chains. We will use the notation $\sigma(n)$ for the steady state probabilities of the above chain, i.e., $\sigma(n) = \lim_{t\to\infty} \mathbf{Pr}[\, Y(t) = n\,]$.

We will next show the following:

Lemma 2.2. *Assuming that $X(t)$ is a multiple successively lumpable Markov chain with its n^{th}- component Markov chain $X^n(t)$ defined as above, the following is true:*

$$\pi^n(n,m,j) = \frac{\pi(n,m,j)}{\sum_{(n,m',j')\in\mathcal{X}^n} \pi(n,m',j')}.$$

Proof. It is clear that $\underline{\pi}^n\, \underline{1}\,' = 1$. Now from Definition 2.5 we see that $\mathcal{X}^n$ has an entrance state $(n, m, \boldsymbol{\varepsilon}_{n,m}(\mathcal{D}))$ and therefore we can use a similar derivation as is used in Proposition 2.1 to complete the proof. □

Proposition 2.3. *For a multiple successively lumpable Markov chain $X(t)$ and with $Y(t)$ defined as above, the following is true:*

$$\sigma(n) = \sum_{(n,m,j)\in\mathcal{X}^n} \pi(n,m,j).$$

Proof. It is clear that $\underline{\sigma}\, \underline{1}\,' = 1$. It suffices to prove that the above choice of $\underline{\sigma}$ is the solution of the steady state equations, of the $Y(t)$ process, below:

$$\sigma(n') = \sum_{n=1}^{N} \sigma(n) q(n'|n) \;\text{ for } n' = 1, 2, \ldots, N.$$

Indeed:

$$\begin{aligned}
\sum_{n=1}^{N} \sigma(n) q(n'|n) &= \sum_{n=1}^{N} \sigma(n) \sum_{(n',m',j')\in\mathcal{X}^{n'}} \sum_{(n,m,j)\in\mathcal{X}^n} \pi^n(n,m,j) p(n',m',j' \,|\, n,m,j) \\
&= \sum_{n=1}^{N} \sum_{(n,m,j)\in\mathcal{X}^n} \pi(n,m,j) \sum_{(n',m',j')\in\mathcal{X}^{n'}} \sum_{(n,m,j)\in\mathcal{X}^n} \frac{\pi(n,m,j) p(n',m',j' \,|\, n,m,j)}{\sum_{(n,m',j')\in\mathcal{X}^n} \pi(n,m',j')} \\
&= \sum_{n=1}^{N} \sum_{(n',m',j')\in\mathcal{X}^{n'}} \sum_{(n,m,j)\in\mathcal{X}^n} \pi(n,m,j) p(n',m',j' \,|\, n,m,j) \quad (2.14) \\
&= \sum_{(n',m',j')\in\mathcal{X}^{n'}} \sum_{n=1}^{N} \sum_{(n,m,j)\in\mathcal{X}^n} \pi(n,m,j) p(n',m',j' \,|\, n,m,j) \quad (2.15)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{(n',m',j')\in\mathcal{X}^{n'}} \sum_{(n,m,j)\in\mathcal{X}} \pi(n,m,j)p(n',m',j' \mid n,m,j) \\
&= \sum_{(n',m',j')\in\mathcal{X}^{n'}} \pi(n',m',j') \\
&= \sigma(n').
\end{aligned}
$$

The second equality above follows from Lemma 2.2. It is clear that the summations in Eq. (2.14) and (2.15) can be interchanged freely. □

The main result of this section is the next theorem, for a multiple successively lumpable Markov chain $X(t)$ with respect to a partition $\mathcal{D}$ and with $|\mathcal{X}| < \infty$.

Theorem 2.4. *If $X(t)$ is multiple successively lumpable with respect to partition $\mathcal{D}$ and $|\mathcal{X}| < \infty$ then:*

$$\pi(n,m,j) = \sigma(n)\upsilon^n_{\Delta^m_n}(n,m,j) \prod_{k=m+1}^{M} \upsilon^n_{\Delta^k_n}(n,k,0) \text{ for all } (n,m,j) \in \mathcal{X}.$$

Proof. Since by Lemma 2.1, $X^n(t)$ is a successively lumpable Markov chain with respect to partition $\mathcal{D}^n$ we know by Theorem 2.3 that for all n

$$\pi^n(n,m,j) = \upsilon^n_{\Delta^m_n}(n,m,j) \prod_{k=m+1}^{M} \upsilon^n_{\Delta^k_n}(n,k,0).$$

The proof is easy to complete using Lemma 2.2 and Proposition 2.3. □

Remark 2.5. When $M_n = 1$ for all n, then Theorem 2.4 becomes the main result in [39].

Remark 2.6. For a multiple successively lumpable Markov chain we can solve a total of $\prod_{n=1}^{N} M_n$ Markov chains of sizes $\ell^n_{m_n} + \delta(m_n)$ each, instead of one big system of size:

$$\prod_{n=1}^{N} \prod_{m=0}^{M_n} \ell^n_m.$$

For example, if $N = 10^4$, $M_n = 10^2$ for all n and $l^n_{m_n} = 10^4$ for all n, m, we need to solve 10^6 systems of size 10^4 instead of 1 of size 10^{10}.

2.3.2 The algorithm and an example

Similarly to Algorithm 2.1 for a successively lumpable Markov chain presented in Section 2.2.3, we state an algorithm for a Markov chain that is multiple successively lumpable with respect to a partition $\mathcal{D} = \{\mathcal{D}^1, \ldots, \mathcal{D}^N\}$. Again, this algorithm does not require a proof, it is a direct result of Theorem 2.4.

Algorithm 2.2. *Multiple Successive Lumping*

For $n = 1, \ldots N$
1.1 Construct $X^n(t)$ with Def. 2.4.
1.2 Call Algorithm SL and solve $X^n(t)$.
End
2 Construct $\underline{\underline{\mathrm{Q}}}$, cf., Eq. (2.13).
3 Calculate $\underline{\sigma}$ with Proposition 2.3.
4 Calculate $\underline{\pi}$, cf., Theorem 2.4.

To clarify the algorithm, Figure 2.6 shows a multiple successively lumpable Markov chain, with $N = 2, M_1 = 2, M_2 = 2, \ell^1_{1_1} = 2, \ell^2_{1_2} = 3, \ell^1_{2_1} = 2, \ell^2_{2_2} = 3$.

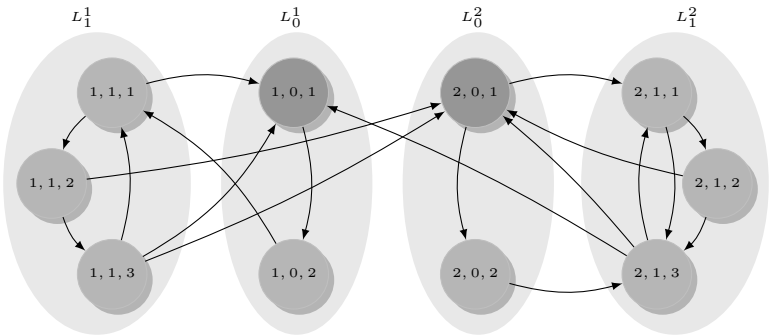


Figure 2.6: Transition diagram of a multiple successively lumpable Markov chain.

It is easy to see that in this example both $(1, 0, 1)$ and $(2, 0, 1)$ are the *entrance states* of D^1_0 and D^2_0 respectively. The procedure will solve $\mathcal{D}^1$ and $\mathcal{D}^2$ separately as successive lumpable Markov chains and conclude with solving a chain $Y(t)$ between two states representing $\mathcal{D}^1$ and $\mathcal{D}^2$. To do so, the arrows from $(1, 1, 2)$ and $(1, 1, 3)$ to $(2, 0, 1)$ will be "redirected" to $(1, 0, 1)$ and the arrow from $(2, 1, 3)$ to $(1, 0, 1)$ redirected to $(2, 0, 1)$ as described above.

2.4 Extension to semi-Markov and continuous time processes

It is possible to extend the successive lumpable and the multiple successive lumpable theory to semi-Markov or a continuous time Markov processes either directly or using the following construction and notation, cf. [90]. A process $\{\mathcal{Z}(t),\ t \geq 0\}$ is a semi-Markov process, on a state space $\mathcal{X}$, if it can be constructed as follows.

i) Transitions from state to state are generated from an "embedded" discrete time chain $\{X_n\ ,\ n = 0, 1, \ldots\}$ with state space $\mathcal{X}$ and transition matrix $\underline{\underline{\mathrm{P}}} = [p_{ij}]$.

ii) The sojourn times (durations of a visit) for any state i are i.i.d. non-negative random variables T_i^m, $m \geq 1$, having an arbitrary distribution F_i, i.e. they are distributed as a random variable T_i with $F_i(t) = \mathbf{Pr}\,(T_i \leq t \,|\, \mathcal{Z}(0) = i)$.

A notable special case is that of a continuous time Markov process, in which case the F_i's are all exponential distributions with parameters λ_i. To avoid trivial cases with instantaneous states, it is assumed that the expected durations $\mu_i = \mathbf{E}T_i$ are finite positive constants.

Let $\mathcal{O}_i^n$ denote the time spent in states other than state i between the n^{th} and $n+1^{st}$ visits to state i. The above assumptions imply that $\mathcal{O}_i^n$, $n = 1, 2, \ldots$ are i.i.d. random variables, i.e., they are distributed as a random variable $\mathcal{O}_i$, having a distribution $G_i(t)$. The main results concerning the long run steady state probabilities ϖ_i, and π_i of $\mathcal{Z}(t)$ and X_n, respectively, are summarized in the next Theorem, cf. [90].

Theorem 2.5. *If $\underline{\underline{\mathrm{P}}}$ is irreducible and if the distribution of $T_i + \mathcal{O}_i$ is non-lattice then the limit $\lim_{t\to\infty} \mathbf{Pr}(\mathcal{Z}(t) = i|\mathcal{Z}(0) = j)$ exists, it is independent of the initial state $\mathcal{Z}(0) = j$ and it is equal to:*

$$\begin{aligned} \varpi_i &= \frac{\mathbf{E}T_i}{\mathbf{E}T_i + \mathbf{E}\mathcal{O}_i} \\ &= \frac{\pi_i \mathbf{E}T_i}{\sum_{j\in\mathcal{X}} \pi_j \mathbf{E}T_j}. \end{aligned} \tag{2.16}$$

Eq. (2.16) provides a method for computing the steady states probabilities ϖ_i, from π_i and the expected sojourn times $\mathbf{E}T_i$. It also makes possible a similar successive construction for continuous time or Semi-Markov processes using the expected sojourn times for all states.

For completeness, we next describe the successively lumpable process for a continuous time Markov process $X(t)$ on a state space $\mathcal{X}$ with a partition $\mathcal{D}$, of size M.

We revert back to the double index state notation of Section 2.2.

The transition rates of $X(t)$ will be denoted by $\mu(k', j' \mid k, j)$ where:

$$\mu(k, i \mid k, i) = - \sum_{(k', i') \neq (k, i)} \mu(k', i' \mid k, i).$$

It is easy to see that $X(t)$ is successively lumpable with respect to $\mathcal{D}$ if the conditions of Definition 2.2 are valid with $p(\cdot \mid \cdot)$ replaced by $\mu(\cdot \mid \cdot)$.

We provide below the rates of the corresponding processes $Z_m(t)$ and $X_m(t)$.

For the process $Z_0(t)$ on $D_0 (= \Delta_0)$, corresponding to the process described in Eq. (2.1), its $((0, i), (0, j))-$element can be shown to be as follows:

$$\lambda_{\Delta_0}(0, j \mid 0, i) = \begin{cases} \mu(0, j \mid 0, i), & \text{if } i \neq \varepsilon_0(\mathcal{D}), j \neq \varepsilon_0(\mathcal{D}), \\ \sum\limits_{(k, j') \notin D_0} \mu(k, j' \mid 0, i) + \mu(0, \varepsilon_0(\mathcal{D}) \mid 0, i), & \text{if } i \neq \varepsilon_0(\mathcal{D}), j = \varepsilon_0(\mathcal{D}), \\ \mu(0, j \mid 0, \varepsilon_0(\mathcal{D})), & \text{if } i = \varepsilon_0(\mathcal{D}), j \neq \varepsilon_0(\mathcal{D}), \\ - \sum\limits_{(0, j') \in D_0} \mu(0, j' \mid 0, \varepsilon_0(\mathcal{D})), & \text{if } i = j = \varepsilon_0(\mathcal{D}). \end{cases}$$

The transition rate matrix of the above rates is denoted as $\underline{\underline{\Lambda}}_{\Delta_0}$ and the steady state equations are:

$$\underline{\upsilon}_{\Delta_0} \underline{\underline{\Lambda}}_{\Delta_0} = 0,$$

and

$$\sum_{(0, i) \in D_0} \upsilon_{\Delta_0}(0, i) = 1,$$

where $\underline{\upsilon}_{\Delta_0}$ as before denotes the steady state probability vector.

As in Section 2.2 we can construct successively a sequence of processes $X_m(t)$ on $\mathcal{X}_m = \Delta_m \cup D_{m+1} \cup \ldots \cup D_M$ and $Z_m(t)$ on $\Delta_m = (m, 0) \cup D_m$ (with steady state vector $\underline{\upsilon}_{\Delta_m}$) as follows for $m = 1, 2 \ldots M$.

i) For the Markov process $X_m(t)$, the transition rates $\mu_m(k', j' \mid k, j)$ are defined as follows.

a) If $(k, j) = (m, 0)$ and $(k', j') \neq (m, 0)$:

$$\mu_m(k', j' \mid m, 0) = \sum_{(m-1, i) \in \Delta_{m-1}} \mu_{m-1}(k', j' \mid m-1, i) \upsilon_{\Delta_{m-1}}(m-1, i).$$

b) If $(k, j) \neq (m, 0)$:

$$\mu_m(k', j' \,|\, k, j) = \begin{cases} \mu_{m-1}(m-1, \boldsymbol{\varepsilon}_{m-1}(\mathcal{D}) \,|\, k, j), & \text{if } (k', j') = (m, 0), \\ \mu_{m-1}(k', j' \,|\, k, j), & \text{if } (k', j') \neq (m, 0). \end{cases}$$

c) And if $(k, j) = (k', j') = (m, 0)$:

$$\mu_m(m, 0 \,|\, m, 0) = -\sum_{(m,j) \in D_m} \mu_m(m, j | m, 0).$$

ii) For $Z_m(t)$:

$$\lambda_{\Delta_m}(m, j \,|\, m, i) = \begin{cases} \mu_m(m, j \,|\, m, i), & \text{if } i \neq \boldsymbol{\varepsilon}_m(\mathcal{D}), j \neq \boldsymbol{\varepsilon}_m(\mathcal{D}), \\ \mu_m(m, j \,|\, m, \boldsymbol{\varepsilon}_m(\mathcal{D})), & \text{if } i = \boldsymbol{\varepsilon}_m(\mathcal{D}), j \neq \boldsymbol{\varepsilon}_m(\mathcal{D}), \\ -\sum\limits_{(m,j') \in \Delta_m \setminus (m, \boldsymbol{\varepsilon}_m(\mathcal{D}))} \mu_m(m, j' \,|\, m, \boldsymbol{\varepsilon}_m(\mathcal{D})), & \text{if } i = j = \boldsymbol{\varepsilon}_m(\mathcal{D}). \end{cases}$$

and otherwise:

$$\lambda_{\Delta_m}(m, j \,|\, m, i) = \sum_{(k,j') \notin \Delta_m} \mu_m(k, j' \,|\, m, i) + \mu_m(m, \boldsymbol{\varepsilon}_m(\mathcal{D}) \,|\, m, i).$$

CHAPTER 3

QSF processes

This chapter appeared as: *DES and RES Processes and their Explicit Solutions*, cf. [S3].

3.1 Introduction to Chapter 3

In this chapter we study the class of *quasi skip free* processes, a subclass of Markov processes, for which the states can be specified by tuples of the form (m, i), where $m \in \mathbb{Z}$ represents the "current" level of the state and $i \in \mathbb{Z}^+$ the current phase of the state. The process is called *quasi skip free* (QSF) *to the left* (down) when its transition rate matrix Q (cf. Eq. (3.1)) does not permit one step transitions to states that are two or more levels away from the current state in the downwards direction of the level variable m. QSF processes generalize the well studied class of *quasi birth and death* processes (QBD) for which one step transitions to states that are two or more levels away from the current state in either direction of the level variable m are not allowed. *QSF to the right* (up) process can be defined with an apparent modification of the definition of the transition rate matrix Q. In the sequel for simplicity we will refer to a QSF process when the skip free direction is apparent (and without loss of generality taken to be the "down" direction).

Definition 3.1. The QSF processes under consideration are level dependent and ergodic processes with a transition rate matrix Q that satisfies one of the two following properties:

i) the *down entrance state* (DES) property: The structure of the non-zero elements of Q is such that *one step "down" transitions from a level m can only reach a single state in level $m-1$, for all levels m.*

ii) the *restart entrance state* (RES) property: The structure of the non-zero elements of Q is such that *one step "up" transitions from a level m can only reach a single state in level M_2 (M_2 is the highest level) for all levels m.*

The main results in this chapter are as follows. First, we derive explicit solutions, cf. Eq. (3.12) and (3.15), for the steady state probabilities of level dependent QSFs that satisfy the

DES property. Second, we use state truncation to derive tight bounds for the steady state probabilities. Then we derive explicit solutions, cf. Eq. (3.22) and (3.24), for the steady state probabilities of level dependent QSFs that satisfy the RES property. For these type of processes we also use state truncation to derive tight bounds for the steady state probabilities.

In Section 3.6 we show that the DES and the RES property are satisfied by many models that arise in practice and we obtain explicit solutions for the well-known open problems of the $M/Er/n$ and the $Er/M/n$ queues with batch arrivals. We note that there are no explicit expressions in the literature for the rate matrix sets (and hence for the invariant measures) for any of these models. This is due to the QSF structure of the $M/Er/n$ model with batch arrivals and to the infinite to the 'down' direction state space of the $Er/M/n$ model. For either of these cases little is known outside this present research. We also demonstrate the applicability of the method with an inventory model with random yield. Regarding the RES property we show that a well known reliability (restart) problem can be easily analysed using the proposed method. Even though the approaches used in this chapter use the successive lumping procedure described in Chapter 2 all the results presented in this chapter are useful by itself, since they provide solutions to notoriously hard to analyse processes.

Although our methodology applies to infinite values for the number of levels in the process in the downward and in the upward direction as well as for the number of phases, we take up explicitly only the case of finite values so that we can employ finite matrices in the analysis. In this way the basic features of the theory will not be obscured by additional formalism. However, we want to emphasize that all the results generalize to infinite values in a natural way using truncation, cf. Section 3.3.1 herein and [112], [102] and [94]. We will allow the lowest level value to be negative in order to have a natural state description for some models, for example, in the $Er/M/1$ queueing system this is necessary. We also note that the results hold for the case that either "boundary-side" of the state space is a transient class of states. However, for simplicity we will not consider this case explicitly.

The smaller class of level homogenous QBD (LHQBD) processes cf. [80], has been used to model systems in many areas including queueing theory, cf. [88], retrial queues, cf. [20]. For algorithmic usages of QBDs we refer to the anti-plagiarism scanner software of Viper. We note that most of the early literature devoted to level homogeneous QBDs typically follows the approach presented in Chapter 6, p. 129, in [70], whereby the computation of the steady state distribution is based on computing a rate matrix "R". This matrix is specified as one of the solutions of the, not easy to solve, matrix quadratic equation: $R^2D + RW + U = 0$, where we use our current (cf. Eq. (3.1)) notation: D, W, U for the matrices A_2 A_1, A_0 in [70]. Numerical methods for computing R involve cyclic reduction [30] and logarithmic reduction [82], [69]. For instance R is expressed in terms of a matrix G (such that $R = U(I-W-UG)^{-1}$) which is the solution of matrix quadratic equation: $UG^2+WG+D = 0$. For recent work on numerical methods for computing the matrix G for QBD processes of special structure we refer to [106], [38], [29] and references therein. A general approach to compute G, exists only in special cases when the down transition matrix has the form $c \cdot r$ with c a column vector and r a row vector normalized to one. We note that the DES property

is implied by the $c \cdot r$ structure but in our present setting we deal with the much more general class of QSF processes.

For level dependent QBD (LDQBD) processes we refer to and Chapters 8 and 12 of [70] and to [64]. In [33] recursive algorithms are given to compute the rate matrices. We note that when the LDQBD model described in [33] satisfies the DES property then we can provide explicit solutions for the rate matrix set.

In [70], Chapter 13, p. 268-270, a method is given to analyse level homogeneous QSF (LHQSF) processes by considering them as embedded processes in suitably defined QBDs. However, as stated therein, the success of this approach has been limited. We note that the matrix analytic method has been used in [84] to derive a recursive solution for the $M/G/1$ queue which is a QSF processes, using matrix 'G' described above.

The much wider class of QSF processes has the capacity to model much more general problems. Indeed in Section 3.6 we obtain explicit solutions for two well known queueing problems. In addition, QSF processes can be used to represent restart systems, cf. [59], [101] and [96], processes that represent inventory systems with random yield or lead times, reliability, cf. [56], and computer science and the theory of branching processes, cf. [34]. And as far as we know, explicit formulas for the rate matrix set of a QSF process have not been derived before.

The rest of this chapter is organized as follows. In Section 3.2 we introduce notation and formally define a QSF process. In Section 3.3 we show that the DES property implies the successive lumpable property of a QSF process. Then, we provide a recursive relation for the steady state probabilities between a level of the state space and its sub-levels. In Section 3.3.1 we show how the state space can be truncated, both in the downwards as in the upward directions. We follow the same line of thought in Section 3.4 and 3.4.1 when a QSF processes possesses the RES property. In Section 3.5 we show how our results when specialized to QBD processes provide simpler proofs and generalizations to well known theorems of the QBD literature. We will use the results of this chapter in the chapter on applications of this dissertation. In Section 3.6 we show how this methodology can be applied to the $M/Er/n$ queue with batch arrivals, to the $Er/M/n$ queue and to an inventory model with random yield.

3.2 Definitions and basic notation

A QSF to the left (or "down") process is a continuous time Markov process $X(t)$ on state space $\mathcal{X}$ that can be expressed as $\mathcal{X} = \bigcup_{m=M_1}^{M_2} \{(m,1),(m,2),\ldots,(m,\ell_m)\}$, where ℓ_m, M_1, M_2, are some fixed finite integers, with $1 \leq \ell_m < \infty$, $-\infty < M_1 < M_2 < \infty$, and $M_1 \leq m \leq M_2$. A state (m,i) specifies its "current" level m and its within the level state i, with $i = 1,\ldots,\ell_m$.

Remark 3.1. The notation in this chapter differs from the notation introduced in Chapter 2, since it handles a slightly different kind of structure on the (discrete time) Markov chains.

For a QSF process, the transition rate matrix has the form:

$$Q = \begin{bmatrix} W^{M_1} & U^{M_1,M_1+1} & \cdots & U^{M_1,m} & U^{M_1,m+1} & \cdots & U^{M_1,M_2-1} & U^{M_1,M_2} \\ D^{M_1-1} & W^{M_1+1} & \cdots & U^{M_1+1,m} & U^{M_1+1,m+1} & \cdots & U^{M_1+1,M_2-1} & U^{M_1+1,M_2} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & W^{m} & U^{m,m+1} & \cdots & U^{m-1,M_2-1} & U^{m-1,M_2} \\ 0 & 0 & \cdots & D^{m+1} & W^{m+1} & \cdots & U^{m,M_2-1} & U^{m,M_2} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & W^{M_2-1} & U^{M_2-1,M_2} \\ 0 & 0 & \cdots & 0 & 0 & \cdots & D^{M_2} & W^{M_2} \end{bmatrix}, \tag{3.1}$$

where in the above specification of Q we use the notation D^m, W^m, and $U^{m,k}$ to describe respectively the "down" (to level $m-1$), "within" (level m), and "up" (to level $k = m+1, m+2, \ldots, M_2$) transition rate sub-matrices in relation to the current level m of a state (m,i). The dimensions of these matrices are respectively $\ell_m \times \ell_{m-1}$, $\ell_m \times \ell_m$ and $\ell_m \times \ell_k$; the 0 sub-matrices of Q have position dependent dimensions so that Q is a well defined transition rate matrix.

Note that in the special case that there exist matrices D, W, U^k such that $D^m = D$, $W^m = W$ and $U^{m,k} = U^k$, for all m, the process is called (level) homogeneous. When $U^{m,m+k}$ are all 0 matrices for all $k \geq 2$ and all m, the QSF process reduces to the well studied QBD process, cf. [21].

In the sequel we will assume that the QSF processes discussed are ergodic. Explicit sufficient conditions for the elements of Q can be derived to support this claim, cf. Remark 3.5; such conditions on ergodicity for multi dimensional Markov Chains can be found in [100] and various criteria in [103], in [53] and in [97].

We consider Markov process $X(t)$, introduced above. Clearly the state space $\mathcal{X}$ of this process can be partitioned into a (possibly infinite) sequence of mutually exclusive and exhaustive *level sets*: $L_m = \{(m,1),(m,2),\ldots,(m,\ell_m)\}$.

Definition 3.2. For any fixed m we define the ***sub-level set*** of L_m to be the set of states $\underset{\sim}{L}_m = \cup_{k=M_1}^{m} L_k$ while the set $\widetilde{L}_m = \cup_{k=m}^{M_2} L_k$ is the ***super-level set*** of L_m.

We will express the steady state distribution of states in L_m in that of $\underset{\sim}{L}_m$. The presence of entrance states guarantee that sojourn times in $\underset{\sim}{L}_m$ are independent of the rate structure in $\widetilde{L}_m$.

We let $\pi(m,i)$ denote the steady state probability of state (m,i). The vectors $\pi^m := [\pi(m,1),\ldots,\pi(m,\ell_m)]$ and $\underset{\sim}{\pi}^m := [\pi^{M_1},\ldots,\pi^m]$, will denote respectively the steady state

probabilities of states in level L_m and sub-level $\underset{\sim}{L}_m$. The vector of the steady state probabilities over all states will be denoted by $\pi := [\pi^{M_1}, \ldots, \pi^{M_2}] = \underset{\sim}{\pi}^{M_2}$.

Using the QSF structure of Eq. (3.1) of the rate matrix Q, the (potentially non-zero) elements of the matrices D^m, W^m, $U^{m,k}$ will be denoted respectively by $d(m-1, j \,|\, m, i)$, (a "down" rate), $w(m, j \,|\, m, i)$ (a "within" rate) and $u(k, j \,|\, m, i)$, (an "up" rate) for $k > m$. Note that the diagonal elements of W^m are the negative sum of all other elements in that row of Q.

3.3 Explicit Solutions for DES QSF processes

The following lemmas show that the simple algebraic characterization of the DES property is a sufficient condition for a QSF process to be successively lumpable. Following Chapter 2, a state is an *entrance state* of a subset $\mathcal{X}_0$ of the state space $\mathcal{X}$ if all one step transitions from outside this set $\mathcal{X}_0$ into $\mathcal{X}_0$ can only occur via a transition to the entrance state.

We start with the following Lemma which characterizes an entrance state of a sub-level set L_m of a QSF process in terms of an algebraic property of the "down" transition sub-matrix D^m of its transition rate matrix Q.

Lemma 3.1. *For a QSF process $X(t)$ and for a fixed $m \in \{M_1, \ldots, M_2\}$, the state*

$$(m, \boldsymbol{\varepsilon}(L_m)) \in L_m,$$

is an **entrance state** *for $\underset{\sim}{L}_m$ if and only if the following is true for all $(m+1, i) \in L_{m+1}$:*

$$d(m, j \,|\, m+1, i) = 0, \;\; \textit{if } (m, j) \neq (m, \boldsymbol{\varepsilon}(L_m)). \tag{3.2}$$

Proof. The structure of the rate matrix Q implies that "down" transitions leaving the set $\widetilde{L}_{m+1} = L_{m+1} \cup L_{m+2} \cup \ldots$ can only come from states in L_{m+1}. Further, by Eq. (3.2) the latter type of transitions are possible only when they lead into the same state $(m, \boldsymbol{\varepsilon}(L_m)) \in L_m$. □

It is easy to see that Eq. (3.2) of Lemma 3.1 is equivalent to the statement that the $\ell_m \times \ell_{m-1}$ matrix D^m has a single non-zero column.

For any fixed $n \in \{M_1, \ldots, M_2\}$, let $\mathcal{D}_n$ denote the partition $\{\underset{\sim}{L}_n, L_{n+1}, \ldots, L_{M_2}\}$ of $\mathcal{X}$. For a fixed n, the next lemma establishes that when D^m has a single non-zero column for all $m \geq n+1$, i.e., Q has the DES property, then the QSF process is successively lumpable with respect to the partition $\mathcal{D}_n$.

Lemma 3.2. *A QSF process is successively lumpable with respect to a partition $\mathcal{D}_{M_1}$ if D^m contains a single non-zero column vector for all $m = M_1 + 1, \ldots, M_2$.*

Proof. It is a direct consequence of Lemma 3.1, that for a QSF process, a sub-level set $\underset{\sim}{L}_m$ has an entrance state $(m, \boldsymbol{\varepsilon}(L_m))$ if D^{m+1} contains a single non-zero column vector. This is true for all $m \geq M_1$. Since $\underset{\sim}{L}_m = \underset{\sim}{L}_{m-1} \cup L_m$ it follows from the definition that the process is successively lumpable: in the notation of Chapter 2, "D_0" corresponds to $\underset{\sim}{L}_{M_1}$ and for $m > M_1$: "D_m" corresponds to L_{m-M_1}.

□

Note that when a QSF process is successively lumpable with respect to a partition $\mathcal{D}_n$ then it is successively lumpable with respect to partition $\mathcal{D}_m$ for all $m > M_1$. A graphical representation of the transitions that are allowed in a successively lumpable QSF process can be found in Figure 3.1.

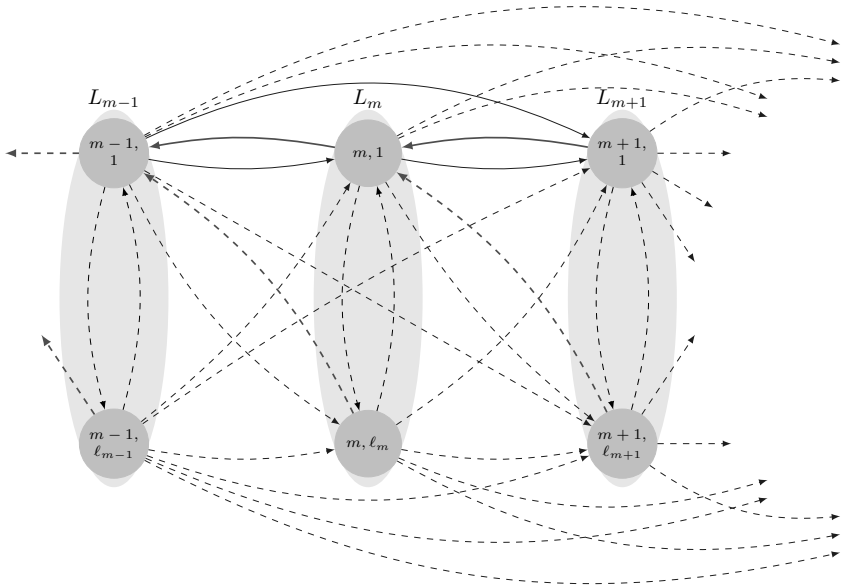


Figure 3.1: Graphical representation of a DES QSF process

We now state the following assumption that will be used in the sequel, where for notational simplicity we let the entrance state of a set $\underset{\sim}{L}_m$ be state $(m, 1)$, for all m without loss of generality.

Assumption 3.1. In this section the QSF process has a transition rate matrix Q with the following properties:

A1. The QSF process is ergodic (irreducible).

A2. For all $m \in \{M_1+1, \ldots, M_2\}$, only the *first column* of sub-matrix D^m contains non-zero elements, i.e., $d(m-1, 1 \,|\, m, i) > 0$ for at least one $(m, i) \in L_m$ and all other columns of D^m are equal to zero.

A3. The QSF process has bounded rates.

Remark 3.2. Part (A3) of Assumption 3.1 is given to make the proofs applicable for an infinite state space and is not strictly necessary. It can be relaxed without further conditions; it has been added to the assumption to make some of the proofs easier to state.

We will use the following notation. We first define the scalar $\underset{\sim}{\ell}_m := \sum_{k=M_1}^{m} \ell_k$ and use the symbols I_m and $\underset{\sim}{I}_m$ for the identity matrices of dimension $\ell_m \times \ell_m$ and $\underset{\sim}{\ell}_m \times \underset{\sim}{\ell}_m$, respectively. Second we define the (row)vectors of dimension ℓ_m, 0_m, $\mathbf{1}_m$ and δ_m to represent a vector identically equal to 0, a vector identically equal to 1, a vector with 1 as its first coordinate and 0 elsewhere respectively. Next we define the (row)vectors of dimension $\underset{\sim}{\ell}_m$: $\underset{\sim}{0}_m$ and $\underset{\sim}{\mathbf{1}}_m$ to be the vectors will all its coordinates equal to 0 and 1, respectively. Finally, we define the matrix $\widetilde{U}^{\,m,n}$ of dimension $\ell_m \times \ell_m$ by:

$$\widetilde{U}^{\,m,n} = \sum_{k=n}^{M_2} U^{m,k} \mathbf{1}_k' \delta_m, \tag{3.3}$$

where the elements of $\widetilde{U}^{\,m,n}$ will be denoted by $\tilde{u}(m,i)$, thus,

$$\tilde{u}(m,i) := \sum_{k=n}^{M_2} \sum_{j=1}^{\ell_k} u(k, j \,|\, m, j).$$

We next define the rate sub-matrices:

$$A_m = \begin{bmatrix} \widetilde{U}^{\,M_1, m+1} + U^{M_1, m} \\ \vdots \\ \widetilde{U}^{\,m-1, m+1} + U^{m-1, m} \end{bmatrix}, \tag{3.4}$$

$$B_m = \widetilde{U}^{\,m, m+1} + W^m. \tag{3.5}$$

and

$$\Gamma_m := \begin{bmatrix} A_m \\ B_m \end{bmatrix}.$$

Note that:

i) The matrix A_m contains all rates of Q corresponding to transitions from states in $\underset{\sim}{L}_{m-1}$, *into* states of L_m plus rates corresponding to transitions under Q from states in $\underset{\sim}{L}_{m-1}$ into states of $\widetilde{L}_{m+1}$, which under A_m have been re-directed to transitions into the entrance state $(m, 1)$.

ii) The $\ell_m \times \ell_m$ matrix B_m contains all rates of Q corresponding to transitions from states in L_m, *into* states of L_m plus rates corresponding to transitions under Q from states in L_m into states of $\widetilde{L}_{m+1}$, which under B_m have been re-directed to transitions into the entrance state $(m, 1)$. Thus, for $m > M_1$ since the construction of B_m excludes all down transitions it a transient transition rate matrix. However, by the definition of $\widetilde{U}^{M_1, M_1+1}$ and by its construction the matrix B_{M_1} is an $\ell_{M_1} \times \ell_{M_1}$ conservative transition rate matrix.

We next state and prove a proposition regarding basic properties of B_m.

Proposition 3.1. *The following are true:*
i) The matrix B_{M_1} is irreducible.
ii) The matrices B_m are non-singular, for all $m > M_1$.
iii) All elements of the inverse of B_m are non-positive, for all $m > M_1$.

Proof. To prove i), we will show that every state (M_1, j) in level L_{M_1} that is not the entrance state $(M_1, 1)$ communicates with state $(M_1, 1)$. First, recall that in the construction of B_{M_1}, "up"-transitions are redirected to the entrance state. If there are no "up"-transitions from the communicating class containing (M_1, j), this class would be a closed class in Q which would not contain state $(M_1, 1)$, a contradiction to the irreducibility assumption of Q. So the entrance state is reachable from (M_1, j) under B_{M_1}. Second, we show that (M_1, j) is reachable from state $(M_1, 1)$ under B_{M_1}. Indeed, the only way to reach (M_1, j) from a state in $\widetilde{L}_{M_1+1}$ is via the entrance state, and such a path has to exist by irreducibility of Q. Since the "within" L_{M_1} transition rates under Q are all preserved under B_{M_1}, state (M_1, j) is reachable from state $(M_1, 1)$ under B_{M_1}. Thus every state communicates with the entrance state, and we conclude that B_{M_1} is irreducible.

For ii), we will call a matrix *diagonally dominant* if the absolute value of a diagonal elements are greater or equal than the sum of the absolute values of the off diagonal elements in that row, and strict inequality holds for at least one row. An irreducible diagonally dominant matrix is non-singular by the well-known Levy-Desplanques theorem, cf. [109] (p. 85) or [110].

The construction of B_m excludes all down transitions and that makes it a diagonally dominant matrix. So when B_m is irreducible, the claim of the lemma is true.

However, in general it is possible that the matrix B_m is not irreducible (even though the matrix Q is irreducible). In this case we will show that the construction of B_m with the irreducibility of Q implies the non-singular property of B_m. Indeed, suppose B_m contains two

or more communicating classes of states, say $C_1, \ldots, C_{k_m}$, where C_e contains the entrance state $(m, 1)$. We will relabel the states such that states in the same class have adjacent indices and such that if a state in a class C_i has a transition to a state in a class C_j then $i < j$. It is clear that this relabeling is feasible and we can write B_m as:

$$B_m = \begin{bmatrix} Z_{11} & Z_{12} & Z_{13} & \ldots \\ 0 & Z_{22} & Z_{23} & \ldots \\ 0 & 0 & Z_{33} & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

where Z_{ii} is a matrix of size $|C_i| \times |C_i|$ containing transition rates within C_i and the matrix Z_{ij} is of size $|C_i| \times |C_j|$ containing transition rates from C_i to C_j (and is possible to have some non-zero elements). We next show that the determinants of Z_{ii} are all non-zero. This is sufficient to show that B_m is non-singular, since its determinant is the product of the determinants of the Z_{ii}'s.

First, at least one state in C_e has to have a transition "down" or to another class C_j within B_m, since otherwise C_e would be part of a closed absorbing class under Q. This class does not contain any states in $\underset{\sim}{L}_m$ and this is a contradiction to the irreducibility of Q. Thus Z_{ee} is diagonally dominant and therefore non-singular. Furthermore, any other class C_j $(j \neq e)$ has to have a state that has a transition rate leaving this class (i.e., down, up or to another class $C_{j'}$ $(j' \neq j)$) otherwise C_j would be a closed class under Q. Therefore, the sum of the off diagonal elements of Z_{jj} in at least one of its row is strictly less than the absolute value of the corresponding diagonal element, since at least one transition leads to a state out of C_j. Thus Z_{jj} is diagonally dominant for all j and the proof of part ii) is complete. For iii), let τ_i denote the maximum of the absolute values of the diagonal elements of Z_{ii} and let

$$\Gamma_i = \tau_i I + Z_{ii}.$$

Since τ_i is finite and positive and Γ_i is non-negative with row sum less or equal to τ_i, (and strictly less than τ_i for at least one row), the row sum of $\tau_i^{-1}\Gamma_i$ is smaller or equal to one for each row. This implies that $\tau_i^{-1}\Gamma_i$ is a transient transition matrix with spectral radius smaller than 1, and thus all elements of $\sum_{j=0}^{\infty}(\tau_i^{-1}\Gamma_i)^j$ are non-negative and finite, see for example [95], Theorem 4.3. Note also that

$$(Z_{ii})^{-1} = (\tau_i(\tau_i^{-1}\Gamma_i - I))^{-1} = \tau_i^{-1}(\tau_i^{-1}\Gamma_i - I)^{-1} = \tau_i^{-1}\sum_{j=0}^{\infty} -(\tau_i^{-1}\Gamma_i)^j.$$

The above implies that all elements of $(Z_{ii})^{-1}$ are non-positive.

By Woodbury's identity, cf. [115] we know:

$$\begin{bmatrix} Z_{11} & Z_{12} \\ 0 & Z_{22} \end{bmatrix}^{-1} = \begin{bmatrix} Z_{11}^{-1} & -Z_{11}^{-1}Z_{12}Z_{22}^{-1} \\ 0 & Z_{22}^{-1} \end{bmatrix},$$

which is a non-positive matrix by the above; an induction argument can establish the same for $(B_m)^{-1}$. □

We emphasize the following notational change with Chapter 2.

Remark 3.3. Let $\Delta_0 = \underset{\sim}{L}_m = \{(M_1,1),\ldots,(M_1,\ell_{M_1}),\ldots,(m,1),\ldots,(m,\ell_m)\}$. The lumped process on Δ_0 has a rate matrix "U_{Δ_0}" (defined in Chapter 2) of size $\underset{\sim}{\ell}_m \times \underset{\sim}{\ell}_m$ that can be written as:

$$[\Lambda_m \,|\, \Gamma_m]$$

where Λ_m contains the rates of transitions into states of the set $\underset{\sim}{L}_{m-1}$ (i.e., it is a matrix of dimension $\underset{\sim}{\ell}_m \times \underset{\sim}{\ell}_{m-1}$ and the construction of the $\underset{\sim}{\ell}_m \times \ell_m$, matrix Γ_m is done following Chapter 2). Note that we do not need to explicitly define the elements of the matrices Λ_m as they are not explicitly used in the sequel.

We next state the following theorem for successively lumpable Markov processes within the context and notation of the present chapter. This theorem is a consequence of Theorem 2.2, as is shown in the proof.

Theorem 3.1. *Under Assumption 3.1, the following equality is true for the steady state probabilities $\underset{\sim}{\pi}^m$ of $X(t)$ for every m :*

$$\underset{\sim}{\pi}^m \Gamma_m = 0_m.$$

Proof. For any fixed m, we consider the partition $\mathcal{D} = \{\underset{\sim}{L}_m, L_{m+1},\ldots,L_{M_2}\}$ of the state space $\mathcal{X}$ of the process. We note that the sets Δ_m of Chapter 2 are, within the present context, given by: $\Delta_k = \{(k\,0)\} \cup L_k$; where $(k\,0)$ represents the "lumped state".

By Lemma 3.2 we know that $X(t)$ is successively lumpable with respect to $\mathcal{D}$. Let $v_{\underset{\sim}{L}_m}$ denote the steady state probability vector of the lumped process on $\Delta_0 = \underset{\sim}{L}_m$, cf. Remark 3.3. By Proposition 2.1 (with $(k,i) \in \underset{\sim}{L}_m$ in place of $(0,i) \in \Delta_0$) we know that for all $k \le m$:

$$\pi(k,i) = \sum_{(k',j)\in \underset{\sim}{L}_m} \pi(k'\,j) v_{\underset{\sim}{L}_m}(k,i). \tag{3.6}$$

Further, since $v_{\underset{\sim}{L}_m}$ is a steady state probability vector of the lumped process on $\underset{\sim}{L}_m$(="Δ_0"), it is the normalized to 1 solution of the equation below (see Remark 3.3):

$$v_{\underset{\sim}{L}_m}[\Lambda_{m-1} \,|\, \Gamma_m] = \underset{\sim}{0}_m. \tag{3.7}$$

If we let $c = \sum_{(k'\,j)\in \underset{\sim}{L}_m} \pi(k',j) \ge 0$, then from Eq. (3.6) we know $\underset{\sim}{\pi}^m = c\, v_{\underset{\sim}{L}_m}$. Eq. (3.8) below then follows by multiplying both sides of Eq. (3.7) by c:

$$\underset{\sim}{\pi}^m[\Lambda_{m-1} \,|\, \Gamma_m] = [\,\underset{\sim}{\pi}^m \Lambda_{m-1} \,|\, \underset{\sim}{\pi}^m \Gamma_m] = \underset{\sim}{0}_m = [\,\underset{\sim}{0}_{m-1} \,|\, 0_m], \tag{3.8}$$

and the proof is complete. □

Note that Theorem 3.1 implies, since $\Gamma_{M_1} = B_{M_1}$ and since $\underset{\sim}{\pi}^{M_1-1}$ does not exist:

$$\pi^{M_1} B_{M_1} = 0_{M_1}.$$

We next introduce the idea of a *rate matrix set* for Q as a sequence of matrices $\mathcal{R} = \{\mathcal{R}_m^k\}_{m,k}$ such that $\mathcal{R}_m^k$ satisfy Eq. (3.9), for all $k = 1, \ldots, m - M_1$ and $m = M_1 + 1, \ldots, M_2$; cf. [70] and references therein.

$$\pi^m = \underset{\sim}{\pi}^{m-k} \mathcal{R}_m^k. \tag{3.9}$$

Note that there are multiple rate matrix sets, for a given Q. To see this note that for any fixed k, m and known vectors $\pi^m = [\pi(m,1), \ldots, \pi(m,\ell_m)]$ and $\underset{\sim}{\pi}^{m-k} = [\pi^{M_1}, \ldots, \pi^{m-k}]$ Eq. (3.9) is essentially a system of ℓ_m equations with $\underset{\sim}{\ell}_{m-k} \times \ell_m$ unknowns, the elements of the matrix $\mathcal{R}_m^k$. These equations have many solutions.

In Theorem 3.2 we show that the specific set $\mathcal{R}_0 := \{R_m^k\}_{m,k}$ obtained recursively using Eq. (3.10) starting with Eq. (3.11), is a rate matrix set for Q. For all $m = M_1 + 1, \ldots M_2$ with $k = 2, \ldots m - M_1$ we define:

$$R_m^k := \left[\underset{\sim}{I}_{m-k} \mid R_{m-(k-1)}^1\right] R_m^{k-1}, \tag{3.10}$$

where:

$$R_m^1 := -A_m (B_m)^{-1}. \tag{3.11}$$

By virtue of Proposition 3.1 ii) we know that B_m is non-singular.

Theorem 3.2. *The set $\mathcal{R}_0$ defined by Eq. (3.10) and (3.11) above is a rate matrix set for Q.*

Proof. The proof is by induction. For $k = 1$ we know by Theorem 3.1 that $\underset{\sim}{\pi}^m \Gamma_m = 0_m$. We can rewrite this as:

$$[\underset{\sim}{\pi}^{m-1} | \pi^m] \begin{bmatrix} A_m \\ B_m \end{bmatrix} = 0_m,$$

and thus:

$$\pi^m = -\underset{\sim}{\pi}^{m-1} A_m (B_m)^{-1} = \pi^{m-1} R_m^1.$$

Suppose the statement is true for any m and for $k-1$. We next show that the statement holds for k:

$$\begin{aligned}
\pi^m &= \underset{\sim}{\pi}^{m-(k-1)} R_m^{k-1} \\
&= [\underset{\sim}{\pi}^{m-k} | \pi^{m-(k-1)}] R_m^{k-1} \\
&= [\underset{\sim}{\pi}^{m-k} | \underset{\sim}{\pi}^{m-k} R_{m-(k-1)}^1] R_m^{k-1} \\
&= \underset{\sim}{\pi}^{m-k} \left[\underset{\sim}{I}_{m-k} | R_{m-(k-1)}^1\right] R_m^{k-1} = \underset{\sim}{\pi}^{m-k} R_m^k.
\end{aligned}$$

Thus the statement is true for $k = 1, \ldots, m - M_1$ and therefore:

$$\pi^m = \underset{\sim}{\pi}^{m-k} R_m^k. \qquad \square$$

Note that the above implies that we can express all vectors π^m in terms of the steady state distribution of level M_1, since M_1 is finite, By the irreducibility assumption all vectors are strictly larger than 0. Therefore we state:

$$\pi^m = \pi^{M_1} R_m^{m-M_1} > 0_m, \tag{3.12}$$

For any $m_1, m_2 \in \{M_1, \ldots, M_2\}$, with $m_1 < m_2$, we define the column vector $S_{m_1}^{m_2}$ of length ℓ_{m_1} by Eq. (3.13) below.

$$S_{m_1}^{m_2} = \left[\mathbf{1}'_{m_1} + \sum_{m=m_1+1}^{m_2} R_m^{m-m_1} \mathbf{1}'_m \right]. \tag{3.13}$$

Remark 3.4. The elements of R_m^k are non-negative $\forall\ k, m$ where $M_1 + 1 \leq m \leq M_2$, $1 \leq k \leq m - M_1$. To check this claim for R_m^1, it suffices to note that $A_m \geq 0$ by definition and $-B_m^{-1} \geq 0$, by Proposition 3.1. Alternatively, the (i, j)-th element of R_m^1 can be given an expected first passage time interpretation as is described for QBD processes in [70], Chapter 6. The claim for R_m^k, with $k \geq 2$, follows using Eq. (3.10).

The lemma below establishes the relation between π^{M_1} and $S_{M_1}^{M_2}$.

Lemma 3.3. *The following relation holds for π^{M_1} and $S_{M_1}^{M_2}$:*

$$\pi^{M_1} S_{M_1}^{M_2} = 1. \tag{3.14}$$

Proof. Since the process is ergodic we have:

$$\pi^{M_1} \mathbf{1}'_{M_1} + \sum_{m=M_1+1}^{M_2} \pi^m \mathbf{1}'_m = 1,$$

thus Eq. (3.12) implies:

$$\pi^{M_1} \left[\mathbf{1}'_{M_1} + \sum_{m=M_1+1}^{M_2} R_m^{m-M_1} \mathbf{1}'_m \right] = 1.$$

Substituting $\left[\mathbf{1}'_{M_1} + \sum_{m=M_1+1}^{M_2} R_m^{m-M_1} \mathbf{1}'_m \right]$ by $S_{M_1}^{M_2}$ in the above gives Eq. (3.14).

$\square$

We now state and prove the following theorem.

Theorem 3.3. *Under Assumption 3.1, the following is true:*

$$\pi^{M_1} = \delta_{M_1} \left[S_{M_1}^{M_2} \delta_{M_1} - B_{M_1} \right]^{-1}. \tag{3.15}$$

Proof. Since B_{M_1} is an irreducible rate matrix (Proposition 3.1 i)), it has rank $(\ell_{M_1} - 1)$ by basic linear algebra theory, see for example [95]. Furthermore, we know that $\pi^{M_1} B_{M_1} = 0_{M_1}$ and $\pi^{M_1} S_{M_1}^{M_2} = 1$, thus that the vector $S_{M_1}^{M_2}$ is not an element of the linear space spanned by the columns of B_{M_1}. Therefore $[S_{M_1}^{M_2} \delta_{M_1} - B_{M_1}]$ has full rank and is invertible.

We use Lemma 3.3 to state

$$\left[\pi^{M_1} S_{M_1}^{M_2} \right] \delta_{M_1} = \delta_{M_1},$$

and via

$$\pi^{M_1} \left[S_{M_1}^{M_2} \delta_{M_1} - B_{M_1} \right] = \delta_{M_1} - 0_{M_1} = \delta_{M_1},$$

we conclude

$$\pi^{M_1} = \delta_{M_1} \left[S_{M_1}^{M_2} \delta_{M_1} - B_{M_1} \right]^{-1}.$$

□

The results above justify the following algorithm to find the steady state distribution of a DES-QSF process.

Algorithm 3.1. *DES-QSF*

- Calculate R_m^1 with Eq. (3.11) for all $m = M_1 + 1, \ldots, M_2$.
- Compute R_m^k recursively via Eq. (3.10) for $m = M_1 + 1 \ldots, M_2$ and $k = 2, \ldots, m - M_1$.
- Calculate $S_{M_1}^{M_2}$ via Eq. (3.13).
- Calculate π^{M_1} via Eq. (3.15).
- Calculate π^m via Eq. (3.12) for all $m = M_1 + 1, \ldots, M_2$.

3.3.1 State space truncations

In this section we show how to truncate the state space in the upward direction, in order to obtain upper bounds for the steady state probabilities $\pi(m, i)$ of states in $\underset{\sim}{L}_{m_2}$ where

$m_2 \in \{M_1, M_1+1, \ldots, M_2-1\}$. To this end we first define a process $X_{m_2}(t)$ with truncated state space $\mathcal{X}_{m_2} = \underset{\sim}{L}_{m_2}$ and transition rate matrix:

$$Q_{X_{m_2}} = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots \\ \cdots & D^{m_2-2} & W^{m_2-2} & U^{m_2-2,m_2-1} & U^{m_2-2,m_2} + \widetilde{U}^{m_2-2,m_2} \\ \cdots & 0 & D^{m_2-1} & W^{m_2-1} & U^{m_2-1,m_2} + \widetilde{U}^{m_2-1,m_2} \\ \cdots & 0 & 0 & D^{m_2} & W^{m_2} + \widetilde{U}^{m_2,m_2} \end{bmatrix},$$

where the elements of the last column are given by Eq. (3.3). We denote the steady state distribution of this process as the row vector $\pi_{X_{m_2}} = [\pi^{M_1}_{X_{m_2}}, \ldots, \pi^{m_2}_{X_{m_2}}]$ of size: $\sum_{m=M_1}^{m_2} \ell_m$, where its m^{th} component contains the steady state probabilities for level m of the truncated process.

We next state the following. We emphasize that this proposition clearly holds for $M_2 = \infty$ under the ergodicity assumption.

Proposition 3.2. *For all finite $m_2 \geq M_1$, and any level $m = M_1, M_1+1, \ldots, m_2$, the following are true:*

$$i) \qquad \pi^m_{X_{m_2}} = \pi^{M_1}_{X_{m_2}} R_m^{m-M_1}, \tag{3.16}$$

$$\pi^{M_1}_{X_{m_2}} = \delta_{M_1} \left[S^{m_2}_{M_1} \delta_{M_1} - B_{M_1} \right]^{-1}. \tag{3.17}$$

$$ii) \qquad \pi(m,i) < \pi_{X_{m_2}}(m,i).$$

iii) For all states (m,i), $\pi_{X_\nu}(m,i)$ is a strict decreasing function in $\nu = m_2, m_2+1, \ldots$

Proof. By its construction, the process $X_{m_2}(t)$ is a QSF process which satisfies Assumption $A1$. Further, by its definition the matrix $Q_{X_{m_2}}$ gives rise to the same rate matrices R^k_m as the matrix Q of the original process $X(t)$. This follows from the fact that this specific truncation ensures that the matrices $A_{m,X_{m_2}}$, $B_{m,X_{m_2}}$ of the truncated process corresponding to the matrices A_m B_m of the original process are identical and this proves Eq. (3.16). The proof of Eq. (3.17) follows as the proof of Theorem 3.3, if we replace M_2 with m_2.

For the proof of part ii), using Proposition 2.1, (where $\pi_{X_{m_2}}(m,i) = v_{\underset{\sim}{L}_m}(m,i)$) we obtain that Eq. (3.18) below is valid for all $m \leq \nu$:

$$\pi_{X_\nu}(m,i) = \frac{\pi(m,i)}{\sum_{(k,j)\in \underset{\sim}{L}_\nu} \pi(k,j)} \; for\ all\ \nu = m_2, m_2+1, \ldots. \tag{3.18}$$

Since $\sum_{(k,j)\in \underset{\sim}{L}_\nu} \pi(k,j) < 1$, for all finite ν, it follows from the above that $\pi(m,i) < \pi_{X_{m_2}}(m,i)$.

For the proof of part iii), note that since $\underset{\sim}{L}_\nu \subset \underset{\sim}{L}_{\nu+1}$, we have that $\sum_{(k,j)\in \underset{\sim}{L}_\nu} \pi(k,j) <$

$\sum_{(k,j)\in \underset{\sim}{L}_{\nu+1}} \pi(k,j)$. Thus, we conclude that $\pi_{X_{\nu+1}}(m,i) < \pi_{X_\nu}(m,i)$. We can repeat this argument for $\underset{\sim}{L}_{\nu+2}, \underset{\sim}{L}_{\nu+3}, \ldots$ and the proof is complete.

□

Note that Proposition 3.2 is closely related the results of [33] (pp. 499-500), derived for LDQBDs. Specifically, Eq. (3.16)-(3.17) are the QSF process extensions of Eq. (1.7)-(1.8) of that paper, with k and m reversed and the change of notation K^*, x_k, R_{k+1}, R_0 in place of our m_2, π^m, R_m R_1.

Note that the matrix $R_m^{m-M_1}$ is finite even when the the QSF process is not ergodic; such a non-ergodic case exists for instance when there is a drift to "up" direction. We can however state the following:

Remark 3.5. The successively lumpable QSF process is ergodic if $\sum_{m=M_1}^{M_2} R_m^{m-M_1} < \infty$, since then Theorem 3.3 and Eq. (3.12) show that there exist positive steady state probabilities for all states. Similarly, it follows that for QSF processes to be ergodic it is sufficient that $S_{M_1}^{M_2} < \infty$.

Remark 3.6. One can also construct truncations with respect to M_1 or to any ℓ_m separately. This is especially important when some of these constants are infinite. There are various truncations methods possible to truncate the matrix Q of infinite size, some are described in [112], [94] and [102]. Most of these truncations will preserve the successively lumpable property. Such as truncation to $m_1 \geq M_1 = -\infty$ we provide below, following [94].

Define a process $X_{m_1}(t)$ with state space $\mathcal{X}_{m_1} = \widetilde{L}_{m_1}$ and transition rate matrix:

$$Q_{X_{m_1}} = \begin{bmatrix} \overline{W}^{m_1} & U^{m_1,m_1+1} & U^{m_1,m_1+2} & U^{m_1,m_1+3} & \cdots \\ D^{m_1+1} & W^{m_1+1} & U^{m_1+1,m_1+2} & U^{m_1+1,m+3} & \cdots \\ 0 & D^{m_1+2} & W^{m_1+2} & U^{m_1+2,m_1+3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

where $\overline{w}(m_1, j \,|\, m_1, i) = w(m_1, j \,|\, m_1, i)$ if $j \neq 1$ and $\overline{w}(m_1, 1 \,|\, m_1, i) = w(m_1, 1 \,|\, m_1, i)$ $+d(m_1 - 1, 1 \,|\, m_1, i)$ otherwise. Let $\pi_{X_{m_1}}$ denote the steady state distribution of $X_{m_1}(t)$. For this truncation it is shown in [94] (Theorem on p. 262) that for all (m, i):

$$\pi(m,i) = \lim_{m_1 \to -\infty} \pi_{X_{m_1}}(m,i).$$

3.4 Explicit solutions for RES QSF processes

In this section we consider QSF processes of the form described in Definition 3.1 ii). Because the current section has the same structure as Section 3.3, most of the theorems and lemmas

can be given without proof. We will refer to the corresponding statement in the previous section and add some clarification if necessary.

We start with the following lemmas that show that the simple algebraic characterization of the RES property is a sufficient condition for a QSF process to be successively lumpable.

We start with the following lemma that characterizes an entrance state of the super-level set $\widetilde{L}_m$ of a QSF process in terms of an algebraic property of the "up" transition sub-matrices $U^{m,k}$ of its transition rate matrix Q.

Lemma 3.4. *For a QSF process $X(t)$ and for all $m \in \{M_1, \ldots, M_2\}$, the state*

$$(M_2, \boldsymbol{\varepsilon}(L_m)) \in L_{M_2},$$

is an **entrance state** *for $\widetilde{L}_m$ if the following is true for all states $(n, i) \in \underline{L}_{m-1}$:*

$$u(k, j \,|\, n, i) = 0, \;\; \textit{if } (k, j) \neq (M_2, \boldsymbol{\varepsilon}(L_m)). \tag{3.19}$$

Proof. Directly from the definition of an entrance state (cf. Definition 2.1). □

It is easy to see that Eq. (3.19) of Lemma 3.4 is equivalent to the statement that the matrices U^{m,M_2} have a single non-zero column and that $U^{m,k} = 0$ for all $k \in \{m+1, \ldots, M_2 - 1\}$. This is equivalent to the RES property.

For any fixed $n \in \{M_1, \ldots, M_2\}$, let $\mathcal{D}_n$ denote the partition $\{L_{M_1}, \ldots, L_{n-1}, \widetilde{L}_n\}$ of $\mathcal{X}$. For a fixed n, the next lemma establishes that when Q has the RES property (cf. 3.1 ii)), then the QSF process is successively lumpable with respect to the partition $\mathcal{D}_n$.

Lemma 3.5. *A QSF process is successively lumpable with respect to a partition $\mathcal{D}_{M_2}$ if the matrices U^{m,M_2} have a single non-zero column and that $U^{m,k} = 0$ for all $k \in \{m + 1, \ldots, M_2 - 1\}$.*

Proof. Similar to the proof of Lemma 3.2 in the previous section. □

A graphical representation of the transitions that are allowed in a RES QSF process can be found in Figure 3.2.

We now state the following assumption that will be used in the sequel of this section, where for notational simplicity we let the entrance state of a set $\widetilde{L}_m$ be state $(M_2, 1)$, for all m without loss of generality.

Assumption 3.2. The QSF process under consideration has a transition rate matrix Q with the following properties:

A1. The QSF process is ergodic (irreducible);

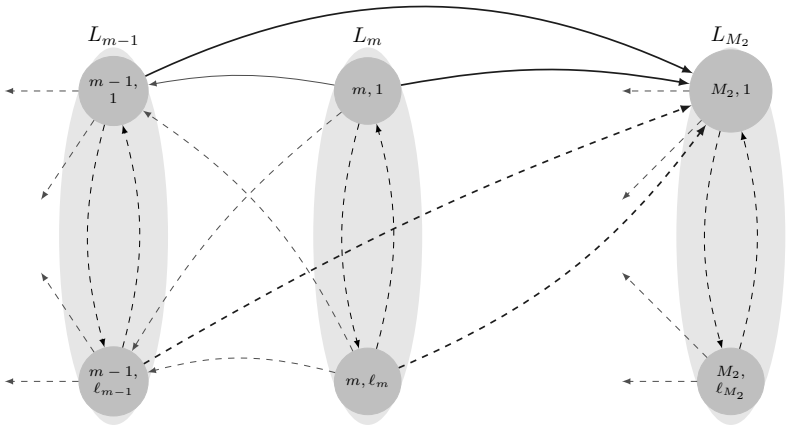


Figure 3.2: Graphical representation of a RES QSF process

A2. For all $m \in \{M_1, \ldots, M_2 - 1\}$, only the *first column* of sub-matrix U^{m,M_2} can contain non-zero elements. Since the process is positive recurrent, there has to be a transition back to the entrance state from a state in $\underset{\sim}{L}_m$, i.e. $u(M_2, 1 \,|\, n, i) > 0$ for at least one $(n, i) \in \underset{\sim}{L}_m$ and all other columns of U^{m,M_2} are equal to zero. In addition, $U^{m,k} = 0$ for all $k \in \{m, \ldots, M_2 - 1;$

A3. The QSF process has bounded rates.

We will use the notation introduced in the previous section. In addition we define the matrix $\widetilde{D}^m$ of dimension $\ell_m \times \ell_m$ by:

$$\widetilde{D}^m = D^m \mathbf{1}'_k \delta_m.$$

We next state and prove a proposition regarding basic properties of W_m and $\widetilde{D}^{M_2}$.

Proposition 3.3. *The following are true:*
i) The matrix $W^{M_2} + \widetilde{D}^{M_2}$ is irreducible.
ii) The matrices W^m are non-singular, for all $m \in \{M_1, \ldots, M_2\}$.
iii) The elements of the inverse of W^m are non-positive, for all $m < M_2$.

Proof. The proves are analogous to the one of Proposition 3.1 in the previous section, that states similar results for a DES QSF process. □

We can now state the following theorem for successively lumpable Markov processes, that satisfy the RES property, within the context and notation of the present section.

Theorem 3.4. *Under Assumption 3.2, the following equality is true for the steady state probabilities π^m of $X(t)$ for every $m \in \{M_1, \ldots, M_2 - 1\}$:*

$$\pi^m W^m + \pi^{m+1} D^{m+1} = 0_m.$$

$$\pi^{M_2}(W^{M_2} + \widetilde{D}^{M_2}) = 0_{M_2}.$$

Proof. The proof is along the same lines as Theorem 3.1 and is a consequence of Theorem 2.2, since the QSF process is successively lumpable. We can complete the proof by only considering the states that have possibly positive transitions out of set $\widetilde{L}_{m+1}$. □

For a RES QSF process, we define a *rate matrix set* for Q as a sequence of matrices $\mathcal{R} = \{\mathcal{R}_m\}_m$ such that $\mathcal{R}_m$ satisfy Eq. (3.20), for all $m = M_1, \ldots, M_2 - 1$. Note that this is a slightly different definition than the ones introduced for DES QSF processes.

$$\pi^m = \pi^{m+1} \mathcal{R}_m. \tag{3.20}$$

In Theorem 3.5 we show that the specific set $\mathcal{R}_0 := \{R_m\}_m$ obtained Eq. (3.21), is a rate matrix set for Q. For all $m = M_1, \ldots, M_2 - 1$ we define:

$$R_m := -D^{m+1}(W^m)^{-1}. \tag{3.21}$$

By virtue of Proposition 3.3 ii) we know that W^m is non-singular.

Theorem 3.5. *The set $\mathcal{R}_0$ defined by (3.21) above is a rate matrix set for Q.*

Proof. Follows directly from Theorem 3.4. □

Note that the above implies that we can express all vectors π^m in terms of the steady state distribution of level M_2, since M_2 is finite. By the irreducibility assumption all vectors are strictly larger than 0. Therefore we state:

$$\pi^m = \pi^{M_2} \prod_{k=0}^{M_2-1-m} R_{M_2-1-k} > 0_m, \tag{3.22}$$

For any $m_1 \in \{M_1, \ldots, M_2\}$, with $m_1 < M_2$, we define the column vector $T_{m_1}^{M_2}$ of length ℓ_{M_2} by Eq. (3.23) below.

$$T_{m_1}^{M_2} = \left[\mathbf{1}'_{M_2} + \sum_{m=m_1}^{M_2-1} \prod_{k=0}^{M_2-1-m} R_{M_2-1-k} \mathbf{1}'_m \right]. \tag{3.23}$$

Note that R_m is non-negative for all m.

The lemma below establishes the relation between π^{M_2} and $T_{M_1}^{M_2}$.

Lemma 3.6. *The following relation holds for π^{M_2} and $T_{M_1}^{M_2}$:*

$$\pi^{M_2} T_{M_1}^{M_2} = 1.$$

Proof. Analogous to Lemma 3.6. □

We now state and prove the following theorem.

Theorem 3.6. *Under Assumption 3.2, the following is true:*

$$\pi^{M_2} = \delta_{M_2} \left[T_{M_1}^{M_2} \delta_{M_2} - W^{M_2} - \widetilde{D}^{M_2} \right]^{-1}. \tag{3.24}$$

Proof. Since $W^{M_2} - \widetilde{D}^{M_2}$ is an irreducible rate matrix (Proposition 3.3 i)), it has rank $(\ell_{M_2} - 1)$ by basic linear algebra theory, see for example [95]. Furthermore, we know that $\pi^{M_2}(W^{M_2} - \widetilde{D}^{M_2}) = 0_{M_2}$ and $\pi^{M_2} T_{M_1}^{M_2} = 1$, thus that the vector $T_{M_1}^{M_2}$ is not an element of the linear space spanned by the columns of $W^{M_2} - \widetilde{D}^{M_2}$. Therefore $[T_{M_1}^{M_2} \delta_{M_1} - W^{M_2} - \widetilde{D}^{M_2}]$ has full rank and is invertible. The remainder of the proof is similar to the proof of Theorem 3.3. □

The results above justify the following algorithm to find the steady state distribution of a RES QSF process.

Algorithm 3.2. *RES-QSF*

- Compute R_m via Eq. (3.21) for $m = M_1 \ldots, M_2 - 1$.
- Calculate $T_{M_1}^{M_2}$ via Eq. (3.23).
- Calculate π^{M_2} via Eq. (3.24).
- Calculate π^m via Eq. (3.22) for all $m = M_1, \ldots, M_2 - 1$.

3.4.1 State space truncations

In this section we show how to truncate the state space of a RES QSF process in the downward direction in order to obtain upper bounds for the steady state probabilities $\pi(m, i)$ of

states in $\widetilde{L}_{m_1}$ where $m_1 \in \{M_1+1, M_1+2, \ldots, M_2\}$. To this end we define a process $X_{m_1}(t)$ with truncated state space $\mathcal{X}_{m_1} = \widetilde{L}_{m_1}$ and transition rate matrix:

$$Q_{X_{m_1}} = \begin{bmatrix} W^{m_1} & 0 & \cdots & 0 & 0 & 0 & \widetilde{D}^{m_1} + U^{m_1,M_2} \\ D^{m_1+1} & W^{m_1+1} & \cdots & 0 & 0 & 0 & U^{m_1+1,M_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & D^{M_2-2} & W^{M_2-2} & 0 & U^{M_2-2,M_2} \\ 0 & 0 & \cdots & 0 & D^{M_2-1} & W^{M_2-1} & U^{M_2-1,M_2} \\ 0 & 0 & \cdots & 0 & 0 & D^{M_2} & W^{M_2} \end{bmatrix}.$$

We denote the steady state distribution of this process as the row vector:

$$\pi_{X_{m_1}} = [\pi^{m_1}_{X_{m_1}}, \ldots, \pi^{M_2}_{X_{m_1}}].$$

of size: $\sum_{m=m_1}^{M_2} \ell_m$, where its m^{th} component contains the steady state probabilities for level m of the truncated process.

We next state the following. We emphasize that this proposition clearly holds for $M_1 = \infty$ under the ergodicity assumption.

Proposition 3.4. *For all finite $m_1 \leq M_2$, and any level $m = m_1, m_1+1, \ldots, M_2$, the following are true:*

$i)$
$$\pi^{m}_{X_{m_1}} = \pi^{M_2}_{X_{m_1}} \prod_{k=0}^{M_2-1-m} R_{M_2-1-k}$$
$$\pi^{M_2}_{X_{m_1}} = \delta_{M_2} \left[T^{M_2}_{m_1} \delta_{M_2} - W^{M_2} - \widetilde{D}^{M_2} \right]^{-1}.$$

$ii)$
$$\pi(m,i) < \pi_{X_{m_1}}(m,i).$$

iii) For all states (m,i), $\pi_{X_\nu}(m,i)$ is a strict decreasing function in $\nu = m_1, m_1-1, \ldots$

Proof. Similar to the proof of Proposition 3.2: the RES property remains intact, the rate matrices do not change. The entrance state will never be removed from the state space. □

Remark 3.7. It is not useful to truncate the process in the upward direction. Since we consider a ergodic process with a restart entrance state, returns will go to the highest level. Removing this set would effect the structure of the process too much to give any bounds of interest.

3.5 A special case of QSF processes: QBD processes

A Quasi Birth and Death process is a special case of a QSF process, where $U^{m,k} = 0$ for $k \geq m+2$. Therefore we rename in this section $U^{m,m+1}$ to U^m. All the proofs of the

previous section for a QSF process also hold for a QBD process, but the algebra simplifies. In the sequel we will assume that $M_1 = 0$ and that Q is irreducible. The QBD process $X(t)$ has the transition rate matrix of Eq. (3.25) below.

$$Q = \begin{bmatrix} W^{M_1} & U^{M_1} & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ D^{M_1-1} & W^{M_1+1} & U^{M_1+1} & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & W^{m} & U^{m} & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & D^{m+1} & W^{m+1} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & D^{M_2-1} & W^{M_2-1} & U^{M_2-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & D^{M_2} & W^{M_2} \end{bmatrix}. \quad (3.25)$$

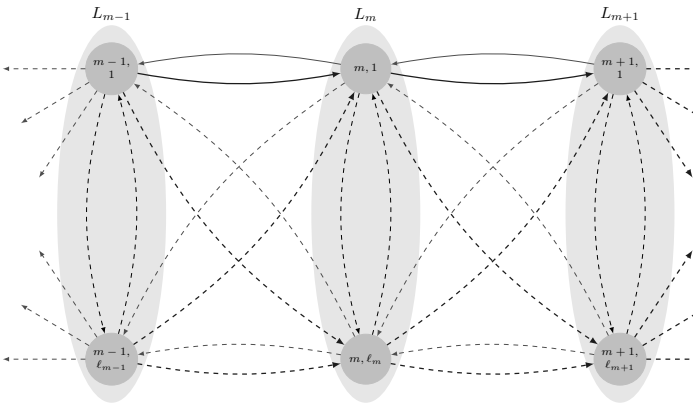


Figure 3.3: Graphical representation of a QBD process

In a QBD process $\widetilde{U}^m$ is as follows:

$$\widetilde{U}^m = U^m \mathbf{1}'_{m-1} \delta_m.$$

In a successively lumpable QBD process, the rate matrix set $\{\mathcal{R}_m\}_m$ satisfies the equation below, since there are only transitions in the upward direction from one level lower.

$$\pi^m = \pi^{m-1} \mathcal{R}_m\,, \quad \text{for } m = 1, \ldots, M_2. \tag{3.26}$$

Remark 3.8. Note that Eq. (3.26) is Eq. (1.5) in [33] (with k and m reversed and the change of notation their x_k denotes π^m and R_{k+1} denotes our $\mathcal{R}_m$. To find the rate matrix set they provided iterative algorithms. In contrast, Eq. (3.27) below provides explicit formulas for the

rate matrix set $\{\mathcal{R}_m\}_m$; this equation is made possible by the assumption of the successively lumpable property.

Again, let $\mathcal{R}_0 := \{R_m\}_m$ be as follows (the QBD equivalent of Eq. (3.10)):

$$R_m = -U^{m-1}(\widetilde{U}^m + W^m)^{-1}. \tag{3.27}$$

We now state and prove the following simplification of Theorem 3.2 for a QBD process. In the sequel, we will let $0_{m,n}$ denote a matrix of size $\ell_m \times \ell_n$ with a 0 at every entry.

Theorem 3.7. *The set $\mathcal{R}_0$ is a rate matrix set for Q.*

Proof. Note that in the present QBD case the matrices A_m, B_m (defined in Eq. (3.4) and (3.5)) simplify to:

$$A_m = \begin{bmatrix} 0_{m-2,m} \\ U^{m-1} \end{bmatrix},$$

$$B_m = \widetilde{U}^m + W^m, .$$

By Theorem 3.1 we know:

$$\pi^m = -\underset{\sim}{\pi}^{m-1} A_m (B_m)^{-1},$$

and because of the structure of A_m we write:

$$\underset{\sim}{\pi}^{m-1} A_m = \pi^{m-1} U^{m-1}.$$

And thus:

$$\pi^m = -\underset{\sim}{\pi}^{m-1} A_m (B_m)^{-1} = -\pi^{m-1} U^{m-1} (\widetilde{U}^m + W^m)^{-1}$$

and the proof is complete. □

Remark 3.9.
i) Note that Eq. (3.27) implies that the following recursive relation holds for all values $\nu = 0, \ldots, m-1$:

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

ii) It is easy to see that the above defined π^m and R_m satisfy the non-linear Eq. (12.2) of [70]. The matrices R_m are solutions to Eq. (12.11) of the same book, given there without explicit solution.

Remark 3.10. For each $m = 1, 2, \ldots, M_2$ the matrices R_m are easy to compute; the computation only involves inversion of the $\ell_m \times \ell_m$ matrix $\widetilde{U}^m + W^m$ and pre-multiplication of the inverse by the $\ell_{m-1} \times \ell_m$ matrix U^{m-1}.

Theorem 3.8. *The following are true for the successively lumpable QBD process $X(t)$:*

$$\pi^0 = \delta_0 \left[S_0^{M_2} \delta_0 - (\widetilde{U}^0 + W^0) \right]^{-1}$$

where

$$S_0^{M_2} = (\mathbf{1}_0' + \sum_{m=1}^{M_2} \prod_{k=1}^{m} R_k \mathbf{1}_m').$$

Proof. Direct consequence of the QBD structure and Theorems 3.2 and 3.3. □

Remark 3.11. Note that a RES QSF process does not have a simplified QBD simplification. Transition need to go up to the restart state from the lowest level, otherwise the Markov process is not positively recurrent. Therefore the process is not a QBD process.

3.6 Applications

To illustrate the application of the results we provide explicit solutions and approximations to well known open problems of queueing, cf. [9], and to a stochastic inventory theory problem, cf. [111]. The queueing models under consideration are the $M/Er/n$ queueing model with batch arrivals and the $Er/M/n$ queueing model. In the two subsequent sections, we take the number of phases to be constant, i.e. $\ell_m = \ell$ for all m, this is done solely for presentation simplicity. The analysis is easy to extend when the number of phases of the corresponding distribution is a function of "m" - the state of the queue, the number of customers in line. The steady state distribution of the *M/Er/n* is known for the case of Poisson arrivals, as for example discussed in [70]. As far as we known this is the first time the steady state distribution of the $M/Er/n$ model with batch arrivals is obtained. We note that the same book gives an exact solution procedure for QBD processes only when M_1 is finite. Below we show that our direct method works for the $Er/M/n$ queueing system, i.e., we provide explicit formulas for the rate matrix set, even when $M_1 = -\infty$. For the inventory model we show that is has the same structure as the *M/Er/n* and it can be handled similarly.

The construction in the following remark can be used to extend the applicability of the methods described in the previous section to models that are QSF and successively lumpable in the 'upward' direction. The 'QBD' version of this remark is used in Section 3.7.2.

Remark 3.12. Consider a process with a transition rate matrix Q that has the block form shown in Eq. (3.28), where its elements are labeled by $(m, i) \in \mathcal{X}$, the states of the underly-

ing process.

$$Q = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \iddots \\ \cdots & D^{m-1,m-2} & W^{m-1} & U^{m-1} & 0 & 0 & \cdots \\ \cdots & D^{m,m-2} & D^{m,m-1} & W^{m} & U^{m} & 0 & \cdots \\ \cdots & D^{m+1,m-2} & D^{m+1,m-1} & D^{m+1,m} & W^{m+1} & U^{m+1} & \cdots \\ \iddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \quad (3.28)$$

Then, we can construct a transition rate matrix $\hat{Q}$ of the form of Eq. (3.1) by relabeling the states so that a new state $(-m, i)$ corresponds to the original $(m, i) \in \mathcal{X}$ by redefining the down, within and up sub-matrices of $\hat{Q}$ as follows: $\hat{D}^{-m} = U^m$, $\hat{U}^{-m,-k} = D^{m,k}$, and $\hat{W}^{-m} = W^m$. The steady state probabilities of the Q-process can be readily obtained from those of the $\hat{Q}$-process.

In the following sections we will use the notation introduced in this chapter, although in some cases it might not be induced directly by the model.

3.7 Two classic queueing models

3.7.1 The $M/Er/n$-queue with batch arrivals

In a $M/Er/n$ queueing system with batch arrivals the service of a customer occurs in ℓ phases, each exponentially distributed with parameter μ_i for the i-th phase of the service. For notational simplicity of the exposition we describe in detail the case in which a batch may contain either 1 or 2 customers. Batches with a single customer arrive according to a Poisson process with rate $p\lambda_{m,i}$ when there are m customers in the system and the served customer has gone through the first i phases of services. Similarly, batches of 2 customers arrive with rate $(1-p)\lambda_{m,i}$ with $p \in [0, 1]$. The service of a customer has to be completed before another customer can start his first phase.

In order to have state notation that is consistent with that of Section 3.2, we use the following state description. For $i < \ell$, state (m, i) denotes the event that there are m customers in the waiting line of the system and a customer in service that has gone through i phases of the service. State (m, ℓ) denotes the event that there are m customers in the waiting line and a service completion has just occurred, so that one of the waiting customers is starting service. Note that with this awkward but convenient notation the empty state of the system is state $(0, \ell)$. Then, it is easy to see that this system can be modeled as a QSF process. Its state space is $\mathcal{X} = \{L_0, L_1, \ldots, L_{M_2}\}$ with $L_m = \{(m, 1), \ldots, (m, \ell)\}$ and $M_2 \leq \infty$. The Q matrix is defined by Eq. (3.1), with U^m, W^m and D^m (all of size $\ell \times \ell$) as given below.

$$W^0 = \begin{bmatrix} -\lambda_{0,1}-\mu_1 & \mu_1 & 0 & \cdots & 0 \\ 0 & -\lambda_{0,2}-\mu_2 & \mu_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -\lambda_{0,\ell-1}-\mu_{\ell-1} & \mu_{\ell-1} \\ 0 & \cdots & 0 & 0 & -\lambda_{0,\ell} \end{bmatrix},$$

and for $m \geq 1$:

$$W^m = \begin{bmatrix} -\lambda_{m,1}-\mu_1 & \mu_1 & 0 & \cdots & 0 \\ 0 & -\lambda_{m,2}-\mu_2 & \mu_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -\lambda_{m,\ell-1}-\mu_{\ell-1} & \mu_{\ell-1} \\ 0 & \cdots & 0 & 0 & -\lambda_{m,\ell}-\mu_\ell \end{bmatrix},$$

$$D^m = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mu_\ell & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

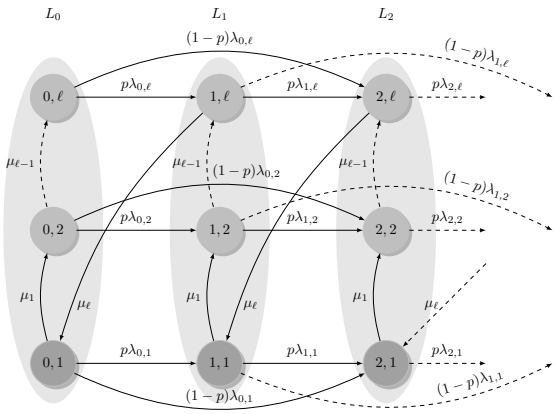


Figure 3.4: An $M/Er/n$ queueing process with batch arrivals

For $m = 0, 1, \ldots$:

$$U^{m,m+1} = p \begin{bmatrix} \lambda_{m,1} & 0 & 0 & \cdots & 0 \\ 0 & \lambda_{m,2} & 0 & \cdots & 0 \\ 0 & 0 & \lambda_{m,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_{m,\ell} \end{bmatrix},$$

$$U^{m,m+2} = (1-p)\begin{bmatrix} \lambda_{m,1} & 0 & 0 & \cdots & 0 \\ 0 & \lambda_{m,2} & 0 & \cdots & 0 \\ 0 & 0 & \lambda_{m,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_{m,\ell} \\ 0, & & & & \end{bmatrix}.$$

Note that $(m,1)$ is the entrance state of the set $\underset{\sim}{L}_m$, because D_m has a single non-zero column. The matrices A_m and B_m, described in Section 3.3, are:

$$A_m = \left[\begin{array}{ccccc} \multicolumn{5}{c}{0_{m-3,m}} \\ \hline (1-p)\lambda_{m-2,1} & 0 & 0 & \ldots & 0 \\ 0 & (1-p)\lambda_{m-2,2} & 0 & \ldots & 0 \\ 0 & 0 & (1-p)\lambda_{m-2,3} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & (1-p)\lambda_{m-2,\ell} \\ \lambda_{m-1,1} & 0 & 0 & \ldots & 0 \\ (1-p)\lambda_{m-1,2} & p\lambda_{m-1,2} & 0 & \ldots & 0 \\ (1-p)\lambda_{m-1,3} & 0 & p\lambda_{m-1,3} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (1-p)\lambda_{m-1,\ell} & 0 & 0 & \ldots & p\lambda_{m-1,\ell} \end{array}\right]$$

where $0_{m-3,m}$ is a matrix of size $\underset{\sim}{\ell}_{m-3} \times \ell_m$ with a 0 at every entry. For $m = 1, 2, \ldots$:

$$B_m = \begin{bmatrix} -\mu_1 & \mu_1 & 0 & \cdots & 0 \\ \lambda_{m,2} & -\lambda_{m,2} - \mu_2 & \mu_2 & \cdots & 0 \\ \lambda_{m,3} & 0 & -\lambda_{m,3} - \mu_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_{m,\ell} & 0 & 0 & \cdots & -\lambda_{m,\ell} - \mu_\ell \end{bmatrix}$$

and

$$B_0 = \begin{bmatrix} -\mu_1 & \mu_2 & 0 & \cdots & 0 \\ \lambda_{0,2} & -\lambda_{0,2} - \mu_2 & \mu_2 & \cdots & 0 \\ \lambda_{0,3} & 0 & -\lambda_{0,3} - \mu_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_{0,\ell} & 0 & 0 & \cdots & -\lambda_{0,\ell} \end{bmatrix}.$$

Now we can calculate R^1_m using Eq. (3.11): $R^1_m = -A_m(B_m)^{-1}$, where $\pi^m = \underset{\sim}{\pi}^{m-1} R^1_m$.

Since the first $\underset{\sim}{\ell}_{m-3}$ rows of R^1_m are zero (due to multiplication of $(B_m)^{-1}$ with the $0_{m-3,m}$ sub-matrix of A_m) this expression reduces to the following:

$$\pi^m = [\pi^{m-2} | \pi^{m-1}] R^{*1}_m,$$

where R^{*1}_m denotes the non-zero rows of R^1_m.

Using the results above, we can construct the sequence of rate matrices R^k_m using Eq. (3.10) and the notation R^{*k}_m for the sub-matrix of the non-zero rows of R^k_m we obtain straightfor-

wardly:

$$\pi^m = [\pi^{m-k-1}|\pi^{m-k}]R_m^{*k},$$

and $\pi^m = \pi^0 R_m^{*m}$.

When M_2 is finite (i.e. there is a finite buffer for the number of customers allowed in the system), then Theorem 3.3 readily provides the solution: $\pi^0 = \delta_0 \left[S_0^{M_2}\delta_0 - B_0\right]^{-1}$, $\pi^m = \pi^0 R_m^{*m}$.

When M_2 is infinite, using Proposition 3.2, we can construct upper bounds for $\pi(m,i)$ via the process $X_{m_2}(t)$ described therein. This result is stated in the next theorem.

Theorem 3.9. *The following is true for the $M/Er/n$ model with batch arrivals:*

$$\pi^0_{X_{m_2}} = \delta_0 \left[S_0^{m_2}\delta_0 - B_0\right]^{-1} \tag{3.29}$$

where $S_0^{m_2} = [\mathbf{1}_0' + \sum_{m=1}^{m_2} R_m^{*m}\mathbf{1}_m']$ *and*

$$\pi^m \leq \pi^m_{X_{m_2}} = \pi^0_{X_{m_2}} R_m^{*m}.$$

Proof. Directly from Theorem 3.3 (for Eq. (3.29)) and Section 3.3 (Proposition 3.2) for the second claim. □

3.7.2 The $Er/M/n$-queue

In this section we derive limit solutions for the $Er/M/n$ queueing system. Specifically we consider a system with n identical servers, where the service time of a customer is exponentially distributed with parameter μ. The inter-arrival times are modeled as a sum of ℓ ($\ell < \infty$) distinguishable exponentially distributed 'phases', where the rate of the i-th phase may be a function of the number m of customers in line and it is denoted by $\lambda_{m,i}$. We can model the $Er/M/n$ queueing process as a QBD process on the state space $\mathcal{X} = \{L_0, L_1, \ldots\}$ with $L_m = \{(m,1), \ldots, (m,\ell)\}$. We use the state notation (m,i) to denote the event that there are m customers in the system and the customer inter-arrival time is at its i-th phase. In this QBD process, U^m, W^m and D^m are of size $\ell \times \ell$ and are given below. For $m = 0, 1, \ldots$:

$$D^m = \mu_m I_\ell, \quad U^m = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \lambda_{m,\ell} & 0 & \cdots & 0 & 0 \end{bmatrix},$$

$$W^0 = \begin{bmatrix} -\lambda_{0,1} & \lambda_{0,1} & 0 & \cdots & 0 \\ 0 & -\lambda_{0,2} & \lambda_{0,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -\lambda_{0,\ell-1} & \lambda_{0,\ell-1} \\ 0 & 0 & \cdots & 0 & -\lambda_{0,\ell} \end{bmatrix},$$

and for $m = 1, 2, \ldots$:

$$W^m = \begin{bmatrix} -(\lambda_{m,1}+\mu_m) & \lambda_{m,1} & 0 & \cdots & 0 \\ 0 & -(\lambda_{m,2}+\mu_m) & \lambda_{m,2} & \cdots & 0 \\ \vdots & \vdots & \vdots\ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -(\lambda_{m,\ell}+\mu_m) & \lambda_{m,\ell} \\ 0 & 0 & \cdots & 0 & -(\lambda_{m,\ell}+\mu_m) \end{bmatrix},$$

where $\mu_m = m\mu$ for $m \leq n-1$, $\mu_m = n\mu$ for $m \geq n$. We can now use the relabeling specified in Remark 3.12. In this case $\hat{D}^{-m} = U^m, \hat{U}^{-m} = D^m$, and $\hat{W}^{-m} = W^m$. The relabeled process satisfies Assumption 3.1 and has a QBD structure with $M_1 = -\infty$. The state space can be truncated at level m_1 as in Remark 3.6 (where the $\overline{\hat{w}}$'s are defined). Then, Eq. (3.27) and Theorem 3.8 from Appendix B can be used to compute limiting approximations $\hat{\pi}_{X_{m_1}}(-m, i)$ for the steady state probabilities $\pi(m, i)$ of the $Er/M/n$ model as described below.

$$\hat{R}_{-m} = \mu_{m+1} \begin{bmatrix} \lambda_{m,1} & -\lambda_{m,1} & 0 & \cdots & 0 \\ -\mu_m & \lambda_{m,2}+\mu_m & -\lambda_{m,2} & \cdots & 0 \\ -\mu_m & 0 & \lambda_{m,3}+\mu_m & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mu_m & 0 & \cdots & 0 & \lambda_{m,\ell}+\mu_m \end{bmatrix}^{-1},$$

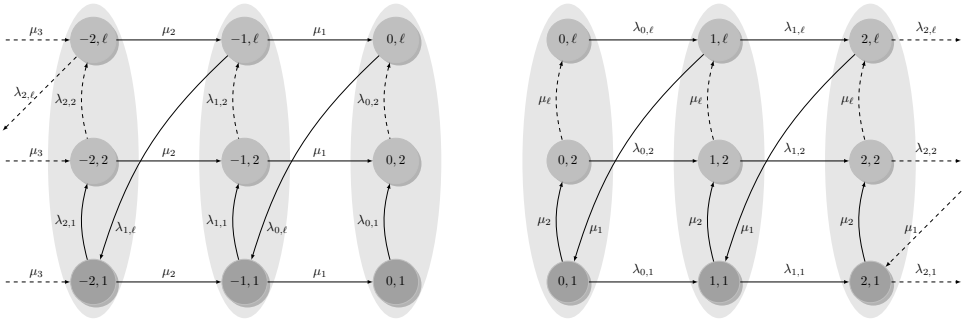
$$\hat{R}_0 = \mu_1 \begin{bmatrix} \lambda_{0\,1} & -\lambda_{0\,1} & 0 & \cdots & 0 \\ 0 & \lambda_{0\,2} & -\lambda_{0\,2} & \cdots & 0 \\ 0 & 0 & \lambda_{0\,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_{0\,\ell} \end{bmatrix}^{-1}.$$

The following are true for the steady state probabilities of the $\hat{Q}$-process are: $\hat{\pi}^{m_1}_{X_{m_1}} = \delta_{m_1}\left[S^0_{m_1}\delta_{m_1} - \widetilde{\hat{U}}^{\,m_1} - \overline{\hat{W}}^{\,m_1}\right]^{-1}$, $\hat{\pi}^{m}_{X_{m_1}} = \hat{\pi}^{m_1}_{X_{m_1}} \prod_{k=m_1}^{m-1} \hat{R}_k$ where $S^0_{m_1} = 1'_{m_1} + \sum_{m=m_1}^{0} \prod_{k=m_1}^{m-1} \hat{R}_k 1'_m$.

In addition, by Remark 3.12 we have:

$$\pi(m, i) = \hat{\pi}(-m, i) = \lim_{m_1 \to -\infty} \hat{\pi}_{X_{m_1}}(-m, i).$$

The homogeneous $M/Er/n$ and the $Er/M/n$ queueing systems have a very similar structure when the number of phases ℓ is equal. When considering the relabeled level process of $Er/M/n$ process and change the role of λ and μ, the $Er/M/n$ process is exactly the negative extension of the $M/Er/n$ queue as it is shown in Figure 3.5.



(a) The $Er/M/n$ process

(b) The $M/Er/n$ process

Figure 3.5: The left figure displays the transition rate diagram of the $\hat{Q}$ matrix of the $Er/M/n$ queue; the right figure is the diagram for the $M/Er/n$ queue.

3.8 An inventory model with random yield

In this section we consider an inventory model with random yield. Specifically, we investigate a system where customers arrive with rate λ and a batch of products arrives according to an exponential distribution with rate μ. Random yield is possible in this model, i.e. the size of the batch is $n\ell$ with probability p_n, where ℓ is a fixed positive constant, cf. [111]. We model this inventory model process as a QSF process $X(t)$ on state space $\mathcal{X} = \{L_0, L_1, \ldots\}$ with $L_m = \{(m,1), \ldots, (m,\ell)\}$. In state (m,i) there are $m\ell + i$ products in stock. Figure 3.6 displays the transition diagram of the described model with $\ell = 3$ and where the size of the batch is 3 with probability p and 6 with probability $1-p$. This model is a successively lumpable QSF process, where states $(m, \ell-1)$ are the entrance states of sub-sets $\underset{\sim}{L}_m$. In this QSF process, $U^{m,k}$, W^m and D^m are of size $\ell \times \ell$. It has the same structure as the queueing model described in Section 3.7.1 and it can be solved analogously.

We note that we can easily obtain explicit formulas for the steady state probabilities even in the case that both λ and μ may depend on the state. For example, when there is too much (little) inventory a discount (premium) price may be used for the product and this may change the arrival rate of the customers, i.e., $\lambda = \lambda(m,i)$. Also, when there is a high level

of inventory one may decide not to order. This can be easily incorporated in product arrival rate, i.e., $\mu = \mu(m, i)$. Finally we note that just as easily one can handle the extension where the batch size $n\ell$ is replaced by $n\ell_m$ to represent dependency on the inventory level m. We omitted all these dependencies in this exposition only to simplify the notation.

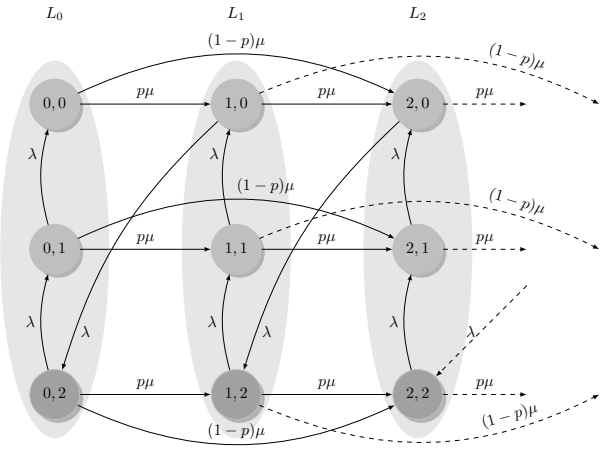


Figure 3.6: An Inventory Model with random yield.

3.9 A restart system

Consider the classical restart (reliability) problem where a system of known structure is composed of N components and it operates continuously. The time to failure of component $i = 1, \ldots, N$ is exponentially distributed with rate μ_i and it is independent of the state of the other components. This type of systems has been studied in [36], [58], [43] and [51] as well as [60], [87], [68] and [105].

In this section we assume that when the system fails it is restored (or replaced) to a state "as good as new" and the time it takes for this restoration is exponentially distributed with rate λ.

These assumptions imply that at any point in time the state of the system can be identified by a boolean M-vector $x = (x_1, \ldots, x_M)$, with $x_i = 1$ if the $i-$th component is working, else $x_i = 0$. Hence $\mathcal{X} = \{0, 1\}^M$ is the set of all possible states. Under these conditions the time evolution of the state of the system can be described by a continuous time Markov

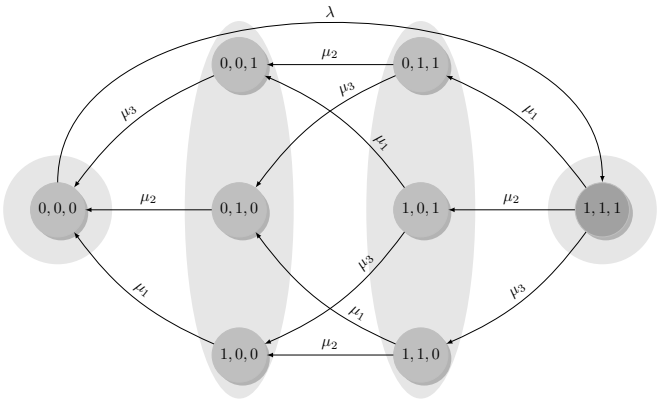


Figure 3.7: Transition diagram for parallel system with $M = 3$ servers.

process. The structure of the system is specified by a binary function ϕ defined on $\mathcal{X}$. Let $G = \{x : \phi(x) = 1\}$ denote the set of all operational (good) states of the system and let $B = \{x : \phi(x) = 0\}$ denote all failed states of the system. For such a system it is important to compute measures of performance such as the availability of the system defined as $\alpha_\phi = \sum_{x \in G} \pi(x)$. Regardless of the choice of the structure ϕ it is easy to see that the corresponding process is successively lumpable.

For example, for the parallel system we have $B = \{(0, \ldots, 0)\}$. Figure 3.7 illustrates the transition diagram of the corresponding Markov process for the parallel system when $M = 3$. It is clear that this process is a RES QSF process with respect to partition $\mathcal{D} = \{L_0, \ldots, L_3\}$ of size $M + 1$, with $\forall x \in \mathcal{X}$:

$$x \in L_m, \text{ if } \sum_i x_i = m.$$

In this example of a restart process, when modeled as a QSF process, $M_1 = 0$ and $M_2 = 3$. The states are ordered as is shown in Figure 3.7, i.e. for example $(0, 0, 1)$ is the first state of level 1. We derive that D^m, W^m and U^{m,M_2} have the from given below:

$$W^0 = -\lambda, \;\; U^{0,3} = \lambda,$$

and:

$$D^1 = \begin{bmatrix} \mu_3 \\ \mu_2 \\ \mu_1 \end{bmatrix}, \;\; W^1 = \begin{bmatrix} -\mu_3 & 0 & 0 \\ 0 & -\mu_2 & 0 \\ 0 & 0 & -\mu_1 \end{bmatrix}, \;\; U^{1,3} = 0'_1$$

$$D^2 = \begin{bmatrix} \mu_2 & \mu_3 & 0 \\ \mu_1 & 0 & \mu_3 \\ 0 & \mu_1 & \mu_2 \end{bmatrix}, \;\; W^2 = \begin{bmatrix} -\mu_2 - \mu_3 & 0 & 0 \\ 0 & -\mu_1 - \mu_3 & 0 \\ 0 & 0 & -\mu_1 - \mu_2 \end{bmatrix}, \;\; U^{2,3} = 0'_2$$

and

$$D^3 = \left[\begin{array}{ccc} \mu_1 & \mu_2 & \mu_3 \end{array} \right], \; W^3 = -(\mu_1 + \mu_2 + \mu_3).$$

Using the first step in Algorithm 3.2 we derive the following rate matrices:

$$R_0 = -D^1(W^0)^{-1} = 1/\lambda \left[\begin{array}{c} \mu_3 \\ \mu_2 \\ \mu_1 \end{array} \right], R_1 = -D^2(W^1)^{-1} = \left[\begin{array}{ccc} \mu_2/\mu_3 & \mu_3/\mu_2 & 0 \\ \mu_1/\mu_3 & 0 & \mu_3/\mu_1 \\ 0 & \mu_1/\mu_2 & \mu_2/\mu_1 \end{array} \right],$$
$$R_2 = -D^3(W^2)^{-1} = \left[\begin{array}{ccc} \mu_1/(\mu_2+\mu_3) & \mu_2/(\mu_1+\mu_3) & \mu_3/(\mu_1+\mu_2) \end{array} \right].$$

We can find the steady state distribution with the remaining steps of Algorithm 3.2. Note that in this case T_0^3 is a scalar and $\delta_3 = 1$.

- $T_0^3 = 1 + \sum_{m=0}^{2} \prod_{k=0}^{2-m} R_{2-k} 1'_m = 1 + R_2 R_1 R_0 1'_0 + R_2 R_1 1'_1 + R_2 1'_2$,
- $\pi^3 = \pi(1,1,1) = \delta_3 [T_0^3 \delta_3 - W^3 - \widetilde{D}^3]^{-1} = [T_0^3]^{-1}$,
- $\pi^2 = \pi^3 R_2$, $\pi^1 = \pi^2 R_1$, $\pi^0 = \pi^1 R_0$.

It is important to note that the successively lumpable property of the process results in the following computational gains for large M: instead of solving a system of size 2^M we only need to solve M systems the largest of which is of size $\binom{M}{\lfloor M/2 \rfloor} + 1$.

CHAPTER 4

LPC compared with successive lumping

This chapter appeared as: *A Comparative Analysis of the Successive Lumping and the Lattice Path Counting Algorithms*, cf. [S4].

4.1 Introduction to Chapter 4

Two dimensional Markov processes arise as a natural way 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 to [9, 11, 12, 32, 37, 38, 83, 104, 113, 118]. Other areas in which these processes will arise outside queueing are for example inventory models, cf. [S1], reliability, cf. [62, 61] and pricing models. In this chapter we are particularly interested in a comparison of the new successive lumping (SL) methodology described in Chapter 2 with the popular lattice path counting [78] in obtaining rate matrices for queueing models, as in [108] and [107]. The two methodologies are compared both in terms of applicability requirements and numerical complexity by analysing their performance for the same classical queueing models considered in [107]. 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 two dimensional Markov process with a transition generator matrix Q that contains non-zero 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 analyse 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. For general level-independent QBD processes, it is known that R satisfies a matrix-quadratic equation. Algorithms for solving this equation were given in [80] and Latouche and

Ramaswami [70]. A current state of the art software implementing quadratically-convergent algorithms with a number of speed-up features is described in [31]. A general algorithm for the level-independent case can be found in [33] and a discussion of the Quasi Skip Free case in [72].

There are various methods that make use of a special structure of the transition rate matrix Q, to provide efficient computation procedures for the rate matrix R. Such a procedure is available in the 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 to [106] and [79]. Recent studies, cf. [108, 107], 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 analysed by this method, but also with other techniques, see e.g. [45] and references therein. The idea of counting the number of paths on a lattice, cf. [78, 40], has been used in many fields of applied probability, cf. [98].

A new alternative method to compute the rate matrix for certain QBD processes can be based on the successive lumping (SL) procedure described in Chapter 2. It is employed in Chapter 3 to obtain explicit solutions for 'rate sets' for large classes of QSF processes, the so-called DES and RES 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 [45, 70, 33] can be used on other, more general (in terms of down-transitions) processes. The advantages of using SL are described in Chapter 3. Although the nature of a path counting based method and the successive lumping based method are very different, a comparison can be done, since they both rely on the absence of certain kind of transitions. Herein we compare the method introduced in [107] with the one based on successive lumping for QSF processes.

The main goal of this chapter is to provide a clear comparison between successive lumping (SL) based methods and the lattice path counting based algorithm, introduced in [107], 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 non-homogenous transition rate structures and to models with a quasi skip free (QSF) structure of Chapter 3. However, there seem to exist some artificial queueing models that do not possess the SL property, for which a lattice path counting algorithm is applicable. Finally, this chapter continues the work of Chapter 3, and it specializes its results to homogenous QBD processes, in order to make the comparison of successive lumping (SL) based methods and the lattice path counting procedure possible.

This chapter has the following structure. In Section 4.2 we first define the notation for the QBD processes that we will use throughout this chapter. In Section 4.2.1 we summarize the results of Chapter 3 for the DES processes as they apply to quasi birth and death processes with a down entrance state and the resulting *quasi birth and death down entrance state al-*

gorithm (QDESA). In Section 4.2.2 the QDESA procedure is specialized depending on the structure of the transition rate Q, applicable to the models under investigation in this chapter. Then, in Section 4.3 the introduced procedures are clarified by applying them to two specific queueing examples. In Section 4.4 we review the lattice path counting algorithm. In Section 4.5 we compare the procedures in speed (computational complexity). In Section 4.6 we discuss the the type of models for which each procedure can be applied. We conclude with some models that further illustrate these comparisons.

4.2 Preliminary results

4.2.1 Successive lumping in QBD processes

In the sequel we consider an ergodic QBD process $X(t)$ with states in a finite or countable set $\mathcal{X}$. The states (after re-labeling) will be written as tuples (m, i), where in the state description the first entry $m = 0, 1, \ldots, M$ represents the 'level' of the state and the second entry $i = 0, 1, 2, \ldots, \ell_m$ 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 $(\ell_m + 1) \times (\ell_{m+1} + 1)$. Further, we will use the notation $\mathcal{L}_n = \{(n, i),\ i = 0, 1, \ldots, \ell\}$ for the level sets $(n = 0, 1, \ldots, M)$.

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., $\pi^n = [\, \pi(n, 0), \ldots, \pi(n, \ell)\,]$.

In the context of the current chapter we will assume that every matrix D^m has only one non-zero 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 Chapter 2 for lumping and Chapter 3 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 LPC procedure, that will be discussed in Section 4.4. Below we will repeat the important definitions from Chapter 3, specialized for a QBD process.

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

$$\widetilde{U}^m = U^m 1'_m \delta_m,$$

where 1_m is a rowvector of size $\ell+1$ with 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:

$$B^m = W^m + \widetilde{U}^m. \tag{4.1}$$

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

$$\pi^m = \pi^{m-1}\mathcal{R}_m\,, \quad \text{for } m = 1,\ldots,M_2. \tag{4.2}$$

In Chapter 3 it was shown that the matrix B_m is invertible. A simplification of Theorem 2 of that paper for the special case of a QBD process implies that the matrix set $\mathcal{R}_0 := \{R_m\}_m$ defined by:

$$R_m = -U^{m-1}(B^m)^{-1}, \tag{4.3}$$

is a rate matrix set for Q, when D^m has a single non-zero column.

Remark 4.1.
i) Note that Eq. (4.2) and Eq. (4.3) imply that the following recursive relation holds for all $\nu = 0,\ldots,m-1$:

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

ii) It is easy to see that the above defined π^m and R_m satisfy the non-linear Eq. (12.2) of [70]. The matrices R_m are solutions to Eq. (12.11) of the same book, given there but without the explicit procedure of Eq. (4.3) to compute them.

To obtain the steady state distribution, $\pi = [\pi^0, \pi^1, \ldots]$, one only needs to compute π^0, which per Theorem 3.3, is given by Eq. (4.4) - (4.5) below.

$$\pi^0 = \delta_0 \left[S_0^{M_2}\delta_0 - B^0 \right]^{-1}, \tag{4.4}$$

where

$$S_0^{M_2} = 1'_0 + \sum_{m=1}^{M_2} \prod_{k=1}^{m} R_k 1'_m. \tag{4.5}$$

The procedure to calculate the steady state distribution π when there is a down entrance state in every level that is based on Eq. (4.4), (4.5) and (4.3) above will be referred to in the sequel as the *quasi birth and death down entrance state algorithm* (QDESA).

4.2.2 Solution procedures for specific QBD processes

Unless otherwise stated in the remainder of this chapter we will consider homogenous level processes. Note that for these processes $B^m = B = W + \widetilde{U}$ (defined in Eq. (4.1)) 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 down entrance state where the matrix B is of countable dimension and has the following form:

$$B = \begin{bmatrix} -b_0^d - b_0^u & b_0^u & 0 & 0 & 0 & \cdots \\ b_1^d + b_1^z & -b_1^w & b_1^u & 0 & 0 & \cdots \\ b_2^z & b_2^d & -b_2^w & b_2^u & 0 & \ddots \\ b_3^z & 0 & b_3^d & -b_3^w & b_3^u & \ddots \\ b_4^z & 0 & 0 & b_4^d & -b_4^w & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{4.6}$$

where

$$b_i^w = b_i^z + b_i^d + b_i^u,$$

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

Second, we consider homogenous QBD processes with a down entrance state where the matrix B has the structure of Eq. (4.6) and is *element homogenous* i.e.,

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

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

In Chapter 5 we present a fast $\mathcal{O}(\ell^2)$ algorithm to compute the inverse of matrix B of Eq. (4.6), when it is element homogenous, and thus used in QDESA^{++}. In that chapter we described a procedure with the same complexity to compute the inverse of B, when it has the structure of Eq. (4.6) and it is not required to be element homogenous. An alternative method of computation with the same complexity is given in [50], pp. 62, but only if $\ell < \infty$ and B is element homogenous.

Remark 4.2. One can determine which solution method is applicable by inspection of the matrix Q. If W^m has a birth and death structure, QDESA$^+$ is applicable, and when both W and $\widetilde{U}$ have a homogenous birth and death structure, QDESA^{++} is applicable.

When W has another structure than the one described above, it might still have a sparse form. In that case it might be beneficial to use other fast matrix inversion algorithms, like in [48] and [115].

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

4.3 Applications: classic queueing models

In this section we will discuss two classical queueing models and analyse 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 4.3.1, or the longest queue of Section 4.3.2, respectively.

4.3.1 A priority queue model

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 4.1.

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 $d = (\lambda_1 + \lambda_2 + \mu)$. 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.

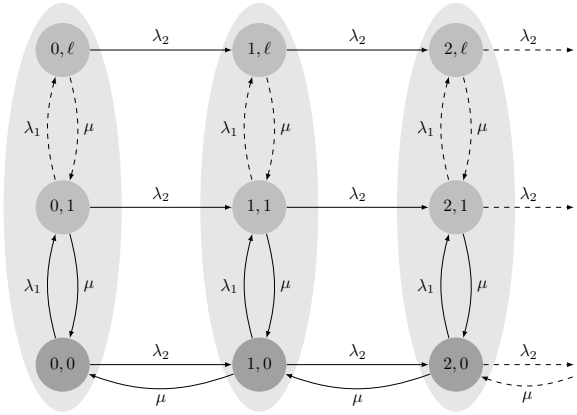


Figure 4.1: Transition diagram of the priority queue model.

$$D = \begin{bmatrix} \mu & 0 & \cdots \\ 0 & 0 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, U, U^0 = \begin{bmatrix} \lambda_2 & 0 & 0 & \cdots \\ 0 & \lambda_2 & 0 & \ddots \\ 0 & 0 & \lambda_2 & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}, W = \begin{bmatrix} -d & \lambda_1 & 0 & \cdots \\ \mu & -d & \lambda_1 & \ddots \\ 0 & \mu & -d & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix},$$

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

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

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.

4.3.2 A longest queue model

In a longest queue model, cf. [119], two types of customers 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 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 4.6.2.

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 3.2 and entrance states $(n, 1)$ for level n where matrices D, U, W as given below, $d = 2\lambda + \mu$. 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.

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

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

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

Note that the matrix B^L has a structure similar (but not identical) to that of B defined in Eq. (4.6); its structure from the second column on is identical to that of B, but an extra column has been added in front. This can be easily resolved with a suitable modification of QDESA^{++}.

Remark 4.3. The Feedback queue, the third model that is discussed in [107], fits the QDESA framework as well; its analysis goes analogous to the analysis of the priority queue.

4.4 Lattice path counting

A different approach to compute the steady state distribution π for a class of Markov process that includes the queueing models described before, is the *Lattice Path Counting Algorithm* (LPCA) of [108], see also [107]. In this section we will repeat LPCA in the notation used in this chapter.

Throughout this chapter we use a labeling of states that is consistent with our notation introduced in Chapter 2 and 3. In [107] 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 [107] 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 $\mathcal{L}_n = \{(n, i),\ i = 1, \ldots, \ell\}$ where $n = 0, 1, \ldots$ 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 $(0, 1), \ldots, (M, 1), (0, 2), \ldots (M, 2), \ldots, (0, \ell), \ldots, (M, \ell)$. In this case we define the stage sets to be: $\mathcal{K}_i = \{(n, i),\ n = 0, 1, \ldots\}$. 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 the current chapter is the same as the matrix Q of [107], subject to appropriate relabeling of states, as is mentioned above. Note that in that paper the notation M is used for our ℓ above and their corresponding ℓ is infinite.

Following the approach introduced in [107], 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 stage $i > 0$: $\pi_{i+1} = \pi_i \widehat{R}$,

where $\widehat{R}$ is the minimal non-negative 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 Eq. (4.3) above.

Figure 4.2 displays 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.

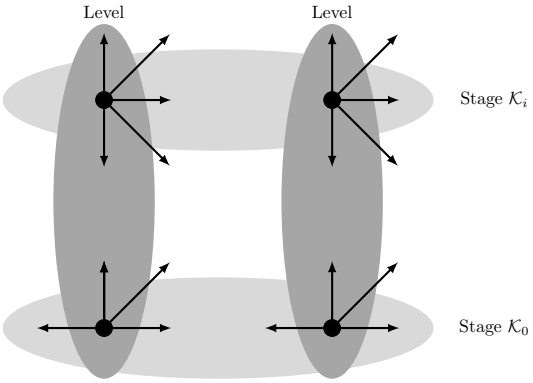


Figure 4.2: Levels and Stages.

Further, it is known, cf. for example [70], that the elements $\hat{r}(n|m)$ of the matrix $\widehat{R} = [\hat{r}(n|m)]$ 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:

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

Theorem 4.1 below provides an explicit expression for the elements of $\widehat{R}$. It is the main result of [107] and uses the following expressions:

$$
\begin{aligned}
P_h(s,u,m) &= \phi\langle 1,-1\rangle^s \phi\langle 1,0\rangle^t \phi\langle 1,1\rangle^u \phi\langle 0,1\rangle^{m-u} \phi\langle 0,-1\rangle^{m+1-s} \\
L_h(s,u,m) &= \frac{1}{m+1}\binom{2m}{m}\binom{m+1}{s}\binom{m}{u}\binom{2m+t}{t} \\
G_h &= \sum_{s=0}^{h}\sum_{u=0}^{h-s}\sum_{m=\max(u,s-1)}^{\infty} L_h(s,u,m)P_h(s,u,m) \qquad (4.7)\\
\kappa_h &= \frac{\phi\langle 1,0\rangle\kappa_{h-1} + \phi\langle 0,1\rangle \sum_{j=0}^{h-1} G_{h-j}\kappa_j + \phi\langle 1,1\rangle \sum_{j=0}^{h-1} G_{h-j-1}\kappa_j}{1-\phi\langle 0,1\rangle G_0},
\end{aligned}
$$

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

$$
\hat{r}_h = 2\,\frac{\phi\langle 0,1\rangle\kappa_h + \phi\langle 1,1\rangle\kappa_{h-1}}{1+\sqrt{1-4\phi\langle 0,1\rangle\phi\langle 0,-1\rangle}}. \qquad (4.8)
$$

The LPCA is based on the calculation of Eq. (4.8), utilizing a new computation of the G_h in Eq. (4.7) above using hypergeometric functions, cf. Eq. (26) and (27) of [107].

4.5 Comparative analysis

In this section we will compare the efficiency of LPCA and QDESA described in the previous section. To make a fair comparison between these algorithms we will compare their complexities in Section 4.5.1 for transition rate matrices on which they can *both* be applied. In Section 4.6 we discuss classes of models for which a version of QDESA is applicable while LPCA is not. We will also distinguish structures for which the LPCA can be used efficiently, but for which QDESA is not readily applicable.

It is important to note that 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 Section 5 of [107] and summarized in the previous section. The non-homogeneous part of the state space is considered to be (part of) stage $\mathcal{K}_0$. This non-homogeneous part may induce that 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 to Chapter CH:PF. In that we will describe 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, that QDESA is no longer applicable.

When a process has such a structure that QDESA applies (with respect to the levels) *and* LPCA (with respect to the stages) we note that B, (where $R = UB^{-1}$) has to have the structure of Eq. (4.6), up to a permutation of the columns, due to the fact that the process is a QBD process in the stage direction, see Remark 4.2. 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 4.1. *Suppose that the following are both true:*

- *LPCA is applicable to a QBD process with respect to the stages,*
- *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 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 LPCA with QDESA^{++}.

4.5.1 Computational complexity of the procedures

By Eq. (4.3) we know that the computational complexity of 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 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 4.2 below. To compare the complexities of QDESA to that of LPCA, we take $\ell = M$, e.g. 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 $\mathcal{O}(1)$.

Theorem 4.2. *When the steady state distribution of a QBD process can be found both by using LPCA and using QDESA the following are true:*

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

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

Proof. To prove part *i*) we assign 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, cf. Eq. (26) and Eq. (27) of [107], 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 Eq. (4.7) we see that 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 non-zero elements, $\hat{r}_0, \ldots, \hat{r}_{M-1}$ and each of these computations is of complexity $\mathcal{O}(h^3)$. The complexity of the computation of rate matrix $\widehat{R}$ is: $\sum_{h=0}^{M-1} \mathcal{O}(h^3) = \mathcal{O}(M^4)$.

For part *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 3 non-zero elements per row), induced by the fact that 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. The proof is complete. □

Remark 4.4. For some special cases, e.g. the priority queue, the complexity of 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 LPCA is $\mathcal{O}(M^2)$, because in the computation of G_h, both $s = 0$ and $u = 0$ and the summation in Eq. (4.7) is only over m; i.e., the complexities of LPCA and QDESA are the same in this case.

Remark 4.5. When there is no additional structure on matrix B, QDESA^{+} and QDESA^{++} both can not be used, so we need a general matrix inversion to compute B^{-1} of dimension ℓ by ℓ that is in complexity less than $\mathcal{O}(\ell^{2.379})$, cf. [114], when ℓ is finite. When U is a non-sparse matrix this provides a solution procedure with total complexity $\mathcal{O}(l^3)$ for QDESA.

4.6 The applicability of QDESA to more general models

In this section we will determine the differences in applicability between QDESA and LPCA, and display these differences with examples. We will consider variations of the queues in Section 4.3.1 and 4.3.2 that can be solved with QDESA but not with LPCA.

One of the main advantages of QDESA over LPCA is that 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 LPCA does not require any restrictions on the non-homogenous part $\mathcal{K}_0$, the structure on this set can be very complex and a direct technique to do this step is absent and not trivial to include. Therefore QDESA can be viewed as a more complete solution procedure. And 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 LPCA, but perhaps not with QDESA, within the LPCA no procedure is provided to find the steady state distribution.

There are four important classes of models for which (an extension of) QDESA is applicable and for which the LPCA can not be used at all. The first class involves element non-homogenous 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 3.2; 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 involve all DES processes, i.e., Markov processes 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 as in Chapter 3.

Conversely, there are processes for which the LPCA is applicable, but 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 one can construct a DES process in a lot of cases.

Table 4.1 identifies 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. matrix B_0 and B_1 are possibly non-sparse matrices in the LPCA procedure. We compare this with the restrictions that are imposed by QDESA.

4.6.1 A priority queue model with batch arrivals

Consider the priority queue model where two types of customers 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 iid random variables that follow a known discrete distribution: $P(Z_i = z) = p_i(z)$.

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

Stage $\mathcal{K}_0$, the Non-Homogeneous 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.
Element Non-Homogeneous.	Element Non-Homogeneous.
Sol. Proc. 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.
Element *Homogeneous*.	Element Non-Homogeneous.
No transitions to 'NW', 'W', 'SW' allowed.	Trans. to 'W' allowed to *entrance* state.
Number of stages must be infinite.	Number of stages can be finite or infinite.

Table 4.1: Restrictions for the applicability of LPCA and QDESA.

2 (respectively in queue 1). Because we assume that there is no maximum for 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 $d = (\lambda_1 + \lambda_2 + \mu)$:

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

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 QDESA, but LPCA is not applicable.

4.6.2 A longest queue model with non-homogeneous arrival rates

We will extend the model discussed in Section 4.3.2 in such a way that now two types of customers 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 4.3.2 does not induce

a Markov process. 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 4.3 and the level partition is highlighted by the grey background. It is easy to deduct that this is a DES process where the level sets $\mathcal{L}$ are formally described as follows:

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

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

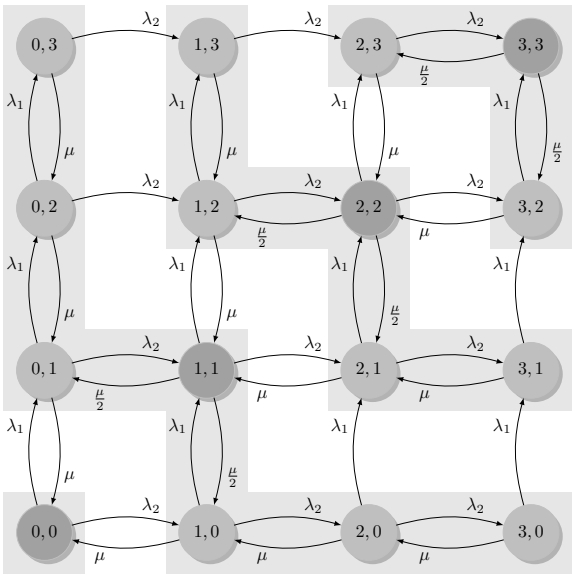


Figure 4.3: Longest Queue model.

CHAPTER 5

The inverse of a restart birth-and-death matrix

This chapter will appear as: *On the Solution to a System of Equations arising in Stochastic Processes*, cf. [S5].

5.1 Introduction to Chapter 5

5.1.1 Motivation

In this chapter we develop a method to compute the solution to a countable (finite or infinite) set of equations that occurs in many different fields and systems including Markov processes that model queueing systems, birth-and-death processes and inventory systems. For such systems in order to compute performance measures and other quantities of interest, it is often required to invert the matrix associated with the transition rates of the system. A class of problems for which the inverse of this matrix must be computed is given in Chapter 4. The method provides a fast and exact computation of this inverse matrix and applies to more general systems of linear equations. If the matrix is of countable size, the method provides an exact solution, independent of an arbitrary truncation size. In contrast, alternative inverse techniques perform much slower and work only for finite size matrices. The more relevant alternative methods are discussed in this chapter, for comparison purposes. It is shown that although some of these methods cover more general classes of matrices, the method developed in this chapter provides a procedure for matrices of infinite size and outperforms all alternative methods in speed. As far as we could find, there are no results in literature for countably sized matrices of this specific form. Existing algorithms for finite matrices in general perform slower than the method we provide.

Apart from the inverse, we have also identified a fast way to compute the eigenvalues of the matrix under consideration, starting with those of a much easier to analyse matrix, one with a

birth-and-death structure. Knowing the values of these eigenvalues or just their bounds may considerably aid the analysis of the models of interest.

This chapter is organized as follows. First we will introduce some of the notation that is used throughout this chapter and necessary to understand the review of other methods. Second, in Section 5.1.3, we will identify some of the existing procedures and specify in what directions they overlap the method discussed in this chapter. In Section 5.2 we formally provide the method with its specifications subdivided in four possible matrix forms. Specifically, see Algorithm 5.1, for the computation of the inverse. Next, in Section 5.3 we will exploit the structure of the matrix and derive some results regarding its eigenvalues. We present these results in a more general matrix framework, and apply the results to the matrix under consideration. Finally, in Section 5.4 we will give several applications wherein this matrix-structure appears naturally.

5.1.2 Preliminaries

In this chapter we develop an efficient computation procedure for the solution vector x of size $\ell+1$ ($\ell \leq \infty$) of a possibly countable system of linear equations of the form

$$xB = y, \tag{5.1}$$

where y is a vector of size $\ell+1$, B is a matrix of size $(\ell+1)\times(\ell+1)$ and it has a structure of the form below:

$$B = \begin{bmatrix} -b_0^d - b_0^u & b_0^u & 0 & 0 & 0 & \cdots \\ b_1^d + b_1^z & -b_1^w & b_1^u & 0 & 0 & \cdots \\ b_2^z & b_2^d & -b_2^w & b_2^u & 0 & \ddots \\ b_3^z & 0 & b_3^d & -b_3^w & b_3^u & \ddots \\ b_4^z & 0 & 0 & b_4^d & -b_4^w & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}. \tag{5.2}$$

In the above for $i = 1, 2, 3, \ldots$ and $j = d, w, u$, the indices b_i^j are non-negative real numbers and for $i \geq 1$ they must satisfy the conditions below:

$$b_i^z + b_i^d + b_i^u = b_i^w, \tag{5.3}$$

$$b_i^z + b_i^d > 0, \tag{5.4}$$

$$b_0^d > 0. \tag{5.5}$$

Also, we assume that the return probability to the first state (when B is a transition rate

matrix of a transient Markov process) is 1. This could for example be realized by assuming that $b_i^z > 0$ or by assuming that $b_i^d > \epsilon > b_i^u$ for some ϵ and all $i = 1, 2, \ldots$.

In Section 5.2 it will be shown that relations (5.3)-(5.5) imply the existence of a non-positive inverse of the matrix B, i.e., $C = B^{-1}$. See also Lemma 6.1 of this thesis for a proof of the existence of a negative solution of $CB = BC = I$ when ℓ is countable. In the latter case C may not be unique, but we are interested in computing the maximum negative solution for the applications we have in mind. Furthermore, we will provide constructive algorithms for computing C, and thus a way to compute the solution x to Eq. (5.1), provided that Cy is well defined. This solution is unique when ℓ is finite.

All our algorithms are based on the following order of computations. First, the inverse elements in the left-most column are trivially computed, using the definition of the inverse. Secondly, we compute the first row of the inverse matrix, using the just computed elements of the left-most column. Then we compute the element $c(1,1)$ of the inverse using the elements $c(0,1)$ and $c(1,0)$. Similarly, $c(2,1)$ is computed from $c(1,1)$ and $c(0,1)$ and finally $c(i,j)$ is computed from $c(i-1,j), c(i-2,j)$, and $c(0,j)$, for $i,j = 1, \ldots$.

An important benefit of this order of computation is that for each integer $n \geq 2$, the finite matrix $C(n)$ with elements $c_n(i,j) = c(i,j)$, for $i,j \leq n$, is the inverse of the matrix $B(n)$ with elements $b_n(i,j) = b(i,j)$, for $i,j \leq n$ with the provision that $b_n(n,0)$ and $b_n(n,n)$ are given by the right-hand side of Eq. (5.27) and (5.28).
In this chapter we will consider both a finite and an infinite sized matrix B, and in either case the elements of B can be dependent on i (called the non-homogeneous case) or not (the homogeneous case). All of these four cases require a similar solution procedure, but per subsequent case we exploit the additional structure to enhance the speed and usability of the algorithm.

5.1.3 Related literature

The computation of the inverse of a general matrix is a procedure with a relatively high complexity when the size of the matrix is large. Therefore, there is an extensive literature devoted to studying this problem for matrices with a special structure cf. [25, 26, 65, 73]. Since all related procedures in the literature are applicable only to finite sized matrices, we will assume that ℓ is finite in the discussion in this section. We note that relations in Eq. (5.3), (5.4) and (5.5) make it possible to decompose B as the following sum of two matrices $\widetilde{U}$ and W:

$$B = \widetilde{U} + W, \tag{5.6}$$

where

$$\widetilde{U} = u\delta,$$

with $u = (0, b_1^z, \ldots, b_\ell^z)'$, a column vector and $\delta = (1, 0, \ldots, 0)$, a row vector. We will use this notation throughout the chapter. Matrix W is a tridiagonal matrix, with at least a

negative row sum in the first row. The matrices B and W are invertible, since they can be viewed as transition rate matrices of transient Markov processes. Alternatively, invertibility can be shown by using a similar argument as used in Proposition 3.1(ii) and Lemma 6.1 of this thesis.
Below we describe the main existing approaches in the literature to compute the inverses, where each one utilizes different special properties of W.

First, one can analyse W from a tridiagonal form perspective cf. [73]. A tridiagonal matrix is a band matrix (see [25] and [65]) with a bandwidth of 2, i.e., only the main diagonal and the two adjacent diagonals contain non-zero elements. In [73] a procedure to construct an LU decomposition is provided for tridiagonal matrices. However, multiplying a lower diagonal matrix with an upper diagonal matrix still takes $\mathcal{O}(\ell^3)$ elementary operations. From [50, page 61], one can construct the inverse of B using this constructed inverse of W in the following manner:

$$B^{-1} = W^{-1} + W^{-1}u(a^{-1})\delta W^{-1}, \tag{5.7}$$

where the scalar a is defined as $a = 1 - \delta W^{-1}u$. The matrix multiplications that are used to construct B^{-1} using Eq. (5.7) from the separate terms, are of complexity $\mathcal{O}(\ell^2)$, as is seen from the order of operations below:

$$B^{-1} = W^{-1} + a^{-1}W^{-1}u\delta W^{-1} = W^{-1} + a^{-1}((W^{-1}u)\delta)W^{-1}.$$

Indeed with the above order, each of the multiplication steps requires at most $\mathcal{O}(\ell^2)$ operations. Therefore, if the computation of W^{-1} takes at most $\mathcal{O}(\ell^2)$ operations, then inverting B has the same order of operations, since the other actions are vector times matrix multiplications.

Second, we note that if matrix B is element homogeneous (see Definition 5.1), then matrix W is (almost) of Toeplitz form (cf. [17, 18, 74]). Indeed, only the lower right corner element is inconsistent with this classification, since the matrix is of finite dimension and row ℓ has a zero row sum. We can make approximations to estimate the influence of this disruption. Many of the methods associated with Toeplitz matrices are devoted to computing the solution of the system $Wx = k$ where x is the unknown vector to be computed and k a given vector of size $\ell + 1$. For our purposes, where we need to compute *every* element of W^{-1} explicitly, most of these algorithms are not applicable. However, some of them are super-fast (i.e. of $\mathcal{O}(\ell\ log(\ell))$) in solving the system of equations.

Third, in [71, Section 3.3] a fast algorithm of order $\mathcal{O}(\ell^2)$, for inverting a tridiagonal matrix like W (the "M-matrix" of that paper) is developed. This algorithm computes first the diagonal elements of the inverse matrix using two recursive formulae, one of which starts from the upper left corner of the matrix and the other from the lower right one. Finiteness of the matrix is consequently essential. In addition, this algorithm requires the assumption that $b_i^d b_i^u \neq 0$ for all $i \neq 0$, while the algorithms in this chapter extend to countable matrices and do not require these properties. Further, [71, Table 1, pp. 979] contains a complexity comparison table with several different algorithms from the literature, some of which also

have complexity of order $\mathcal{O}(\ell^2)$, but are not extensible to countable matrices.

Fourth, matrix W satisfies the framework of a Hessenberg matrix, which is studied for example in [54]. That paper constructs an algorithm that computes two vectors x and y such that xy is the upper part of the inverse $W^{-1} = [w_{ij}^{-1}]$, i.e. $w_{ij}^{-1} = x_i y_j$, $j \geq i$. Similarly, for the lower part of W^{-1} the algorithm constructs two other vectors. This algorithm has complexity of order $\mathcal{O}(\ell^2)$ and is applicable to the wider class of finite Hessenberg matrices. However, similarly to the algorithm in [71], it relies on the upper left and lower right elements and finiteness of the matrix is essential.

In conclusion of this literature overview, we note that: a) our method works for non-homogeneous matrices, b) it is applicable to countable matrices, c) it gives an explicit solution independently of the truncation size, as described at the end of Section 5.1.2. Also, we will show later that we can readily extend the method to more general matrices. Therefore, we consider our algorithms to be a significant contribution to the existing literature.

5.2 Efficient computation of the inverse of matrix B

In this section we will describe the four appearances of matrix B and describe their solution procedures. The general idea of the methods is similar, but their are certain differences and simplifications, specific for each case.

In the sequel, for notational simplicity we let C denote the inverse of B, i.e., $CB = BC = I$. The $(i,j)th$ element of C will be denoted by $c(i,j)$, with $i,j = 0,1,2,\ldots,\ell$. We will use the notation B_i and B'_j (respectively C_i and C'_j) to denote the i^{th} row and j^{th} column of matrix B (respectively matrix C), where $i,j = 0,1,\ldots,\ell$.

5.2.1 The non-homogeneous and infinite dimension case

Algorithm 5.1 below is based on the results of Proposition 5.1 to 5.5. This algorithm constructs a matrix C that satisfies $CB = BC = I$, by computing recursively the so far unknown elements of the sequence $C'_0, C_0, C'_1, C_1, C'_2, C_2, \ldots, C'_{n-1}, C_{n-1}, C'_n$, for increasing $n \geq 1$.

The algorithm depends on the computation of a sequence of constants γ_i, where $i = 1,2,\ldots,$ that can be computed if we initially assume that $b_i^d > 0$ for all $i = 0,1,\ldots$. In addition we also require that $b_i^u > 0$: for simplicity we present the algorithm under these assumptions. In Proposition 5.6 we will show how γ_i can be computed when $b_j^d = 0$ for some j. In addition, we will briefly explain how adjustments can be made to compute the inverse of matrices B when $b_j^u = 0$ for some j.

Algorithm 5.1. *Computation of $C := B^{-1}$ for a non-homogeneous count. matrix B*

At stage 0:

a) The column C'_0 (i.e., the column containing elements $c(i,0)$) is computed using Eq. (5.1) of Proposition 5.1.

b) All elements of row C_0 (elements $c(0,j)$) are computed using Eq. (5.8) and (5.9) of Proposition 5.2, where γ_1 is given by Eq. (5.12) of Proposition 5.3.

At stage $i = 1, 2, \ldots, \ell$:

a) For the i^{th} column, C'_i we calculate its remaining elements by Eq. (5.13) of Proposition 5.4, since $c(0,i), \ldots, c(i-1,i)$ have already been computed.

b) For the i^{th} row C_i we calculate its remaining elements by Eq. (5.14) of Proposition 5.5, since $c(i,0), \ldots, c(i,i)$ have already been computed.

In the following propositions we will show that Algorithm 5.1 above is correct.

Proposition 5.1. *The following is true for all $i \geq 0$:*

$$c(i,0) = -1/b_0^d.$$

Proof. By considering the set of equations $BC'_0 = \delta'$ we find:

$$(-b_0^d - b_0^u)c(0,0) + b_0^u c(1,0) = 1,$$
$$b_i^z c(0,0) + b_i^d c(i-1,0) - b_i^w c(i,0) + b_i^u c(i+1,0) = 0, \text{ for } i \geq 1.$$

Using the transition interpretation of Lemma 6.1, C'_0 is constant. Using the definition of b_i^w in Eq. (5.3) it is easy to see that for all $i \geq 0$, $c(i,0) := -1/b_0^d$, is the unique constant solution to the system above. □

Proposition 5.2 below shows that the elements $c(0,j)$ with $j > 0$ depend directly on $c(0,0)$ and not on any other entries of C. Note that $c(0,0)$ is non-zero by its construction in Proposition 5.1.

Proposition 5.2. *Define:*

$$\gamma_j := c(0,j)/c(0,0). \tag{5.8}$$

There exist scalars ρ_j, η_j, with $j = 1, 2, \ldots$ such that all constants γ_j can be recursively computed as a function of γ_1 as follows:

$$\gamma_j = \rho_j \gamma_1 + \eta_j, \tag{5.9}$$

where, under the assumption that $b_j^d > 0$, the constants ρ_j and η_j, $j \geq 3$, are given by:

$$\rho_j = \frac{b_{j-1}^w \rho_{j-1} - b_{j-2}^u \rho_{j-2}}{b_j^d},$$

and

$$\eta_j = \frac{b^w_{j-1}\eta_{j-1} - b^u_{j-2}\eta_{j-2}}{b^d_j},$$

with initial values: $\rho_1 = 1$, $\eta_1 = 0$, $\rho_2 = b^w_1/b^d_2$, $\eta_2 = -b^u_0/b^d_2$.

Proof. Eq. (5.9) follows by the systems of equations $C_0 B = \delta$, without considering the first equality. It is easy to see that every equation has the form below, where $j > 0$:

$$b^u_{j-1}c(0, j-1) - b^w_j c(0, j) + b^d_{j+1}c(0, j+1) = 0. \tag{5.10}$$

This recursive structure allows us to express all elements $c(0, j)$ in terms of their preceding elements. Thus by substitution, we conclude that every element is a product of $c(0, 0)$ and a constant, depending on j that we denote as γ_j.

Eq. (5.9) uses Eq. (5.8) and follows from the observation that:

$$\gamma_2 = \frac{b^w_1\gamma_1 - b^u_0}{b^d_2},$$

and in general for $j \geq 3$:

$$\gamma_j = \frac{b^w_{j-1}\gamma_{j-1} - b^u_{j-2}\gamma_{j-2}}{b^d_j}. \tag{5.11}$$

It is clear that γ_2 is of the form described in the proposition (since $\rho_2 = b^w_1/b^d_2$ and $\eta_2 = b^u_0/b^d_1$) and by substituting, γ_3 has this structure as well. An induction argument completes the proof. □

Proposition 5.3 provides a method to calculate γ_1, using the expressions derived in the above propositions.

Proposition 5.3. *The scalar γ_1 can be calculated algebraically as follows:*

$$\gamma_1 = \frac{b^u_0 - \sum_{j=1}^{\infty} b^z_j \eta_j}{b^d_1 + \sum_{j=1}^{\infty} b^z_j \rho_j}. \tag{5.12}$$

Proof. To verify this expression for γ_1, we consider the equation $C_0 B'_0 = 1$ and use Proposition 5.1 and 5.2. The result follows immediately. □

The next proposition provides a method of computing the under diagonal and diagonal elements ($c(i, j)$, with $j = 1, 2, \ldots$ and $i = j, j+1, \ldots$) of C. Below we let $\delta_{i,j}$ denote a function that takes the value 1 if $i = j$, and 0 otherwise.

Proposition 5.4. *Assume that $b_i^u > 0$, for all $i = 0, 1, \ldots$. The following is true for all elements $c(i, j)$ with $i \geq j \geq 1$.*

$$c(i,j) = \begin{cases} -\gamma_1(1/b_0^d + 1/b_0^u), & \text{for } i = j = 1, \\ (-b_{i-1}^z c(0,j) - b_{i-1}^d c(i-2,j) + b_{i-1}^w c(i-1,j) + \delta_{i-1,j})/b_{i-1}^u, & \text{otherwise.} \end{cases} \tag{5.13}$$

Proof. To compute $c(1, 1)$ we use the equation $B_0 C_1' = 0$, i.e.,

$$-(b_0^u + b_0^d)c(0,1) + b_0^u c(1,1) = 0.$$

The statement is complete since by Proposition 5.2 we have

$$c(0,1) = \gamma_1 c(0,0) = -\gamma_1/b_0^d.$$

To compute the other elements $c(i, j)$ of C specified in the proposition, we use $B_{i-1}C_j' = \delta_{i-1,j}$. Indeed, the product of the $(i-1)^{th}$ row of B (where $i \geq 2$) and the j^{th} column of C (where $j \geq 1$) is the left-hand side of the equation below and completes the proof:

$$b_{i-1}^z c(0,j) + b_{i-1}^d c(i-2,j) - b_{i-1}^w c(i-1,j) + b_{i-1}^u c(i,j) = \delta_{i-1,j}. \qquad \square$$

Proposition 5.5. *The following is true for $c(i, j)$ with $j > i \geq 1$ and $b_j^d > 0$.*

$$c(i,j) = \frac{b_{j-1}^w c(i,j-1) - b_{j-2}^u c(i,j-2) + \delta_{i+1,j}}{b_j^d}. \tag{5.14}$$

Proof. Eq. (5.14) follows from the systems of equations $C_i B = \delta_i$, without considering the first $i - 1$ equalities. The vector δ_i is identical to zero, with a 1 at its i-th entry (note that $\delta_0 = \delta$). Every equation has the form below, where $j > i$:

$$b_{j-2}^u c(i,j-2) - b_{j-1}^w c(i,j-1) + b_j^d c(i,j) = \delta_{i+1,j}. \qquad \square$$

Allowing zeroes above and below the diagonal

Next we discuss how Algorithm 5.1 can be extended to allow $b_i^d = 0$ and $b_j^u = 0$ to be zero for some i and j. The elements b_i^z were already not required to be positive in the above: there is no difference in the computation procedure if they indeed are zero. When $b_i^d = 0$ for some i, Eq. (5.11) can not be used to compute the corresponding γ_i because of the devision by 0 that occurs. However, we next show that under the conditions in Eq. (5.3)-(5.5) the matrix C

is still readily computable: with a small modification, Algorithm 5.1 can still be used. In the proposition below we show how to compute γ_i.

Let $I_0 = \{i_k, k = 1, \ldots, \nu, \text{ such that } b^d_{i_k} = 0\}$ and let $i_0 = 1$ and $i_{\nu+1} = \infty$.

Proposition 5.6. *Assume that I_0 is not empty, then for $k = 0, 1, 2, \ldots, \nu - 1$:*

$$\gamma_{i_k} = \begin{cases} \dfrac{b^w_{i_{k+1}-1}\eta_{i_{k+1}-1} - b^u_{i_{k+1}-2}\eta_{i_{k+1}-2}}{-b^w_{i_{k+1}-1}\rho_{i_{k+1}-1} + b^u_{i_{k+1}-2}\rho_{i_{k+1}-2}}, & \text{if } i_{k+1} \neq i_k + 1, \\ \dfrac{b^u_{i_k-1}\gamma_{i_k-1}}{b^w_{i_k}}, & \text{if } i_{k+1} = i_k + 1. \end{cases} \tag{5.15}$$

For i_ν:

$$\gamma_{i_\nu} = \frac{b^u_0 - \sum_{j=1}^{i_\nu - 1} b^z_j \gamma_i - b^d_1 \gamma_1 - \sum_{j=i_\nu}^{\infty} b^z_j \eta_j}{\sum_{j=i_\nu}^{\infty} b^z_j \rho_j}. \tag{5.16}$$

The remaining components of γ are constructed for $j \in \{i_k + 1, \ldots, i_{k+1} - 1\}$, with $k = 1, 2, \ldots, \nu$ as:

$$\gamma_j = \rho_j \gamma_{i_k} + \eta_j,$$

where the vectors ρ and η are specified below, with $j = 1, 2, \ldots$.

$$\rho_j = \begin{cases} 1, & \text{if } b^d_j = 0 \text{ or if } j = 1, \\ \dfrac{b^w_{j-1}}{b^d_j}, & \text{if } b^d_j \neq 0 \text{ and } b^d_{j-1} = 0, \text{ or if } j = 2, \\ \dfrac{b^w_{j-1}\rho_{j-1} - b^u_{j-2}\rho_{j-2}}{b^d_j}, & \text{if } b^d_j \neq 0 \text{ and } b^d_{j-1} \neq 0, \ j \geq 3, \end{cases}$$

and:

$$\eta_j = \begin{cases} 0, & \text{if } b^d_j = 0 \text{ or if } j = 1, \\ \dfrac{-b^u_{j-2}\gamma_{j-2}}{b^d_j}, & \text{if } b^d_j \neq 0 \text{ and } b^d_{j-1} = 0, \text{ or if } j = 2, \\ \dfrac{b^w_{j-1}\eta_{j-1} - b^u_{j-2}\eta_{j-2}}{b^d_j}, & \text{if } b^d_j \neq 0 \text{ and } b^d_{j-1} \neq 0, \ j \geq 3, \end{cases}$$

where $\gamma_0 = 1$.

Proof. We show how to compute $\gamma_1, \ldots \gamma_{i_1} - 1$. Consider the equations given in Eq. (5.10): only the equation corresponding to $i = i_1 - 1$ changes and reduces to

$$b^u_{i_1-2}\gamma_{i_1-2} = b^w_{i_1-1}\gamma_{i_1-1}. \tag{5.17}$$

Since there is no change in Eq. (5.17) for $i < i_1 - 1$, Eq. (5.9) holds for $i < i_1 - 1$, with

ρ and η defined as above, from which we obtain the equations below (for $i = i_1 - 1$ and $i = i_1 - 2$ respectively):

$$\gamma_i = \rho_i \gamma_1 + \eta_i.$$

Combining the equations above with Eq. (5.9) we find the expression in Eq. (5.15) for γ_1.

Eq. (5.16) follows as before from $C_0 B'_0 = 1$, using the computed γ_i by Eq. (5.15), $i = 1, \ldots, i_\nu - 1$.

The proof for the subsequent values of γ_i goes analogously, by considering the next segment $\{\gamma_{i_1}, \ldots, \gamma_{i_2 - 1}\}$. More specifically, we substitute i_1 by i_k in Eq. (5.17). The proof is complete noting that the infinity tail of values from i_ν on corresponds to the remaining infinite set, originally considered in Proposition 5.1 to 5.3. □

In the sequel when we refer to Algorithm 5.1 we include the possible modification applied per Proposition 5.6.

The adjustments that need to be made to Eq. (5.13) to compute the remaining elements when $b^u_{i-1} = 0$, are very similar to the procedure described in Proposition 5.6. Instead of using $B_{i-1} C'_j = \delta_{i-1,j}$ we use $B_i C'_j = \delta_{i,j}$ to express the elements $c(i + k, j)$ in terms of $c(i, j)$. We normalize to compute $c(i, j)$. Thus this procedure happens horizontally for upper diagonal elements when $b^d_i = 0$, and vertically for under diagonal elements when $b^u_j = 0$.

Note that when $b^u_i = 0$ then automatically: $c(k, l) = 0$ for $1 \leq k \leq i$ and $l \geq i + 1$.

The case of multiple columns with many non-zero elements

One way to enlarge the class of matrices for which the method above can be applied is by allowing other columns to have more than three elements non-zero, keeping the diagonal elements negative: even the entire column can be non-zero, as long as the row sum equals zero for each row. For example, when matrix B has the first two columns of this type it will have the following form:

$$B = \begin{bmatrix} -b^d_0 - b^u_0 & b^u_0 + b^{z_2}_0 & 0 & 0 & 0 & \cdots \\ b^d_1 + b^{z_1}_1 & -b^w_1 + b^{z_2}_1 & b^u_1 & 0 & 0 & \cdots \\ b^{z_1}_2 & b^d_2 + b^{z_2}_2 & -b^w_2 & b^u_2 & 0 & \ddots \\ b^{z_1}_3 & b^{z_2}_3 & b^d_3 & -b^w_3 & b^u_3 & \ddots \\ b^{z_1}_4 & b^{z_2}_4 & 0 & b^d_4 & -b^w_4 & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Note that the columns do not need to be linearly dependent. As before, the row sum is zero for every row, except the first, i.e. for all $i \geq 1$:

$$b_i^{z_1} + b_i^{z_2} + b_i^d + b_i^u = b_i^w.$$

Algorithm 5.1 can still be applied, when only a few adjustments are made, we will leave the details to the reader. The main difference with the original algorithm is that in this multi-column case, we express all elements of the top row in terms of the top row element corresponding to the non-zero column with the highest index.

As long as the number of non-zero columns remains small, the complexity of the algorithm remains the same, only the normalization takes some extra steps. When the number of non-zero columns becomes large, the algorithm loses its computational advantage and perhaps other methods might be preferable.

Another approach is to use the Toeplitz equation, cf. Eq. (5.7) and [50], but only when the second column is a multiple of the first column. This is a necessary condition to use this equation, and not required for the procedure above.

5.2.2 The non-homogeneous and finite dimension case

A finite representation of the matrix, $\ell + 1 < \infty$, may occur due to modelling or due to truncation. No adjustments are needed in the formulation of Algorithm 5.1, the only difference is the altered condition at the boundary:

$$b_\ell^w = b_\ell^d + b_\ell^z,$$

i.e., the row sum of the final row is zero. The formulas of Proposition 5.2 - 5.6 are adjusted so that the summations up to infinity, are now up to ℓ where ℓ is finite. All other equations remain the same.
For increasing truncation levels, the elements of the finite inverse matrices converge to those of the infinite sized matrix constructed in the previous section that is a solution of $CB = BC = I$.

5.2.3 The homogeneous and infinite dimension case

In this paragraph we define the following subclass of matrices B that not only possesses the structure described in Eq. (5.2) but also the additional structure of the definition below.

Definition 5.1. B is called *element homogeneous* when the following relations hold:

$$b_i^d = b^d, b_i^u = b^u, b_i^z = b^z \text{ for all } i \geq 1.$$

We note that when B is *element homogeneous* then necessarily $b_i^w = b^w$, for all $i \geq 1$.

In the rest of this section we denote the constant γ_1 as γ. We assume that none of the elements given are zero to avoid trivialities and more tedious notation. First we derive the following extended result regarding the scalar γ, then we will give the algorithm, and then prove it is correct.

Proposition 5.7. *When B has a homogeneous structure and its size ℓ is countable, the following is true for all $j \geq 2$:*

$$c(i,j) = \gamma^{j-i} c(i,i), \text{ for } j > i \geq 0,$$

where

$$\gamma = \frac{b^w - \sqrt{(b^w)^2 - 4b^u b^d}}{2b^d}. \tag{5.18}$$

Proof. The proof follows by considering the systems of equations $C_i B'_j = 0$ for $j = i+1, i+2, \ldots$ All equalities have the form displayed below.

$$b^u c(i,j-1) - b^w c(i,j) + b^d c(i,j+1) = 0.$$

Because of the homogeneous structure and since we have assumed that the return probability to state zero is 1 (when we consider B as a transition rate matrix), it is clear that the corresponding elements of row C_i have a product form with a constant factor $\gamma < 1$. Thus, the following relation of γ and element $c(i,j)$ holds:

$$b^u c(i,j) - b^w \gamma c(i,j) + b^d \gamma^2 c(i,j) = 0, \tag{5.19}$$

thus in other words:

$$c(i,j) = \gamma^{j-i} c(i,i). \tag{5.20}$$

When solving for γ, Eq. (5.19) simplifies to $b^d\gamma^2 - b^w\gamma + b^u = 0$. This quadratic equation has a single solution γ provided in Eq. (5.18) greater than 0 since $\sqrt{(b^w)^2 - 4b^u b^d} < \sqrt{(b^w)^2} = b^w$. Second, the described choice of γ is smaller than 1 since:

$$\begin{aligned}
\gamma = \frac{b^w - \sqrt{(b^w)^2 - 4b^u b^d}}{2b^d} &< \frac{b^w - \sqrt{(b^w)^2 - 4b^u b^d - 4b^z b^d}}{2b^d} \\
&= \frac{b^w - \sqrt{(b^w)^2 + 4(b^d)^2 - 4b^w b^d}}{2b^d} \\
&= \frac{b^w - \sqrt{(b^w - 2b^d)^2}}{2b^d} \\
&= \frac{2b^d}{2b^d} = 1.
\end{aligned}$$

□

When B is element homogeneous, we do not need Proposition 5.3 to compute C, since $\gamma_1 = \gamma$ and computable by Proposition 5.7. Proposition 5.4 remains the same when B is element homogeneous. In addition, the expression for the computation of the under diagonal elements can also be simplified. Using both expressions, we can even directly express the diagonal elements in terms of its predecessor.

The above leads to the following considerable simplification of Algorithm 5.1. We denote:

$$\psi := \frac{b^w - \sqrt{(b^w)^2 - 4b^u b^d}}{2b^u} = \gamma b^d / b^u, \tag{5.21}$$

where ψ is between 0 and 1, by the same derivation used for γ.

Algorithm 5.2. *Computation of $C := B^{-1}$ for an element hom., countable matrix B.*

At stage 1:

a) Calculate γ using Eq. (5.18).

b) Calculate ψ using Eq. (5.21).

c) The elements of row C_0 are computed by Eq. (5.20) in Proposition 5.7.

At stage $i = 1, 2, \ldots, \ell$:

a) The diagonal element $c(i, i)$ is computed by Eq. (5.24).

b) The elements of row C_i are computed using Eq. (5.20) of Proposition 5.7.

c) The elements of column C'_i are computed using Eq. (5.23) of Proposition 5.8.

To summarize, the elements $c(i, j)$, with $i, j \geq 0$ can be calculated as follows:

$$c(i,j) = \begin{cases} \dfrac{-1}{b^d} & \text{if } j = i = 0, \\ \dfrac{-1 + b^u((1-\psi)c(0,i-1) + \psi c(i-1,i-1))}{b^w - b^d\gamma} & \text{if } j = i \geq 1, \\ \gamma^{j-i} c(i,i) & \text{if } j \geq i+1, \\ (c(j,j) - c(0,j))\psi^{i-j} + c(0,j) & \text{if } j \leq i-1. \end{cases} \tag{5.22}$$

For a feasible order of computations to compute all elements, we refer to the order in Algorithm 5.2.

Remark 5.1. It is possible that b_0^d is different from $b^d := b_1^d = b_2^d = \ldots$. Algorithm 5.2 remains the same, only in the computation of $c(0, 0)$ in Eq. (5.22), b^d is substituted by b_0^d.

Below we will prove that Algorithm 5.2 is correct. As a part of this algorithm, the under diagonal elements of column C'_i can be expressed in terms of the first element $c(i,0)$ of this column and its diagonal element $c(i,i)$. This calculation is given in the proposition below.

Proposition 5.8. *The under diagonal elements of C ($i > j \geq i$) can be calculated as follows:*

$$c(i,j) = (c(j,j) - c(0,j))\psi^{i-j} + c(0,j). \tag{5.23}$$

Proof. We consider the equation $B_i C'_j = 0$, with $i \geq j+1$:

$$b^z c(0,j) + b^d c(i-1,j) - b^w c(i,j) + b^u c(i+1,j) = 0.$$

Secondly, we define $d(i,j) := c(i,j) - c(0,j)$ for all $i \geq j$. Then it is easy to see that the above is equal to:

$$b^d d(i-1,j) - b^w d(i,j) + b^u d(i+1,j) = 0.$$

Similarly to the proof of Proposition 5.7, we can show that the elements $d(i,j)$ have a product form in i with a constant factor ψ, for all $i \geq j+1$:

$$b^d d(i-1,j) - b^w d(i,j) + b^u d(i+1,j) = 0,$$

thus in other words:

$$d(i,j) = \psi^{i-j} d(j,j).$$

When solving for ψ, this quadratic equation has a unique solution between 0 and 1, given by Eq. (5.21). Next, substituting $d(i,j) = c(i,j) - c(0,j)$ completes the proof:

$$d(i,j) = c(i,j) - c(0,j) = \psi^{i-j}(c(j,j) - c(0,j)). \qquad \square$$

Since ψ and γ are expressed explicitly, we are able to calculate the diagonal elements using Eq. (5.24) below. This is true by isolating $c(i,i)$ in the equation $C_i B'_i = 1$:

$$c(i,i) = (-1 + b^u((1-\psi)c(0,i-1) + \psi c(i-1,i-1)))/(b^w - b^d\gamma). \tag{5.24}$$

In the next proposition we derive an interesting result regarding the convergence of the diagonal elements, where we abbreviate $D := (b^w)^2 - 4b^u b^d$. Note that $D > 0$.

Proposition 5.9. *The diagonal elements of C converge as follows:*

$$\lim_{i\to\infty} c(i,i) = \frac{-1}{\sqrt{D}}.$$

Proof. First, it is important to note that by definition, $b^d\gamma^2 - b^w\gamma + b^u = 0$, and thus:

$$b^d\gamma^2 = b^w\gamma - b^u. \tag{5.25}$$

We will use the above statement throughout the proof. We want to show that the diagonal elements $c(i,i)$ converge to a (negative) constant, for all $i = 1, 2, \ldots$. Therefore we consider the difference $v(i)$ between two subsequent diagonal elements, defined as: $v(i) := c(i,i) - c(i-1,i-1)$.

In addition, we use an alternative expression for $c(i,i)$, true by the equality $B_{i-1}C_i' = 0$, given in Eq. (5.26) below:

$$c(i,i) = (-b^z c(0,i) - b^d\gamma^2 c(i-2,i-2) + b^w\gamma c(i-1,i-1))/b^u. \tag{5.26}$$

We find a recursive expression for $v(i)$, using Eq. (5.25) and (5.26):

$$\begin{aligned}
-b^z c(0,i) &= b^u c(i,i) - b^w\gamma c(i-1,i-1) + b^d\gamma^2 c(i-2,i-2) \\
&= b^u\left(c(i,i) - c(i-2,i-2)\right) - b^w\gamma\left(c(i-1,i-1) - c(i-2,i-2)\right) \\
&= b^u\left(v(i) - v(i-1)\right) - b^w\gamma v(i-1),
\end{aligned}$$

and thus since $\gamma^i b^z/b^d = -b^z c(0,i)$ we get $v(i) = \gamma^i b^z/(b^d b^u) + \frac{b^w\gamma - b^u}{b^u} v(i-1)$. In the sequel we use that $\lim_{i\to\infty}\gamma^i = 0$ and that by Eq. (5.25): $\frac{b^w\gamma - b^u}{b^u} = b^d\gamma^2/b^u = \psi\gamma$.
Note that both ψ and γ are between 0 and 1, thus their product is in the same interval. We can use the above to prove the convergence of $c(i,i)$, by considering the absolute value $|v(i)|$.

$$\begin{aligned}
\lim_{i\to\infty}|v(i)| &= \lim_{i\to\infty}|\gamma^i b^z/(b^d b^u) + \frac{b^w\gamma - b^u}{b^u} v(i-1)| \\
&= \frac{b^d\gamma^2}{b^u}\lim_{i\to\infty}|\gamma^{i-2} b^z/(b^d)^2 + v(i-1)| \\
&\leq \psi\gamma\lim_{i\to\infty}|\gamma^{i-2} b^z/(b^d)^2| + \lim_{i\to\infty}|v(i-1)| \\
&= \psi\gamma\lim_{i\to\infty}|v(i-1)| = \psi\gamma\lim_{i\to\infty}|v(i)|,
\end{aligned}$$

and thus zero. Since the series converges, we can express its limit and complete the proof.

$$\begin{aligned}
\lim_{i\to\infty} c(i,i) &= \lim_{i\to\infty}(-1 + (b^u + b^z - b^u\psi)\gamma^i c(0,0) + b^d\gamma c(i-1,i-1))/(b^w - b^u\psi) \\
&= -1/(b^w - b^u\psi) + b^d\gamma/(b^w - b^u\psi)\lim_{i\to\infty} c(i,i) \\
&= -1/(b^w - b^d\gamma - b^u\psi) = \frac{-1}{\sqrt{D}}.
\end{aligned}$$

□

5.2.4 The homogeneous and finite dimension case

In this section we consider the element homogeneous and finite representation of matrix B. To be able to use Algorithm 5.2 and thus to compute the row and column independent scalars

γ and ψ, it is necessary that the final row satisfies the condition described in the proposition below. When truncating a homogeneous, infinite sized matrix, the truncation should satisfy this structure to use Algorithm 5.2.

Proposition 5.10. *Consider a finite and element homogeneous matrix B of size $(\ell+1)\times(\ell+1)$ satisfying Eq. (5.2). If:*

$$B(\ell,\ell) = -b^u/\gamma, \tag{5.27}$$

and

$$B(\ell,1) = b^u/\gamma - b^d, \tag{5.28}$$

then $c(i,j)$ can be computed per Eq. (5.22) where γ is defined as in Eq. (5.18) and ψ as in Eq. (5.21).

Proof. When considering $C_i B'_\ell = 0$ with $0 \le i \le \ell-1$ we find, under the assumption of the proposition, that:

$$b^u c(i,\ell-1) - b^u/\gamma c(i,\ell) = 0,$$

and thus that $c(i,\ell) = \gamma c(i,\ell-1)$. All upper diagonal elements $c(i,j)$ with $j > i$ can now be expressed in terms of γ and its predecessor analogously to Proposition 5.7.

When considering $B_\ell C'_j = 0$ with $0 \le j \le \ell-1$ and the observation that $b^u/\gamma = b^d/\psi$ we find that:

$$(b^d/\psi - b^d)c(0,j) + b^d c(\ell-1,j) - b^d/\psi c(\ell,j) = 0.$$

Rewriting the equation above shows that:

$$b^d(c(\ell,j) - c(0,j)) = \psi b^d(c(\ell-1,j) - c(0,j)).$$

All lower diagonal elements can thus be expressed in terms of ψ, its predecessor and $c(0,j)$, analogous to Proposition 5.8. □

When an alternative truncation of B is chosen or provided, γ is row dependent and ψ is column dependent and therefore the solution procedure of a non-homogeneous finite matrix B should be used.

It is clear that the computation of each element only requires a linear number of steps. Therefore the complexity of Algorithms 5.1 and 5.2 is $\mathcal{O}(\ell^2)$.

5.3 The eigenvalues of matrix B when it is finite

In this section we investigate properties for the eigenvalues $\{\lambda_i^b\}_{i=1,\ldots,\ell+1}$ of matrix B, when it is a finite matrix. The results in this section apply to both the homogeneous and non-homogeneous cases and even to general matrices.

We will make use of the structure of matrix $W = [w(i,j)]_{i,j=0}^{\ell}$, introduced in Eq. (5.6). Matrix B can be written as displayed in Eq. (5.6), as $B = \widetilde{U} + W$. In this section we take the assumption below to be satisfied.

Assumption 5.1. For all $i = 1, \ldots, \ell$ the following condition holds.

$$w(i, i-1)w(i-1, i) > 0.$$

We note that the above assumption implies the irreducibility property of W, when it is the rate matrix of a birth-and-death process.
We start with the following well-known lemma, proved in several papers. One of the more complicated proofs of this lemma uses Sturm sequences cf. [46], other proofs involve diagonalisation, or follow the line of Meurant cf. [75].

Lemma 5.1. *Consider the finite tridiagonal matrix W, (where $W = B - \widetilde{U}$) introduced in Eq. (5.6), satisfying Assumption 5.1. Then its eigenvalues $\{\lambda_i\}_{i=1,\ldots,\ell+1}$ are negative, real and distinct.*

Using the results in [16, Remark 3.4 and Remark 3.6], we know that for arbitrarily small $||u||$ the eigenvalues of B are arbitrarily close to those of W, the eigenvalues of which are negative, real and distinct. Now, if an eigenvalue of B has a non-zero imaginary part, then its complex conjugate is also an eigenvalue and the norm of these two eigenvalues is equal. This will lead to a contradiction to the fact that for each eigenvalue of W there is an eigenvalue of B arbitrarily close to it. For general vectors u, such that B has the prescribed properties of Eq. (5.3)-(5.5), numerical experiments show that B has real eigenvalues (not necessarily distinct) as well, but we have no proof to confirm this.
In the remainder of this section we develop a method to transform eigenvalues and eigenvectors of a given matrix into those of another matrix. First we will provide a schematic overview of how this method will work for matrices W and B, before we go to the rigorous proofs for general matrices in the next section. Then we will specialize to the matrices W and B and use the discussed approach to derive sufficient conditions for B to have real eigenvalues.

5.3.1 Overview of the method

Let $v_1, \ldots, v_{\ell+1}$ be the eigenvectors of W, corresponding to its eigenvalues denoted as $\lambda_1, \ldots, \lambda_{\ell+1}$. The eigenvectors form a basis of $\mathbb{R}^{\ell+1}$ and therefore u can be written as a linear combination of the eigenvectors: $u = \sum_{i=1}^{\ell+1} n_i v_i$. Let $M = \#\{i \,|\, n_i \neq 0\}$, i.e. M is the number of eigenvectors of W contributing to u.
By renormalization, reordering and denoting the contributing eigenvectors by $v_1, \ldots, v_M$, we can write $u = \sum_{i=1}^{M} v_i$. If $M = 1$, then the eigenvalues and eigenvectors of B can be

explicitly expressed in terms of the eigenvectors and eigenvalues of W and Proposition 5.11 below implies that all eigenvalues of B are real.

Suppose that $M > 1$. In this case we would preferably apply the following iterative procedure to construct B from W: first add v_1 to the first column of W, resulting in $B_1 = v_1\delta + W$, the eigenvalues and eigenvectors of which can be computed from Proposition 5.11. One would have hoped to continue this approach, i.e., add the remaining eigenvectors of W one by one to form B_2, etc. and finally arrive at a matrix $B_M = B$.

However, v_i, $i = 2, \ldots, M$ are *not* eigenvectors of the new matrix B_1 anymore, hence we constructed a new procedure to express $\sum_{i=2}^{M} v_i = \sum_{k=1}^{M-1} v_k^1$ as a linear combination of the eigenvectors $\{v_k^1\}_{k=1}^{M-1}$ of B_1. Next add one of these eigenvectors,e.g. v_1^1, to B_1. The problem with this approach is that we have not been able to prove that this scheme yields the matrix B neither in a finite number of steps nor even in the limit. Therefore we suggest another procedure below.

The following alternative approach is always successful to reconstruct B from W in M steps, but it requires extra conditions to guarantee that the eigenvalues of B are real. Instead of adding v_1 to the first column of W, we add $\alpha_1 v_1$ for a yet to be defined constant α_1. This yields the new matrix $B_1 = \alpha_1 v_1 \delta + W$ with eigenvectors

$$v_1^1, v_2^1, \ldots, v_{\ell+1}^1, \text{ where } v_1^1 = v_1.$$

The remaining part of the sum is equal to $u - \alpha_1 v_1 = \sum_{i=2}^{M} v_i + (1 - \alpha_1)v_1$. Suppose α_1 can be chosen in such a way that it is real and that:

$$u - \alpha_1 v_1 = (1 - \alpha_1)v_1 + \sum_{i=2}^{M} v_i = \sum_{i=2}^{M} v_i^1,$$

then we have expressed the 'remaining' part of u as a linear combination of $M - 1$ eigenvectors of the new matrix B_1. If one iterates this procedure, then after M iterations the matrix B_M is equal to B which has thus been reconstructed from W. This is a finite procedure; after each step, the remaining portion is a linear combination of strictly less eigenvectors. It will appear that this constant α_1 always exists since the initial matrix W is tridiagonal and has a basis of eigenvectors. However this constant may be a complex number with a non-zero imaginary part. The conditions provided in Proposition 5.14 are sufficient to guarantee that α_1 and all subsequent constants of this scheme are real numbers.

To make the above scheme rigorous, we will first state and prove Proposition 5.11. We will then present an algorithm for adding a column times row vector to a general $n \times n$ matrix A, provided that the column vector is written as a linear combination of the eigenvectors of A. Then we will specialize to the matrix W, the column vector u and the row vector δ and deduce conditions under which the eigenvalues of $B_1, B_2, \ldots, B_M$, remain real in each iteration step of the algorithm.

5.3.2 Analysis of the eigenvalues of a general matrix

Let A be an $n \times n$ matrix (real or complex) with eigenvalues $\xi_1, \ldots, \xi_n$ (counting multiplicaties) and distinct (right) eigenvectors $z_1, \ldots, z_m$, $m \leq n$, enumerated such that eigenvalue ξ_i corresponds to eigenvector z_i, $i \leq m$. Let $r \in \mathbb{R}^n$ be a row vector.

Proposition 5.11. *Let $\alpha_1 \neq 0$ be a given scalar (real or complex), such that $rz_i \neq 0$ implies $\xi_i - \xi_1 - \alpha_1 r z_1 \neq 0$, for $i = 2, \ldots, n$. Put $A_1 := A + \alpha_1 z_1 r$. Then the following are true:*

a) A_1 has eigenvalues $\{\xi_1 + \alpha_1 r z_1, \xi_2, \ldots, \xi_n\}$.

b) For $i = 2, \ldots, m$ and if $rz_i \neq 0$, let:

$$x_i = \frac{\alpha_1 r z_i}{\xi_i - \xi_1 - \alpha_1 r z_1}, \tag{5.29}$$

otherwise put $x_i = 0$. Then the vectors $\{z_1, z_2 + x_2 z_1, \ldots, z_m + x_m z_1\}$ are eigenvectors of A_1 corresponding to eigenvalues $\xi_1 + \alpha_1 r z_1, \xi_2, \ldots, \xi_m$.

Proof. The proof of part (a) on the eigenvalues is given in [76, Exercise 7.1.17]. For completeness we outline its main steps below.

In the derivation that follows, we use the equality below:

$$z_1 = (A - \lambda I)^{-1}(A - \lambda I)z_1 = (A - \lambda I)^{-1}(Az_1 - \lambda z_1) = (A - \lambda I)^{-1}(\xi_1 - \lambda)z_1,$$

where I is the $n \times n$ identity matrix. The characteristic polynomial for A_1 is given by

$$\begin{aligned} |A_1 - \lambda I| = |A + \alpha_1 z_1 r - \lambda I| &= |A - \lambda I + \alpha_1 z_1 r| = \\ &= |A - \lambda I|(1 + \alpha_1 r (A - \lambda I)^{-1} z_1) \\ &= \left(\pm \prod_{i=1}^{n} (\xi_i - \lambda) \right) \left(1 + \frac{\alpha_1 r z_1}{\xi_1 - \lambda} \right) \\ &= \left(\pm \prod_{i=2}^{n} (\xi_i - \lambda) \right) (\xi_1 + \alpha_1 r z_1 - \lambda), \end{aligned} \tag{5.30}$$

where in the third equality of Eq. (5.30) we have used the formula in [76, p.475 'Rank one updates']. The roots of this polynomial are the eigenvalues of A_1 and they are equal to:

$$\{\xi_1 + \alpha_1 r z_1, \xi_2, \ldots, \xi_n\}.$$

Note that for part (a) of the proposition, it is not necessary to assume that $rz_i \neq 0$ implies $\xi_i - \xi_1 - \alpha_1 r z_1 \neq 0$.

For part (b), we first show that z_1 is an eigenvector to eigenvalue $\xi_1' := \xi_1 + \alpha_1 r z_1$. Indeed,

$$\begin{aligned}(A_1 - \xi_1' I)z_1 &= \left(A + \alpha_1 z_1 r - (\xi_1 + \alpha_1 r z_1)I\right) z_1 \\ &= (A - \xi_1 I)z_1 + \alpha_1 z_1 r z_1 - \alpha_1 z_1 r z_1 \\ &= 0.\end{aligned}$$

Next, $x_i z_1 + z_i$ is eigenvector to eigenvalue ξ_i, $2 \leq i \leq m$, since:

$$\begin{aligned}(A_1 - \xi_i I)(x_i z_1 + z_i) &= \left(A + \alpha_1 z_1 r - \xi_i I\right)(x_i z_1 + z_i) \\ &= (A - \xi_i I)(x_i z_1 + z_i) + \alpha_1 z_1 (x_i r z_1 + r z_i) \\ &= x_i A z_1 - x_i \xi_i z_1 + \alpha_1 x_i (r z_1) z_1 + \alpha_1 (r z_i) z_1 \\ &= x_i(\xi_1 - \xi_i + \alpha_1 r z_1) z_1 + \alpha_1 (r z_i) z_1 = 0,\end{aligned} \tag{5.31}$$

by our definition of x_i. Clearly $\{z_1\} \cup \{z_i + x_i z_1\}_{i=2}^m$ is a linearly independent set. □

We state the following corollary that can be directly derived from Propostion 5.11.

Corollary 5.1. *If the eigenvalues of any matrix A are real and α_1 is real, then the eigenvalues of A_1 are real.*

Note that the construction of x_i for a specific i, in Eq. (5.29) is not possible if both $\xi_i = \xi_1 + \alpha_1 r z_1$, and $r z_i \neq 0$, since then the implication in the theorem is not true. In the proof of Theorem 5.1 we show that this will not happen for a suitable choice of α_1. Next, we write the (column) vector $u \in \mathbb{R}^n$ such that $u = \sum_{i=1}^m n_i z_i$ is in the span of the eigenvectors of A. Again, let $M = \#\{i \,|\, n_i \neq 0\}$. Then after renormalization, reordering, and renaming the eigenvectors, we can write

$$u := \sum_{i=1}^{M} z_i. \tag{5.32}$$

Without loss of generality, all eigenvectors z_i in the above expression correspond to distinct eigenvalues of A.

The next algorithm presents an iterative and finite procedure to compute the eigenvalues of $A + ur$ from the eigenvalues of A, when u is a linear combination of eigenvectors, as described in Eq. (5.32). It formalizes and generalizes the procedure we have described in Section 5.3.1 for reconstructing the eigenvalues and eigenvectors of B from W, and uses Proposition 5.11. Recall that ξ_i, $i = 1, \ldots, m$ are all distinct.

Algorithm 5.3. *Computation of eigenvalues and eigenvectors of $A + ur$ from those of A*

Step 0, Initialisation:
Put $A_0 = A$, $z_0^0 = 0$, $z_i^0 = z_i$ for $i = 1, \ldots, M$, $k = 1$,
$\xi_i^0 = \xi_i$, for $i = 1, \ldots, n$, $Z_0 = \{z_{M+1}, \ldots, z_m\}$.

Step 1:

- If $k = M$ or if $rz_i^{k-1} = 0$ for $i = k+1 \dots M$, put $\alpha_k = 1$;
  else, determine a solution α_k to the following equation in variable α:

$$(1-\alpha)\prod_{i=k+1}^{M}(\xi_i^{k-1} - \xi_k^{k-1} - \alpha r z_k^{k-1})\mathbf{1}_{\{rz_i^{k-1}\neq 0\}} = \alpha \sum_{j=k+1}^{M} rz_j^{k-1} \prod_{\substack{i=k+1 \\ i\neq j}}^{M}(\xi_i^{k-1} - \xi_k^{k-1} - \alpha r z_k^{k-1})\mathbf{1}_{\{rz_i^{k-1}\neq 0\}}. \tag{5.33}$$

- Put $z_k^k = z_k^{k-1}, \quad A_k = A_{k-1} + \alpha_k z_k^k r, \quad \xi_k^k = \xi_k^{k-1} + \alpha_k r z_k^k, \quad \xi_i^k = \xi_i^{k-1}, i \neq k.$
- For $i \in \{k+1, \dots, M\}$ and for all $\{i | z_i^{k-1} \in Z_{k-1}$ and $(\xi_i^k - \xi_k^k \neq 0 \vee (\xi_i^k - \xi_k^k = 0 \wedge rz_i^{k-1} = 0))\}$:

$$x_i^k = \frac{\alpha_k r z_i^{k-1}}{\xi_i^k - \xi_k^k}\mathbf{1}_{\{rz_i^{k-1}\neq 0\}}$$
$$z_i^k = z_i^{k-1} + x_i^k z_k^{k-1}. \tag{5.34}$$

- Put $Z_k := \{z_i^k | z_i^{k-1} \in Z_{k-1}\} \cup \{z_k^k\}$.

Step 2: If $k = M$ stop, else $k := k+1$, goto Step 1.

Before we prove the algorithm, we would like to emphasize some of its properties below.

- Some eigenvectors belonging to eigenvalues ξ_i with $i = M+1, \dots m$ are not updated. This lack of update happens in step k if $\xi_k^k = \xi_i^k$. The eigenvalue will thus have an higher algebraic multiplicity in A_k compared to A_{k-1}, and the algorithm above will possibly not provide all the eigenvectors. The algebraic multiplicity may not be equal to the geometric multiplicity.
- Since the number of distinct eigenvalues may possibly increase, new eigenvectors may emerge. In the algorithm, we only give a recipe on how the eigenvectors of A are being transformed.
- Note further that after performing the algorithm, we can subsequently add another column times row matrix, provided the new column is a linear combination of some of the eigenvectors of the matrix $A + ur$.

Theorem 5.1. *Assume that $u = \sum_{i=1}^{M} z_i$ and that the corresponding eigenvalues $\xi_1, \dots, \xi_M$ of A are all distinct. Algorithm 5.3 terminates in precisely M steps and outputs the matrix $A_M = A + ur$, the eigenvalues of $A + ur$ and eigenvectors of $A + ur$ that are in the set Z_M.*

Proof. We start with the proof for the first step, $k = 1$.

If $rz_i^0 = 0$ for all $i = 1, \dots M$, then $\alpha_1 = 1$. We get $A_1 = A + z_1^0 r = A + z_1 r$. Proposition 5.11 gives that $x_i = 0$, hence the eigenvectors $z_i^1 = z_i^0$ with $i = 2, \dots, M$. We get $u^1 = u^0 - z_1^0 = u - z_1 = \sum_{i=2}^M z_i = \sum_{i=2}^M z_i^1$. The assertion on the other eigenvectors z_i^0 of A, when $i > M$, follows directly from Proposition 5.11 by inspecting Eq. (5.31) and by noting that z_i^0 cannot be transformed into an eigenvector of A_1 if both $rz_i^0 = rz_i \neq 0$ and $\xi_i^0 - \xi_1^0 - rz_1^0 = \xi_i^1 - \xi_1^1 = 0$. This case is excluded by the design of the algorithm.

Suppose that $rz_i^0 \neq 0$ for at least one $i = 2, \dots M$. Then Eq. (5.33) is a polynomial of degree at least 1, and hence it has a solution. In order to apply Proposition 5.11 we have to check for $2 \leq i \leq M$ that $rz_i^0 \neq 0$ implies that $\xi_i^0 - \xi_1^0 - \alpha_1 rz_1^0 = \xi_i - \xi_1 - \alpha_1 rz_1 \neq 0$.

Assume the contrary, namely that $\xi_i^0 - \xi_1^0 - \alpha_1 rz_1 = 0$ for some i. Then the left-hand side of Eq. (5.33) equals 0. The only term on the right-hand side that possibly does not equal 0, is the term corresponding to i, i.e.:

$$rz_i \prod_{\substack{2 \leq j \leq M \\ j \neq i}} (\xi_j - \xi_1 - \alpha_1 rz_1).$$

This expression then necessarily equals 0 as well, for α_1 to be a solution. However, since $\xi_i^0 - \xi_1^0 - \alpha_1 rz_1 = 0$ and all eigenvalues ξ_i, $1 < i \leq M$ are distinct, no element of this product can be zero and we arrive at a contradiction.

Hence, the result of Proposition 5.11 can be applied. We get that $A_1 = A + \alpha_1 rz_1^0 = A + \alpha_1 rz_1$ and:

$$u^1 = u^0 - \alpha_1 z_1^0 = u - \alpha_1 z_1 = \sum_{i=1}^M z_i - \alpha_1 z_1 = (1 - \alpha_1) z_1 + \sum_{i=2}^M z_i.$$

We note that $z_i^1 = z_i^0 = z_i$ if $rz_i^0 = rz_i = 0$ when $2 \leq i \leq M$. Hence:

$$u^1 = (1 - \alpha_1) z_1^0 + \sum_{i=2}^M z_i^1 \mathbf{1}_{\{rz_i^0 = 0\}} + \sum_{i=2}^M z_i^0 \mathbf{1}_{\{rz_i^0 \neq 0\}}. \tag{5.35}$$

The product in the left-hand side of Eq. (5.33) does not equal 0 for $\alpha = \alpha_1$, and so we may divide both sides of this equation by it, yielding the following for expression $1 - \alpha_1$:

$$\begin{aligned} \sum_{i>1}^M \alpha_1 rz_i^0 \frac{\prod_{\substack{2 \leq j \leq M \\ j \neq i}} (\xi_j^0 - \xi_1^0 - \alpha r z_1^0) \mathbf{1}_{\{\alpha_1 rz_i^0 \neq 0\}}}{\prod_{2 \leq j \leq M} (\xi_j^0 - \xi_1^0 - \alpha r z_1^0) \mathbf{1}_{\{rz_j^0 \neq 0\}}} &= \sum_{i=2}^M \frac{rz_i^0}{\xi_i^0 - \xi_1^0 - \alpha r z_1^0} \mathbf{1}_{\{rz_i^0 \neq 0\}} \\ &= \sum_{i=2}^M x_i^1 \mathbf{1}_{\{rz_i^0 \neq 0\}}. \end{aligned}$$

A combination with Eq. (5.35) gives

$$u^1 = \sum_{i=2}^{M} z_i^1 \mathbf{1}_{\{rz_i^0=0\}} + \sum_{i=2}^{M} z_i^0 \mathbf{1}_{\{rz_i^0\neq 0\}} + \sum_{i=2}^{M} x_i^1 z_1^0 \mathbf{1}_{\{rz_i^0\neq 0\}} = \sum_{i=2}^{M} z_i^1.$$

Notice that all eigenvectors z_i^0 have been 'transformed' to eigenvectors z_i^1 of A_1 for $i = 2, \ldots, M$. Hence u^1 is a linear combination of eigenvectors $z_2^1, \ldots, z_M^1$ of A_1, where the total number of contributing eigenvectors has decreased by 1. Also the corresponding eigenvalues $\xi_i^1 = \xi_i$, $i = 2, \ldots, M$ are still distinct.

Therefore, the input parameters satisfy the assumptions allowing to carry out the next iteration of adding a (fraction of) z_2^1 to the first column of A_1 and applying Proposition 5.11. This is justified by carrying out exactly the same analysis as we did in the above for the case $k = 1$.
Iterating and checking the conditions of Proposition 5.11 at each step yields that: $u^k = u - \sum_{i\leq k} z_i^{i-1} = \sum_{i>k} z_i^k$ and

$$A_k = A + \sum_{i\leq k} z_i^{i-1} r = A + (u - u^k) r.$$

As soon as $k = M$, $u^{M-1} = z_M^{M-1}$ is added to the first column of A_{M-1}. Hence $A_M = A_{M-1} + z_M^{M-1} = A + (u - u^{M-1})r + u^{M-1}r = A + ur$ and we are done. The update of u tot u^k is done implicitly by the update of eigenvectors.

The update of the eigenvectors belonging to the class Z^{k-1} is done analogously to the other eigenvectors. However, there is no sufficient condition to ensure that $\xi_i^k - \xi_i^k - \xi_k^k = 0$ for a corresponding $z_i^{k-1} \in Z^{k-1}$. Therefore we only update when this difference is not zero, and do not update otherwise. The eigenvectors that can not be readily computed are not a part of the linear combination that is used to compute r. Therefore, their lack of update does not influence the update of the eigenvalues and that of other eigenvalues. □

Remark 5.2. An alternative way to find the eigenvalues of B is by applying Eq. (5.30) for all eigenvectors and eigenvalues simultaneously, i.e., by substituting αz_1 by $u = z_1 + \ldots + z_M$. This shows that eigenvalues corresponding to eigenvectors of A that do not contribute to u, do not change by adding ur. More precisely, all the eigenvalues of B are the roots of the following polynomial:

$$\prod_{i=M+1}^{n} (\xi_i - \lambda) \prod_{i=1}^{M} \mathbf{1}_{\{rz_i=0\}} (\xi_i - \lambda) \Big(\prod_{i=1}^{M} \mathbf{1}_{\{rz_i\neq 0\}} (\xi_i - \lambda) + \sum_{j=1}^{M} rz_j \prod_{\substack{i=1 \\ i\neq j}}^{M} (\xi_i - \lambda) \Big).$$

This equation seems hard to analyse directly, but will be simple if u is a linear combination of a few eigenvectors of A.

5.3.3 Specialize to $A = W$

In this section we specialize the results of the previous section to the case where A equals tridiagonal matrix W (cf. Eq. (5.6)) and with $r = \delta = (1, 0, 0, \ldots)$ and $B = u\delta + W$, as in the previous sections. We will provide sufficient conditions for the eigenvalues of B to be real. First, since W has distinct eigenvalues (Lemma 5.1) its eigenvectors span the entire space. Therefore, u is always a linear combination of (a subset) of its eigenvectors. Secondly, it is easy to check that $rz_i^k = \delta v_i^k = v_i^k(0) \neq 0$, for all k and i; the Hessenberg structure of $W + u\delta$ implies that if the first entry of an eigenvector is zero, then the remaining elements are zero as well.

Thus the assumptions to use Proposition 5.11 hold and we can use this proposition to derive the eigenvalues of B from those of W. We can also monitor the change of eigenvectors, using Algorithm 5.3.

We state and prove a known result regarding the eigenvalues of B.

Lemma 5.2. *All eigenvalues of B have a negative real part.*

Proof. Since B is invertible (see previous section), all its eigenvalues are non-zero. By the Gershgorin circle theorem we know that every eigenvalue of B lies within at least one of the Gershgorin discs $D(b(i,i), R_i) = \{z \in \mathbb{C} : ||z - b(i,i)|| \leq R_i\}$, where $R_i = \sum_{j \neq i} b(i,j) \leq -b(i,i)$, for all $i \in \{1, \ldots, \ell + 1\}$, with $||\ldots||$ the Euclidian norm. By assumption, the diagonal elements $b(i,i) < 0$, and the result follows. □

Next, we derive the following corollary from Algorithm 5.3 and Theorem 5.1.

Corollary 5.2. *Let $u = \sum_{i=1}^{M} v_i$. Then the eigenvalues λ_i^b of B are:*

$$\lambda_i^b = \begin{cases} \lambda_i + \alpha_i v_i^{i-1}(0), & \text{if } i = 1, \ldots, M, \\ \lambda_i, & \text{otherwise.} \end{cases}$$

Here λ_i are the distinct eigenvalues of W, eigenvector v_i^{i-1} is constructed recursively via Algorithm 5.3 and α_i is the solution of Eq. (5.33).

Note that the eigenvalues of B are not necessarily distinct, despite the fact that those of W are.

We derive from Corollary 5.1 and 5.2 that if all α_i are real, then the eigenvalues of B are real. Below we will provide a sufficient condition for this statement to be true. Without loss of generality, we reorder and rename the eigenvectors and eigenvalues such that $\lambda_1 < \lambda_2 < \ldots < \lambda_M < 0$.

We emphasize again that α_i only depends on the first entry of the eigenvector v_i, with $i = 1, 2, \ldots, M$, since $r = \delta$. In the sequel, we simplify Eq. (5.33) for this specific case by substituting for $v_i(0)$ and rewrite it to make it a function of α, i.e., $f_i(\alpha)$, $i = 1, 2, \ldots M-1$:

$$f_i(\alpha) = 1 - \alpha - \alpha \sum_{j=i+1}^{M} \frac{v_j^{i-1}(0)}{\lambda_j - \lambda_i - \alpha v_i^{i-1}(0)}.$$

In the sequel we are interested in whether the function $f_i(\cdot)$ has at least one real root or no real roots. If it has a real root, then this root will be α_i to construct the eigenvalues of B. In the proposition below we identify a sufficient condition for $f_i(\cdot)$ to have a real root. To do so, we define $S_{i,j} := (\lambda_j - \lambda_i)/v_i^{i-1}(0)$ as the j^{th} singularity of $f_i(\cdot)$, where $j > i$. Since by assumption, $\lambda_{j+1} > \lambda_j$ for all j, this sequence is strictly positive and increasing (negative and decreasing) in j when $v_i^{i-1}(0) > 0 (< 0)$. As is reasoned before, $v_i^{i-1}(0) \neq 0$.

Proposition 5.12. *For any $i = 1, 2, \ldots, M-1$, suppose that $v_i^{i-1}(0)$ and $v_{i+1}^{i-1}(0)$ are both positive. Then $f_i(\cdot)$ has at least one real positive root.*

Proof. By the assumption in this proposition, the quotient $\frac{\alpha v_{i+1}^{i-1}(0)}{\lambda_{i+1} - \lambda_i - \alpha v_i^{i-1}(0)} > 0$ for $0 < \alpha < S_{i,i+1}$: within this interval $\lambda_{i+1} - \lambda_i > \alpha v_i^{i-1}(0)$. This quotient is the dominant term near the singularity $S_{i,i+1}$ and thus $\lim_{\alpha \uparrow S_{i,i+1}} f_i(\alpha) = -\infty$. Since $f_i(0) = 1$, there is at least one positive real root on the interval $(0, S_{i,i+1})$, on which $f_i(\alpha)$ is continuous. □

Proposition 5.13. *If $v_i^{i-1}(0) < 0$, then $f_i(\cdot)$ has at least one positive root, for any $i = 1, 2, \ldots, M-1$.*

Proof. Recall that in this case $S_{i,i+1}$ is the largest singularity and smaller than zero since $v_i^{i-1}(0) < 0$. Therefore, $f_i(\cdot)$ is continuous on $[0, \infty)$. Since $\lim_{y \to \infty} f_i(y) = -\infty$ and $f_i(0) = 1$ the function $f_i(\cdot)$ has at least one positive real root. □

In the proposition below we will provide a condition on the eigenvectors of W, that ensures that all scalars α_i are real.

Proposition 5.14. *All scalars α_i are real when $v_j(0) < 0$ for all $j \leq I$ and $v_j(0) > 0$ for all $j > I$ with $I \in \{0, 1, 2, \ldots, M\}$.*

Proof. With this configuration of first elements of the eigenvectors of W, either $v_1(0) < 0$ or $v_1(0), v_2(0) > 0$. We know by Proposition 5.12 and Proposition 5.13 that in both cases, $f_0(\cdot)$ has at least one real root, and moreover that this root is positive. Let α_1 be this smallest positive real root of $f_1(\cdot)$.

Eq. (5.34) implies the following recursion for the first element of $v_j^1,\ j = 2, 3, \ldots, M$:

$$v_j^1(0) = v_j(0)\left(1 + \frac{\alpha_1 v_1(0)}{\lambda_j - \lambda_1 - \alpha_1 v_1(0)}\right) = v_j(0)\left(\frac{\lambda_j - \lambda_1}{\lambda_j - \lambda_1 - \alpha_1 v_1(0)}\right).$$

If $v_1(0) < 0$, then $\frac{\lambda_j - \lambda_1}{\lambda_j - \lambda_1 - \alpha_1 v_1(0)} > 0$, since $\alpha_1 > 0$ and $\lambda_j > \lambda_1$.

If $v_1(0) > 0$, then $\frac{\lambda_j - \lambda_1}{\lambda_j - \lambda_1 - \alpha_1 v_1(0)} > 1$, since $\alpha_1 \in (0, S_{1,2})$.

In both cases $v_j^1(0)$ has the same sign as $v_j(0)$ for all $j = 2, 3, \ldots, M$. The provided structure of these elements ensures that $f_2(\cdot)$ has at least one positive real root α_2, since now $v_2^1(0) < 0$ or $v_2^1(0), v_3^1(0) > 0$. We can repeat this argument and construct all positive real roots α_i for all i. □

We conjecture that the sequence $v_i(0)$ satisfies the condition given in Proposition 5.14 and thus that B has real eigenvalues. Numerical examples confirm this conjecture, but we leave its proof as an open problem.

5.4 Applications

Matrix B, discussed in the previous sections occurs in many models, particularly when considering a 2 dimensional Markov process and using the successive lumping approach as done in Chapter 3. In order to find the stationary distribution one can use this approach to compute the rate matrix R. In many cases calculating this matrix using successive lumping requires the inversion of a matrix with the structure of B. Examples can be found in queueing (an $M/Er/c$ queue with batch service, discussed as well in Section 3.7.1), reliability systems and inventory models (an inventory model with batch arrivals and random lead time, discussed in Section 3.8). Specific about these systems and the complete procedure to compute the steady state distribution can be found in Chapter 4. Below we will discuss other (classes of) applications in which this type of matrix arises naturally or can be constructed.

5.4.1 A general non-transient Markov process

Let Q be the transition rate matrix of a general, non-transient Markov process and let B be the matrix with elements:

$$b(i,j) = \begin{cases} q(0,0) - 1 & \text{if } (i,j) = (0,0), \\ q(i,j) & \text{otherwise,} \end{cases}$$

i.e., $B = Q - \delta' \cdot \delta$. The solution to the steady state equations:

$$\pi Q = 0, \text{ and } \pi 1 = 1,$$

is given by: $\pi = \dfrac{\delta B^{-1}}{\delta B^{-1}\mathbf{1}}$. Algorithm 5.1 can be used to obtain the solution, subject to the modifications described in Section 5.2.1. Similarly, in the case of a homogeneous process, Algorithm 5.2 can be applied.

5.4.2 A birth-and-death process with an absorbing state

In this section, we consider a special case of the structure described above that has applications to a birth-and-death process with abandonments. These models occur for example in diffusion processes (cf. [55]) and in randomly changing environments (cf. [44]). We assume that in every state there is an (equal) positive rate to leave the system and go to an absorbing state (state 0). The remaining states form a birth-and-death process. This process is element homogeneous according to its description in Definition 5.1, but with the difference that $b_0^u = 0$ and $b_1^d = 0$. Therefore the transition matrix has the following form:

$$B = \begin{bmatrix} -b^d & 0 & 0 & 0 & \cdots \\ b^z & -b^z - b^u & b^u & 0 & \cdots \\ b^z & b^d & -b^w & b^u & \ddots \\ b^z & 0 & b^d & -b^w & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

The procedure to find the inverse of B is described in Algorithm 5.4 below. This is a special case of Algorithm 5.2, that requires element homogeneity and it takes into account that $b_0^u = 0$. A general procedure to handle $b_i^u = 0$ has also been discussed in this chapter, however we can simplify those results for this specific case.

Algorithm 5.4.

At stage 1:

a) Calculate γ using Eq. (5.18) and ψ using Eq. (5.21).

b) The 0^{th} column of C is computed using Eq. (5.1) of Proposition 5.1.

c) The remaining elements of the 0^{th} row of C are 0, as is described in Proposition 5.15.

d) Calculate element $c(1,1)$ by Proposition 5.16.

e) The remaining elements ($j \geq 2$) of this row are: $c(1,j) = \gamma^{j-1} c(1,1)$.

At stage $i = 2, 3, \ldots, \ell + 1$:

a) The diagonal element $c(i, i)$ is computed by Eq. (5.24).

b) The elements of C_i, the i^{th} row of C to the right of the diagonal are computed using Proposition 5.7 and in particular Eq. (5.20) in that proposition.

c) The under diagonal elements of C'_i, the i^{th} column of C are computed using Eq. (5.23) of Proposition 5.8.

Most of the steps above are similar to those in Algorithm 5.2, and the two Propositions below justify the remaining steps.

Proposition 5.15. *The elements $c(0, j)$ are equal to zero for all $j \geq 1$.*

Proof. This result follows immediately from the fact that all elements are non-positive and by considering $C_0 B'_0 = b^d(-1/b^d) + b^z \sum_{j=1}^{\infty} c(0, j) = 1$. □

Proposition 5.16. *The element $c(1, 1)$ can be calculated as follows:*

$$c(1, 1) = \frac{1}{-b^z - b^u + b^d \gamma}.$$

Proof. We consider the equation $C_1 B'_1 = 1$. This gives us:

$$(-b^z - b^u)c(1, 1) + b^d c(1, 2) = 1.$$

Because $c(1, 2) = \gamma c(1, 1)$ we derive:

$$c(1, 1) = \frac{1}{(-b^z - b^u) + b^d \gamma}.$$ □

5.4.3 Value functions

In many models, e.g., to compute the value function of the expected $\alpha-$discounted cost associated with a continuous time Markov process, one needs to solve equations of the following type:

$$\alpha V = c + QV \tag{5.36}$$

where c is the cost rate function defined on the state space and V the unknown value vector to be determined as a solution of Eq. (5.36). We refer to [89, 91] for more background on these models. The solution to this equation is $V = -Q_\alpha^{-1} c$, where $Q_\alpha = (Q - \alpha I)$ is a rate matrix of a transient Markov process where in addition to the rates specified by Q one has introduced an additional event of 'exiting' or 'halting' the process at a rate α. When Q has a tridiagonal form, i.e. it is the transition rate matrix of a birth-and-death process, we can use the algorithm of the previous section to compute this inverse of matrix Q_α.

CHAPTER 6

Level product form QSF processes

This chapter has been submitted as: *Level product form QSF processes and an analysis of queues with Coxian interarrival distribution*, cf. [S6].

6.1 Introduction to Chapter 6

A Quasi-Skipfree (QSF) process is a continuous time Markov process $X = \{X_t\}_{t\geq 0}$, on a two-dimensional state space $\boldsymbol{S} = \{(m,i)|\, m \in \mathbf{Z},\, i \in \{0,\ldots,\ell_m\}\}$, where m denotes the 'level' of the state and i denotes the 'phase' within the level. Additionally, the jump rates are not allowed to cross more than one level in one direction, i.e. either the *downward* direction or *upward* direction. This framework is the natural extension of the embedded GI/M/1 and M/G/1-queues, and it has interesting structural properties in common with these processes.

Neuts [80] investigates the embedded $GI/M/1$-queue as a skip-free to the right process. The matrix-geometric method for computing the stationary distribution of homogeneous Quasi-Birth-Death (QBD) processes has been applied in [80] and [70]. A homogeneous QBD process is a special case of homogeneous QSF processes, where the transition probabilities are not allowed to cross more than one level in both directions. In his book, Neuts discusses some examples of homogeneous QBD processes, e.g., the $M/PH/1$-queue and the $PH/M/c$-queue. He shows that the stationary distribution of these queues can be expressed in terms of a rate matrix.

In Chapter 2 we have introduced a new procedure to compute the invariant measure for a class of Markov chains. This procedure is called the successive lumping method. Further, an explicit solution and bounds for the steady state probabilities for the class of ergodic QSF processes that possess the successive lumpability property have been derived in Chapter 3. Ramaswami and Latouche [85] discuss QBD processes with a special structure, namely where the upward or the downward transitions rates form a row times column matrix. In the latter case the rate matrix can be computed explicitly. In the former case they show that the stationary distribution has a level product form, as is the case with the embedded $GI/M/1$-queue.

Regarding rate matrix analysis, Ramaswami and Lucantoni, [86], exploit the structural properties of the $G|PH|1$-queue, and extend the numerical feasibility of the matrix geometric approach to a wide set of problems by developing efficient schemes to compute R.

One of the main assumptions of [85] is that the transition rates are bounded as a function of the states. In the particular case of a QBD process, where the upward transition rates form a matrix with one non-zero row, the rate matrix cannot be computed explicitly. However, we will show how the stationary distribution of a higher level can be explicitly expressed in terms of the stationary distribution of lower levels, under an additional invertibility condition. Moreover, we will show that for this specific type of non-homogeneous QSF processes it is possible to derive a level product form solution.

We further study a phase-type inter-arrival, batch service queue, denoted $PH/M^Y/1$-queue, as a particular example of a QSF process, where the upward transition rates form a matrix with one non-zero row. We will consider a special case of a phase type distribution, a Coxian distribution (cf. [28]). Herein customers go through a maximum of k exponentially distributed phases, and after each phase the customer can enter the system. We will denote a Coxian distribution with k phases by $Cox(k)$. Since the Coxian distributions are dense in the set of non-negative distribution functions, we can approximate those by a Coxian distribution by using for example an expectation-maximization algorithm (see [23]). Therefore the results presented in this paper can be useful for various queues with different inter-arrival distributions.

We show that the stationary distribution of the phases within a level has a product form as well, if the phase-type inter-arrival distribution is a Coxian distribution. In Chapter 3 the successive lumpable structure is specified for QSF processes, but we did not investigate the level product form that is derived in this paper for QSF problems with an unbounded number of levels to the left. The parameter of the product form is the solution to a fixpoint equation, that can be numerically approximated efficiently.

We conjecture that allowing for c servers will yield a product form solution based on c factors, similarly to the results presented by Adan et al [10, Theorem 4.1]. Actually, we can also handle the $Cox(k)/E_l/1$ by taking the expected workload instead of the number of customers as the batch service distribution.

This paper is organised as follows. In Section 6.2 we show that the stationary distribution of the levels has a product form when choosing the downward matrix D as a multiplication of a column vector c and a row vector r. In Section 6.3 we assume that there exists an exit state per level. This is equivalent to the existence of an entrance state per level for a revised level partition of the state space that keeps the QSF property intact. This means that the process is successively lumpable with respect to the new partition. This is used to derive an expression for the rate matrix with respect to the original partition in Section 6.3.1. This derivation is justified by the invertibility of the generator of a transient Markov chain in the Appendix. Unfortunately, no explicit expression for the rate matrix can be derived. However, we can

explicitly determine a matrix that expresses the stationary distribution of a lower level in terms of the stationary distribution of the higher level.

In Section 6.4 we analyze the non-homogeneous $Cox(k)/M^Y/1$-queue, and find the parameter of the product form, as well as ergodicity properties. In the remainder of this section we specialise this queue to a queue with homogeneous rates and an infinite number of phases. In Section 6.5 we derive monotonicity properties of the stationary mean number of customers in the queue and their mean sojourn time for fixed mean inter-arrival times. To conclude, we show that the stationary distribution of the E_k/M/1-queue (a special case of the $Cox(k)/M^Y/1$ where the number of phases is fixed and the batch size of service is 1) converges monotonically to the stationary distribution of the D/M/1-queue.

6.2 The model and basic properties

In this section we introduce the notation and derive two initial properties for the stationary distribution of homogeneous QSF processes in Lemma 6.1 and Theorem 6.1. Note that by the homogeneity assumption the number of phases per level of the QSF process is constant, say equal to $\ell + 1$, for $\ell \geq 0$. Without loss of generality, we may assume the QSF process to be skip-free to the left. We will also assume that the levels are bounded to the right, since otherwise the stationary distribution cannot exist, as is shown later in this section. Without loss of generality we may then assume that they are non-negative, i.e. $m \leq 0$. Then the infinitesimal generator Q or q-matrix of Eq. (3.1) (see also [19]) takes the form:

$$Q = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & D & W & U^1 & U^2 & U^3 & U^{4'} \\ \cdots & 0 & D & W & U^1 & U^2 & U^{3'} \\ \cdots & 0 & 0 & D & W & U^1 & U^{2'} \\ \cdots & 0 & 0 & 0 & D & W & U^{1'} \\ \cdots & 0 & 0 & 0 & 0 & D & W' \end{bmatrix}, \tag{6.1}$$

where the $(\ell+1)\times(\ell+1)$ sub-matrices U^s $(s = 1, 2, \ldots)$, W, and D represent the transition rates to the s-th higher level, the same level, and the next lower level respectively. Elements of these matrices are u^s_{ij}, w_{ij}, and d_{ij} where u^s_{ij} is the $((m, i), (m + s, j))$ element from the matrix U^s, w_{ij} is the $((m, i), (m, j))$ element from the matrix W, and d_{ij} is the $((m, i), (m-1, j))$ element from the matrix D, for any $i, j \in \{0, \ldots, \ell\}$. The matrices $U^{k'}$ are such that the row sum of Q is zero. For example these matrices can be $U^{k'} = \sum_{i=k}^{\infty} U^i$ and $W' = W + \sum_{i=1}^{\infty} U^i$.

Throughout the paper we assume that the q-matrix Q is irreducible, conservative, stable, and non-explosive. Additionally, we assume that the jump rates are allowed to be unbounded as a function of the state and that X is the minimal process. It is convenient to denote the levels

by L_m, so that $L_m = \{(m,i)\,|\,i \in \{0,\ldots,\ell\}\}$ and $\boldsymbol{S} = \cup_m L_m$, where $-\infty \leq m \leq 0$. In view of the structure of Q given in Eq. (6.1), jumps can only take place to levels L_k for $k \geq m-1$, given that the current level is L_m.

Since the levels are mutually exclusive, they form a partition of the state space. We will introduce some more notation. In accordance with Chapter 3, for any fixed m we define the sub-level set of L_m to be the set of states $\underset{\sim}{L}_m = \cup_{k\leq m} L_k$, while the set $\widetilde{L}_m = \cup_{k\geq m} L_k$ is the super-level set of L_m. Then clearly $\{\underset{\sim}{L}_{m-1}, \widetilde{L}_m\}$ is a partition of $\boldsymbol{S}$ for each m.

Suppose that the QSF process is positive recurrent. Let π denote the (unique) stationary distribution. We denote the m-level sub-vector of π by π_m. The stationary distributions corresponding to $\underset{\sim}{L}_{m-1}$ and $\widetilde{L}_m$ will be denoted by $\underset{\sim}{\pi}_{m-1}$ and $\widetilde{\pi}_m$ respectively. Then by a standard taboo decomposition, as discussed in Chapter 3, we can express $\underset{\sim}{\pi}_{m-1}$ in terms of $\widetilde{\pi}_m$ and the expected amount of time spent in the states of $\underset{\sim}{L}_{m-1}$.

We denote by ${}_m\underset{\sim}{\tau}_{(k,i)(l,j)}$ the expected amount of time spent in state (l,j) without passing through states in the super-level set $\widetilde{L}_m$, given that the system starts in state (k,i), where $(k,i),(l,j) \in \underset{\sim}{L}_{m-1}$.

Further, denote by ${}_m\underset{\sim}{\mathsf{T}}$ the $\left|\underset{\sim}{L}_{m-1}\right| \times \left|\underset{\sim}{L}_{m-1}\right|$ matrix with elements ${}_m\underset{\sim}{\tau}_{(k,i)(l,j)}$. Write:

$$\underline{D} = \begin{bmatrix} 0 & \ldots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & 0 \\ 0 & \ldots & 0 & D \end{bmatrix},$$

where 0 stands for an $(\ell+1)\times(\ell+1)$ zero matrix and $\underline{D}$ has dimension $|\widetilde{L}_m| \times |\underset{\sim}{L}_{m-1}|$. Then by the skip free property to the left

$$\underset{\sim}{\pi}_{m-1} = \widetilde{\pi}_m \, \underline{D} \, {}_m\underset{\sim}{\mathsf{T}}. \tag{6.2}$$

We will next express ${}_m\underset{\sim}{\mathsf{T}}$ in terms of Q. The elements of the q-matrix Q are denoted by $q_{(k,i)(l,j)}$, where $(k,i),(l,j) \in \boldsymbol{S}$. Denote $q_{(k,i)} = -\,q_{(k,i)(k,i)}$, the parameter of the exponential sojourn time in state (k,i). Further denote by ${}_m\underset{\sim}{Q}$ the taboo-generator with taboo set $\widetilde{L}_m$, restricted to the states of $\underset{\sim}{L}_{m-1}$, with elements

$${}_m\underset{\sim}{q}_{(k,i)(l,j)} = \begin{cases} q_{(k,i)(l,j)}, & (k,i),(l,j) \in \underset{\sim}{L}_{m-1}, \\ 0, & \text{otherwise}, \end{cases}$$

and ${}_m\underset{\sim}{q}_{(k,i)} = -\,{}_m\underset{\sim}{q}_{(k,i)(k,i)}$.

The taboo is therefore imposed at the time of the first jump out of a state of the QSF process on $\underset{\sim}{L}_{m-1}$. By assumed irreducibility, ${}_m\underset{\sim}{Q}$ is the q-matrix of a transient, non-conservative and minimal Markov process on the state space $\underset{\sim}{L}_{m-1}$. Since the number of levels is unbounded to the left, ${}_m\underset{\sim}{Q}$ is of infinite dimension.

Denoting the corresponding (minimal) transition function by ${}_mP_t = {}_mp_{t,(k,i)(l,j)}$ where all elements $(k,i),(l,j) \in \underset{\sim}{L}_{m-1}$. It follows that

$${}_m\mathcal{T}_{(k,i)(l,j)} = \int_0^\infty {}_mp_{t,(k,i)(l,j)}dt < \infty.$$

We can state the following lemma.

Lemma 6.1. *The matrix $-{}_m\mathsf{T}$ is the maximum negative solution to the equation:*

$$A\,{}_mQ = {}_mQA = -\mathsf{I}_m.$$

In the sequel we write $-{}_m\mathsf{T} = ({}_mQ)^{-1}$.

Proof. By virtue of Anderson [19] (Theorem 2.2.2) the minimal transition function ${}_mP_t$ satisfies the Kolmogorov forward and backward equations and it is the unique solution on $\boldsymbol{S}$.

Define a scalar $a > 0$. Then ${}_mR(a) = \int_0^\infty e^{-at}\,{}_mP_t\,dt$ is the associated resolvent with elements ${}_mr_{(k,i)(l,j)}(a)$, where $(k,i),(l,j) \in \underset{\sim}{L}_{m-1}$. By virtue of Anderson [19], Propositions (2.1.1) and (2.1.2), it holds that

$$a\,{}_mR(a) = \mathsf{I}_m + {}_mQ\,{}_mR(a) = \mathsf{I}_m + {}_mR(a)\,{}_mQ.$$

Hence, by virtue of the backward equation, for $(k,i),(l,j) \in \underset{\sim}{L}_{m-1}$ it holds that

$$\begin{aligned} a\,{}_mr_{(k,i)(l,j)}(a) &= \delta_{(k,i)(l,j)} + {}_mq_{(k,i)(k,i)}\,{}_mr_{(k,i)(l,j)}(a) \\ &\quad + \sum_{(k',i')\neq(k,i)} {}_mq_{(k,i)(k',i')}\,{}_mr_{(k',i')(l,j)}(a), \end{aligned} \tag{6.3}$$

where Kronecker delta $\delta_{(k,i)(l,j)} = \begin{cases} 1, & (k,i) = (l,j) \\ 0, & \text{otherwise.} \end{cases}$

Observe that ${}_mr_{(k)(l,j)}(a) \uparrow {}_m\mathcal{T}_{(k,i)(l,j)}$ for $a \downarrow 0$. Using that the summation in Eq. (6.3) contains only non-negative terms, we can now take the limit $a \downarrow 0$ in Eq. (6.3) and use the monotone convergence theorem to obtain

$$\delta_{(k,i)(l,j)} + {}_mq_{(k,i)(k,i)}\,{}_m\mathcal{T}_{(k,i)(l,j)} + \sum_{(k',i')\neq(k,i)} {}_mq_{(k,i)(k',i')}\,{}_m\mathcal{T}_{(k',i')(l,j)} = 0. \tag{6.4}$$

Eq. (6.4) is equivalent to

$$\mathsf{I}_{m,(k,i)(l,j)} + ({}_mQ\,{}_m\mathsf{T})_{(k,i)(l,j)} = 0.$$

This yields that ${}_mQ\,{}_m\mathsf{T} = -\mathsf{I}_m$.

The relation ${}_m\mathsf{T}\,{}_mQ = -\mathsf{I}_m$ is analogously proved, by using the forward equation for the resolvent. The maximality follows by using the analogy of Eq. (6.52) expressing ${}_m\mathsf{T}$ in terms of the jump chain, and by an iteration argument. □

As a consequence of the result above we obtain:

$$\underset{\sim}{\pi}_{m-1} = -\widetilde{\pi}_m \underline{D}\,{}_mQ^{-1}.$$

Define T to be the $(\ell+1)\times(\ell+1)$ subblock of $(-\,{}_mQ)^{-1}$ corresponding to the states in L_{m-1}. Note that without further restrictions on D it is unfortunately not possible to calculate it independently of the rates of states in $\underset{\sim}{L}_{m-1}$. From Eq. (6.2) it follows that

$$\pi_{m-1} = \pi_m B, \quad B = DT,$$

where B_{ij} represents the expected local time spent in $(m-1,j)$, before absorption into $\widetilde{L}_m$, given that the process starts in (m,i).

By homogeneity, we can recursively show that

$$\pi_m = \pi_0 B^{-m}, \quad \text{for} \quad m \le 0. \tag{6.5}$$

Now the existence of the stationary distribution implies that

$$\sum_{m\le 0} B^{-m}\mathbf{1}_{\ell+1} = (I-B)^{-1}\mathbf{1}_{\ell+1} < \infty,$$

with I the $(\ell+1)\times(\ell+1)$ identity matrix and $\mathbf{1}_{\ell+1}$ the $\ell+1$-dimensional vector consisting of ones.

The next lemma shows that it is not possible to have an unbounded state space to the right and a stationary distribution for a homogeneous QSF process.

Lemma 6.2. *Consider a homogeneous QSF process where the number of levels is unbounded in the negative direction. In order for the process to be positive recurrent, the number of levels has to be bounded in the positive direction.*

Proof. Suppose that the levels are not bounded to the right. Were the stationary distribution to exist, the same reasoning as in the above (Eq. (6.5)) would yield that $\pi_m = \pi_n B^{-m+n}$, $m \le n$. Analogously, by homogeneity it would follow that $\pi_n = \pi_{(n,0)}\mathbf{a}$, with $\mathbf{a}$ an $(\ell+1)$-dimensional positive column vector, independent of n (see also the proof of Theorem 6.1 below). Then

$$\sum_{m\le n,j} \pi_{(m,j)} = \pi_{(n,0)}\mathbf{a}^T(I-B)^{-1}\mathbf{1}_{\ell+1}.$$

Taking the limit $n\to\infty$, on the one hand yields that $\sum_{m\le n,j}\pi_{(m,j)} \to 1$. On the other hand, the right-hand side of the above equation is the product of two scalars, $\pi_{(n,0)}$ and

$\mathbf{a}^T(I-B)^{-1}\mathbf{1}_{\ell+1}$. Since $\lim_{n\to\infty}\pi_{(n,0)}=0$, the product equals 0 as well, in the limit $n\to\infty$. This is a contradiction. □

6.2.1 Special choice of D

Imposing extra structure on D implies a result on the structure of the stationary distribution, that is described below. This extra structure covers the case when $D=c\cdot r$, where c is a column vector and r is a row vector, and $\cdot$ denotes matrix product (of potentially non-square matrices). In other words, the rows of D are proportional to each other. Let the i^{th} column of T be denoted by the vector $t_i=(t_{0i},t_{1i},\ldots,t_{\ell i})^T$, for $i\in\{0,\ldots,\ell\}$.

Then B has the following form:

$$B=DT=\begin{bmatrix} c_0r\cdot t_0 & c_0r\cdot t_1 & \cdots & c_0r\cdot t_\ell \\ c_1r\cdot t_0 & c_1r\cdot t_1 & \cdots & c_1r\cdot t_\ell \\ \vdots & \vdots & \ddots & \vdots \\ c_\ell r\cdot t_0 & c_\ell r\cdot t_1 & \cdots & c_\ell r\cdot t_\ell \end{bmatrix}=c\cdot\hat{t},$$

where $\hat{t}=(r\cdot t_0,r\cdot t_1,\ldots,r\cdot t_\ell)$ and

$$B^n=(c\cdot\hat{t})^n=c\cdot\left((\hat{t}\cdot c)^{n-1}\hat{t}\right)=\left(\sum_{i=0}^{\ell}c_ir\cdot t_i\right)^{n-1}c\cdot\hat{t},\quad n\geq 1.$$

Notice that matrix B has only one non-zero eigenvalue $\gamma=\hat{t}\cdot c$ with left eigenvector $\hat{t}$ of multiplicity 1, provided that $\hat{t}\cdot c$ is finite. Combination with Eq. (6.5) implies that the stationary distribution of the levels has a product form. The following theorem holds.

Theorem 6.1. *Assume that D has a column times row structure as is described above.*
If $\gamma<1$ and $\ell<\infty$, then the Markov process X is positive recurrent.
If the Markov process X is positive recurrent, then $\gamma<1$. In either case:

$$\pi_{(m,j)}=\gamma^{-(m+1)}\sum_{i=0}^{\ell}\pi_{(0,i)}c_i\,\hat{t}_j. \tag{6.6}$$

Proof. Suppose that X is positive recurrent. By Eq. (6.5), the stationary distribution necessarily has the form of Eq. (6.6). Taking the summation over the levels m in Eq. (6.6) yields a finite expression. This implies that $\sum_{m\leq M}\gamma^{-(m+1)}<\infty$ for any level M. Hence, $\gamma<1$ necessarily.

By separating state $(0,0)$ from level 0, one can compute the expected sojourn time in the states of $\boldsymbol{S}\setminus\{(0,0)\}$ by inverting the (negative of the) finite rate matrix restricted to this set, analogously to the above derivations. Define $x_{(0,i)}$ to be equal to an unknown constant $x_{(0,0)}$

times the expected sojourn time in state $(0, i)$ given a start in state $(0, 0)$ before absorption into $(0, 0)$, for all $i > 0$. Then define $x_{(m,j)}$ from Eq. (6.6) by substituting $\pi_{(0,i)}$ by $x_{(0,i)}$ in the right-hand side. Note that $x_{(0,0)}$ is the only unknown constant. Since $\gamma < 1$, they can be normalized to yield a (unique) probability distribution on $\boldsymbol{S}$. □

It is still hard to compute B directly, therefore we introduce even more structure on D in the next section. More precisely, we assume that c has only one non-zero component. Without loss of generality, we may assume that c_0 is the only non-zero element of c.

6.3 Exit states and successive lumpability

Our main assumption concerns the presence of *exit states*, which are defined below.

Definition 6.1. Let $M \subset \boldsymbol{S}$. Then $(m, i) \in M$ is an exit state for M, if

i) $\sum_{(k',j') \notin M} q_{(k,j)(k',j')} = 0$ for all $(k, j) \in M$ with $(k, j) \neq (m, i)$.

ii) $\sum_{(k',j') \notin M} q_{(m,i)(k',j')} > 0$.

For the remainder of this section we make the following assumption, which implies that for all m, level $m - 1$ can only be reached directly from level m through state $(m, 0)$. In other words, $(m, 0)$ is an exit state to level $m - 1$.

Assumption 6.1. State $(m, 0)$ is an exit state for superset $\widetilde{L}_m$, for all $m = \ldots, -1, 0$.

6.3.1 Stationary distribution

Under assumption 6.1, the sub-matrix D contains only one non-zero row. This implies that matrix B has only one non-zero row (the first) as well. Then by virtue of Theorem 6.1:

$$\pi_{(m-1,j)} = \pi_{(m,0)} b_{0j}, \tag{6.7}$$

where $b_{0j} = c_0\, r \cdot t_j$, for $j = 0, \ldots, \ell$.

Let $\beta_j = b_{0j}/b_{00}$. We know: $\gamma = c_0 \hat{t}_0 = c_0\, r \cdot t_0 = b_{00} < 1$ and thus:

$$\pi_{(m,j)} = \gamma^{-m} \beta_j \pi_{(0,0)}, \tag{6.8}$$

for $j = 0, \ldots, \ell$.

As has been mentioned at the end of section 6.2, explicit computation of the matrix B is in general hard, because it can be of infinite dimension and it can have a complicated structure.

In the presence of an exit state however, it turns out to be simpler to express π_m in terms of π_{m-1}, cf. Eq. (6.9).

By tabooing on the set $\underset{\sim}{L}_{m-1}$, there exists an $|\underset{\sim}{L}_{m-1}| \times (\ell+1)$ matrix R_m, such that

$$\pi_m = \pi_{m-1} R_m. \tag{6.9}$$

By our homogeneity assumptions, R_m is in fact independent of m, and so we suppress the dependance on m in our further notation, whenever possible. We will next define an embedded q-matrix $\check{Q}$ on L_m and an $\underset{\sim}{L}_{m-1} \times L_m$ matrix $\check{A}$ by removing the entrance state $(m+1,0)$, such that the mean (local) sojourn times in the state of L_m do not change, given a start in $\underset{\sim}{L}_m$. We next state the following theorem using this shift.

Theorem 6.2. *Consider a level homogeneous process with an exit state in each level. Then:*

$$\pi_m = \pi_{m-1} R,$$

where

$$R := -\check{A}\check{Q}^{-1},$$

where $\check{A}$ and $\check{Q}$ are $|\underset{\sim}{L}_{m-1}| \times (\ell+1)$ and $(\ell+1) \times (\ell+1)$ matrices with elements

$$\check{q}_{ij} = w_{ij} + \sum_{s=1}^{\infty} \sum_{r=0}^{\ell} u_{ir}^s \frac{d_{0j}}{\sum_{v=0}^{\ell} d_{0v}}, \tag{6.10}$$

$$\check{a}_{(k,i)(m,j)} = u_{ij}^{m-k} + \sum_{s=m+1-k}^{\infty} \sum_{r=0}^{\ell} u_{ir}^s \frac{d_{0j}}{\sum_{v=0}^{\ell} d_{0v}}, \text{ for } k < m, 0 \le i,j \le \ell, \tag{6.11}$$

respectively.

Notice that $d_{0j} / \sum_{r=0}^{\ell} d_{0r}$ is the probability that level set L_m is entered at state (m,j), given that a downward transition (to set L_m) occurs starting in state $(m+1,0)$.

Proof. We will explicitly compute R, containing the expected sojourn times spent in the states of level L_m, before absorption into $\underset{\sim}{L}_{m-1}$, given that the process starts in L_m.

To this end, note that since $(m+1,0)$ is an exit state for level $\widetilde{L}_{m+1}$, it is an entrance state for the set $\underset{\sim}{L}'_m$, defined below:

$$\underset{\sim}{L}'_{m:} = \underset{\sim}{L}_m \cup \{(m+1,0)\}.$$

This implies that $\underset{\sim}{L}'_m$ can only be reached from states in $\boldsymbol{S} \setminus \underset{\sim}{L}'_m$ through $(m+1,0)$. The results of Chapter 2 can be used to compute the expected sojourn time spent in the extended level $L'_m = L_m \cup \{(m+1,0)\}$ before absorption into $\underset{\sim}{L}_{m-1}$, given that the process starts at this extended level L'_m.

We use this technique to compute $\tilde{Q}$, the transition rate matrix of size $(\ell+2)\times(\ell+2)$ embedded on level L'_m for any m. Then $\tilde{Q}$ has elements:

$$\tilde{q}_{(k,i)(l,j)} = \begin{cases} w_{ij}, & (k,i),(l,j)\in L_m, \\ \sum_{s=1}^{\infty}\sum_{r=0}^{\ell} u_{ir}^{s}, & (k,i)\in L_m,(l,j)=(m+1,0), \\ d_{0j}, & (k,i)=(m+1,0),(l,j)\in L_m, \\ -\sum_{j=0}^{\ell} d_{0j}, & (k,i)=(l,j)=(m+1,0). \end{cases} \tag{6.12}$$

Note that the transitions leading to states in superlevel $\widetilde{L}_{m+1}\backslash\{(m+1,0)\}$ are mapped to the entrance state $(m+1,0)$. By virtue of Lemma 6.1 the expected sojourn time spent in level L'_m before absorption into the sub-level set $\underset{\sim}{L}_{m-1}$ is obtained by inverting $-\tilde{Q}$. Eq. 3.8 shows that

$$\pi'_m = -\underset{\sim}{\pi}_{m-1}\,\tilde{A}\tilde{Q}^{-1}, \tag{6.13}$$

where $\pi'_m = (\pi_m, \pi_{(m+1,0)})$ and the elements of $\tilde{A}$ are given below with $k<m$:

$$\tilde{a}_{(k,i)(l,j)} = \begin{cases} u_{ij}^{m-k}, & (l,j)=(m,j) \text{ and } i,j\in\{0,\ldots,\ell\} \\ \sum_{s=m+1-k}^{\infty}\sum_{r=0}^{\ell} u_{ir}^{s}, & (l,j)=(m+1,0). \end{cases} \tag{6.14}$$

The validity of this *exit state lumping* construction of Eq. (6.10) and (6.11) is guaranteed by Lemma 6.6 in appendix A. □

Restriction to QBD processes

For QBD processes, where $U^1 = U$ and $U^s = 0$ for $s\geq 2$, the above derivations imply that:

$$\pi_m = \pi_{m-1}R, \tag{6.15}$$

where $R = -U\check{Q}^{-1}$. If U is an invertible matrix, then the result above is equivalent to

$$\pi_{m-1} = -\pi_m\check{Q}U^{-1}.$$

In this equation we have expressed π_{m-1} in terms of π_m, similarly to Eq. (6.5). However, $\check{Q}U^{-1}$ is explicitly computable, provided that U is invertible, whereas B may not be invertible.

From Eq. (6.8) and (6.15) we have the following result

$$(1/\gamma)\beta = \beta R. \tag{6.16}$$

Particularly if U is an invertible matrix, then

$$\beta(-\check{Q})U^{-1} = \gamma\beta.$$

Lemma 6.3. *If ℓ is finite, then $1/\gamma$ is the maximum eigenvalue of R in absolute value.*

Proof. Let R be an irreducible non-negative matrix. Then by the Perron-Frobenius theorem (for example in Seneta [95]) the maximum eigenvalue is positive and real. We denote this eigenvalue by r. Further, there exist unique strictly positive left and right eigenvectors. Let x denote this positive right eigenvector. Then for all $i = 0, \ldots, \ell$:

$$rx_i = \sum_j r_{ij} x_j. \tag{6.17}$$

From Eq. (6.16) it is clear that $1/\gamma$ is an eigenvalue of R with β, (where $\beta > 0$) as the left eigenvector corresponding to $1/\gamma$, i.e.:

$$(1/\gamma)\beta_j = \sum_i \beta_i r_{ij}. \tag{6.18}$$

Next, we will show that $1/\gamma$ is the maximum eigenvalue of R, thus that $1/\gamma = r$.

From Eq. (6.17) we get

$$\sum_i r x_i \beta_i = \sum_j \sum_i \beta_i r_{ij} x_j.$$

Combining with Eq. (6.18) yields

$$r(x \cdot \beta) = \frac{1}{\gamma}(x \cdot \beta).$$

Since $x \cdot \beta > 0$, it follows that $r = 1/\gamma$. In other words, $1/\gamma$ is the maximum eigenvalue of R. □

In the next section we will discuss a single server queueing model with Coxian arrivals and batch services.

6.4 Queueing application: analysis of the $Cox(k)/M^Y/1$-queue

We consider a queueing model in which customers arrive within a maximum of k exponentially distributed phases. The inter-arrival distribution is a Coxian distribution of order k, see e.g. [28].

In a Coxian distribution of order k, phase $i \in \{0, \ldots, k-1\}$ lasts an exponentially distributed amount of time with parameter λ_i, at the end of which either a new customer arrives with

probability $1 - q_i$, or a new phase starts with probability q_i, where $q_{k-1} = 0$. Upon arrival of a new customer, a new inter-arrival distribution starts. To avoid trivialities, we assume that $q_i > 0$ for $i \leq k-2$. Hence the inter-arrival distribution is a $Cox(k)$ distribution with parameters $(\lambda_0, \ldots, \lambda_{k-1}, q_0, \ldots, q_{k-1})$.

We use this naming of the phases to have a more natural state space description: here, state (m, i) denotes the state of the system when there are m customers in the system and the $m+1$ arriving customer has completed i arrival phases. We will use the random variable $C \sim Cox(k)$ to denote the inter-arrival time.

The $Cox(k)/M^Y/1$ queueing system can be formulated as a QSF process $X(t)$ on the state space $\boldsymbol{S} = \{L_0, L_1, \ldots\}$, where $L_m = \{(m,0), (m,1), ..., (m,k-1)\}$ for all $m \geq 0$. Service occurs according to the distribution induced by Y; in batches of size j, with $1 \leq j \leq b$, each with probability p_j. We will denote the probability-generating function of Y as $\phi_Y(\cdot)$. Note that $\phi_Y(x) = \sum_{j=1}^{b} p_j x^j$.

We will first analyse the general $Cox(k)/M^Y/1$-queue with finitely many phases. Then we will discuss some special subcases, where the exponential phases all have the same parameter, or the probabilities of a new inter-arrival phase are all equal upto the order.

We will briefly discuss the extension to the case of a Coxian distribution of infinite order.

Analysis of the $Cox(k)/M^Y/1$-queue

The transition rate matrix Q for the $Cox(k)/M^Y/1$-queueing model is of the form displayed below:

$$Q = \begin{bmatrix} W_0 & U & 0 & 0 & 0 & \cdots \\ D_1' & W & U & 0 & 0 & \cdots \\ D_2' & D_1 & W & U & 0 & \cdots \\ D_3' & D_2 & D_1 & W & U & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{6.19}$$

with $k \times k$ sub-matrices $D_i = p_i \cdot \mu I$, $D_k' = \sum_{i=k}^{y} D_i$, where I is the identity matrix of size $k \times k$ and

$$W_0 = \begin{bmatrix} -\lambda_0 & q_0\lambda_0 & \cdots & \cdots & 0 \\ 0 & -\lambda_1 & q_1\lambda_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -\lambda_{k-2} & q_{k-2}\lambda_{k-2} \\ 0 & 0 & \cdots & 0 & -\lambda_{k-1} \end{bmatrix}, U = \begin{bmatrix} (1-q_0)\lambda_0 & 0 & \cdots & 0 \\ (1-q_1)\lambda_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{k-1} & 0 & \cdots & 0 \end{bmatrix},$$

and $W = W_0 + \mu I$. Note that U has a $c \cdot r$ structure, with

$$c = \begin{pmatrix} (1-q_0)\lambda_0 \\ \vdots \\ (1-q_{k-2})\lambda_{k-2} \\ \lambda_{k-1} \end{pmatrix} \quad \text{and } r = (1, 0, \cdots, 0),$$

and therefore fits the analysis of Section 6.2. Note further that the structure of Eq. (6.19) is the transpose of the matrix introduced in Eq. (6.1). We have to interchange the roles of the matrices U and D in the formulæ of the previous sections to fit the model under consideration. Another possible way to look at the problem is to use the original negative numbering of the levels. Then we do not need the interchanging described above, but the numbering is not induced by the model anymore.

We assume that the QSF process is ergodic. We shall prove below that for ergodicity it is necessary and sufficient to require that the mean number of arrivals per unit time is smaller than the mean number of service completions, which is specified in the relation below:

$$\frac{1}{\sum_{i=0}^{k-1} \prod_{l=0}^{i} q_l \lambda_i^{-1}} = \frac{1}{\mathsf{E}C} < \mu \sum_{j=1}^{b} j p_j = \mu \mathsf{E}Y. \tag{6.20}$$

When ergodicity holds, we can use the results of Theorem 6.1, and the stationary distribution takes the form: (again, note the change of notation, since we are considering positive levels now)

$$\pi_{(m,j)} = \gamma^{m-1} \sum_{i=0}^{k-1} \pi_{(0,i)} c_i \hat{t}_j = \gamma^{m-1} \sum_{i=0}^{k-1} \pi_{(0,i)} \lambda_i (1-q_i) \hat{t}_j,$$

where for notational convenience, we use $q_{k-1} = 0$. The factor γ and the vector $\hat{t}$ denote the distribution of the phase and are implicitly given as largest positive eigenvalue and corresponding left eigenvector of the rate matrix R for a fixed level. Note that once γ and $\hat{t}$ are known, the distribution of level 0 can be deduced.

An alternative procedure is carried out by observing that the state $(m, 0)$ is an entrance state for $\widetilde{L}_m$ (and an exit state for the shifted partition $\widetilde{L}_{m-1} \cup \{(m, 0)\}$). This allows us to use the results from Section 6.3 for computing the rate matrix in the reverse (downward) direction. However, we prefer not to use the shifted partition, because the emerging stationary distribution in both phases and levels will then have a notationally less amenable form. Therefore we will stay with the current level partitions.

In consideration of the statements above, the following relation holds (cf. Eq. (6.13)) for levels $m \geq 1$:

$$\pi_m = \pi_{m+1} R = -\widetilde{\pi}_{m+1} \tilde{A} \tilde{Q}^{-1}, \tag{6.21}$$

where $\tilde{A}$ and $\tilde{Q}$ are given in Eq. (6.14 and (6.12). We refer to Eq. (3.5), where a similar

formula is provided, but not the results regarding the product form. For this model, the matrix $\tilde{Q}$ takes the form:

$$\tilde{Q} = \begin{bmatrix} -\lambda_0 & q_0\lambda_0 & \cdots & 0 & 0 & 0 \\ \mu & -(\lambda_1+\mu) & q_1\lambda_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \mu & 0 & \cdots & & -(\lambda_{k-2}+\mu) & q_{k-2}\lambda_{k-2} \\ \mu & 0 & \cdots & & 0 & -(\lambda_{k-1}+\mu) \end{bmatrix}.$$

Note that $\tilde{Q}$ is independent of the batch size distribution. The matrix $\tilde{A}$ has dimension $k \times bk$, and can therefore be written as:

$$\tilde{A} = (\tilde{A}_1 \; \tilde{A}_2 \; \ldots \tilde{A}_b)^T,$$

where $\tilde{A}_i$ is the following $k \times k$ matrix:

$$\tilde{A}_i = \begin{bmatrix} \sum_{j \geq i} p_j\mu & 0 & \cdots & 0 \\ \sum_{j > i} p_j\mu & p_i\mu & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j > i} p_j\mu & 0 & \cdots & p_i\mu \end{bmatrix},$$

for $1 < i < b$ and $\tilde{A}_b = p_b \cdot \mu I$.

Since the rate matrix R expresses the stationary solution of lower levels in terms of that of higher ones, we cannot recursively compute the stationary distribution in the infinite capacity case. We still need to find γ and $\hat{t}$ to do so. Instead we will derive equations for γ and the unknown (conditional) stationary probabilities of the phases. To this end we consider the expression:

$$-\pi_m\tilde{Q} = \widetilde{\pi}_{m+1}\tilde{A}.$$

Because of the level product form solution (cf. Eq. (6.8)) we can rewrite this equation as follows:

$$-\pi_m\tilde{Q} = \sum_{i=1}^{b} \pi_{m+i}\tilde{A}_i = \pi_m \sum_{i=1}^{b} \gamma^i \tilde{A}_i. \tag{6.22}$$

Using Eq. (6.7) (i.e. $\pi_{(m,i)} = \hat{t}_i\pi_{(m,0)}$) to calculate the components of the vectors with $1 \leq i \leq k-1$ on both sides of the above equality Eq. (6.22) yields the following expression:

$$(\lambda_i+\mu)\hat{t}_i - q_{i-1}\lambda_{i-1} \cdot \hat{t}_{i-1} = \sum_{j=1}^{b} \gamma^j p_j\mu \cdot \hat{t}_i = \mu\phi_Y(\gamma) \cdot \hat{t}_i.$$

Hence:

$$\frac{\hat{t}_i}{\hat{t}_{i-1}} = \frac{q_{i-1}\lambda_{i-1}}{\lambda_i + \mu - \mu\phi_Y(\gamma)}. \tag{6.23}$$

In other words, writing $\beta_i = q_{i-1}\alpha_i = \hat{t}_i/\hat{t}_{i-1}$, we get that $\hat{t}_i = \prod_{l=1}^{i} q_{l-1}\alpha_l \cdot \hat{t}_0$. Therefore the stationary distribution has the following form for $m \geq 1$

$$\pi_{(m,i)} = \pi_{(1,0)}\gamma^{m-1}\prod_{l=1}^{i} q_{l-1}\alpha_l, \tag{6.24}$$

with α_i the following function of γ derived from Eq. (6.23):

$$\alpha_i = \frac{\lambda_{i-1}}{\lambda_i + \mu - \mu\phi_Y(\gamma)}. \tag{6.25}$$

Note that $(\alpha_1, \ldots, \alpha_{k-1})$ only depends on $(q_0, \ldots, q_{k-2})$ through γ.

Using the balance equation for state $(m, 0)$ $(m \geq 1)$ and Eq. (6.25) we obtain the following expression for γ as a function of $(\alpha_1, \ldots, \alpha_{k-1})$:

$$\gamma = \frac{\alpha_1(1-q_0)\lambda_0 + \alpha_1\sum_{i=1}^{k-1}\prod_{l=1}^{i}\alpha_l q_{l-1}(1-q_i)\lambda_i}{(\lambda_0 - \lambda_1)\alpha_1 + \lambda_0}, \tag{6.26}$$

where we have substituted $\phi_Y(\gamma)\mu = \lambda_1 + \mu - \lambda_0/\alpha_1$. Eq. (6.25) and (6.26) provide a system of k equations in the unknowns γ and α_i, $i = 1, \ldots, k-1$. As is easily checked, $\alpha_i = \lambda_{i-1}/\lambda_i$, $i = 1, \ldots, k-1$, $\gamma = 1$ form a solution of this system.

Using the same balance equation, we obtain a fixpoint equation for γ:

$$\gamma = F(\gamma), \tag{6.27}$$

with $F : \mathbb{R} \to \mathbb{R}$ given by

$$F(\gamma) = \frac{\lambda_0(1-q_0) + \sum_{i=1}^{k-1}\prod_{l=1}^{i} q_{l-1}\frac{\lambda_{l-1}}{\lambda_l + \mu - \mu\phi_Y(\gamma)}(1-q_i)\lambda_i + \gamma\mu\phi_Y(\gamma)}{\lambda_0 + \mu}. \tag{6.28}$$

Again, $\gamma = 1$ is a fixpoint of F. The function F is tediously but easily checked to be a convex function of γ on the interval $[0, 1+\epsilon]$ for some $\epsilon > 0$. It is positive on this interval, and hence larger than the left hand side with $\gamma = 0$, i.e. the value of $F(0)$. Computing the derivative at $\gamma = 1$ yields, that this is strictly larger than 1 if and only if the condition in Eq. (6.20) holds. Hence, F has one fixpoint $\gamma < 1$ if and only if Eq. (6.20) holds, showing that this equation is necessary and sufficient for ergodicity. This is a standard argument used for deriving ergodicity conditions by means of a probability generating function approach,

see e.g. [89] Section 4.5.

Hence solutions γ and $\{\alpha_i\}_{i=1}^k$ can be determined (i) by determining the unique fixpoint $\gamma < 1$ of F and then (ii) insert γ into Eq. (6.25).

Next we will express the steady state probabilities of the first level in terms of the steady state probability in state $(0,0)$. For sake of presentation, we omitted the derivation and present the results immediately:

$$\pi_{(1,0)} = \pi_{(0,0)} \frac{\lambda_0}{\mu} \frac{1-\gamma}{1-\phi_Y(\gamma)}, \tag{6.29}$$

$$\pi_{(0,i)} = \pi_{(0,0)} \frac{\lambda_0}{\lambda_i} \prod_{s=0}^{i-1} q_s \left(1 + \frac{(1-\gamma)\phi_Y(\gamma)}{\gamma(1-\phi_Y(\gamma))} \sum_{j=1}^{i} \prod_{l=1}^{j} \alpha_l \right), \quad \text{for } i \geq 1, \tag{6.30}$$

and

$$\pi_{(0,0)} = \left(\left(1 + \frac{\lambda_0}{\mu(1-\phi_Y(\gamma))} \right) \left(1 + \sum_{i=1}^{k-1} \prod_{j=0}^{i-1} \frac{\lambda_0}{\lambda_i} q_j \right) \right)^{-1}. \tag{6.31}$$

Then by using Eq. (6.24) and Eq. (6.29)-(6.31), for $m \geq 1$ we have

$$\pi_{(m,i)} = \pi_{(0,0)} \frac{\lambda_0(1-\gamma)\gamma^{m-1}}{\mu(1-\phi_Y(\gamma))} \prod_{l=1}^{i} q_{l-1}\alpha_l. \tag{6.32}$$

We summarize all our findings above in the next theorem.

Theorem 6.3. *The $Cox(k)/M^Y/1$ queue is ergodic in and only if Eq. (6.20) is satisfied. If this is the case, the stationary distribution of the $Cox(k)/M^Y/1$ queue on levels ≥ 1 is given by Eq. (6.32). The factors α_i, $i = 1, \ldots, k-1$ and γ can be calculated from Eq. (6.27) and Eq. (6.25). The boundary level can by found by Eq. (6.30) and (6.31).*

Remark 6.1 (Finite capacity queues)**.** In case of a finite capacity queue of size S, the stationary distribution can be computed recursively, in terms of π_S. By using the fact that level S has an entrance state $(S,0)$ from the right, the analysis described in the previous section can be used and yields that

$$\pi_{(S,i)} = -\tilde{Q}_S^{-1} \pi_{(S,0)},$$

where now

$$\tilde{Q}_S = \begin{bmatrix} -\lambda_0 & \lambda_0 & & \cdots & 0 & 0 \\ \mu & -(\lambda_1+\mu) & \lambda_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ \mu & 0 & & \cdots & -(\lambda_{k-2}+\mu) & \lambda_{k-2} \\ \mu & 0 & \cdots & & 0 & -\mu \end{bmatrix}.$$

Then using Eq. (6.21) with the matrices $\tilde{Q}_m$ and $\tilde{A}_m$ associated with level $m < S$, one may compute the lower level stationary probabilities (up to a constant). Final renormalisation yields the correct stationary distribution.

Remark 6.2 (Variable service rates)**.** Suppose that the service rates μ and the batch probabilities p_j are equal to μ^m and p_j^m for level $m \leq S$, $j = 1, ..., b$, respectively, for some $S < \infty$.Then the stationary distribution of $Cox(k)/M^Y/1$ queue on the levels $S + k$, $k \geq 1$, can be computed in exactly the same manner as in the above, yielding the stationary probabilities of these levels upto a constant. Meanwhile, the stationary distribution for levels $m \leq S$ can be computed from $\tilde{\pi}_S$ by using Eq. (6.21) with the matrices $\tilde{Q}_m$ and $\tilde{A}_m$ associated with level m, as described in Remark 6.1. Again, final renormalisation yields the correct stationary distribution.

6.4.1 A $Cox(k)$ inter-arrival distribution with homogeneous parameters

In this section we consider several subcases for the model described in the previous section.

Homogeneous rates

First let us assume that the rates in the exponential distribution are all equal: $\lambda_i = \lambda$, $i = 0, \ldots, k-1$. Then from Eq. (6.25) we get

$$\alpha_i = \frac{\lambda}{\lambda + \mu - \mu\phi_Y(\gamma)} =: \alpha, \quad i = 1, \ldots, k-1, \tag{6.33}$$

in other words, the phase factors α_i have become independent of the phase. Then Eq. (6.26) reduces to

$$\gamma = \alpha(1 - q_0) + \sum_{i=1}^{k-1} \big(\prod_{l=0}^{i-1} q_l \big) \alpha^{i+1} (1 - q_i). \tag{6.34}$$

The function F can be simplified, but we can also directly insert Eq. (6.33) for α in Eq. (6.34) to obtain that the solution γ is a fixpoint of the equation $\gamma = F^h(\gamma)$ with

$$F^h(\gamma) = \frac{\lambda(1 - q_0)}{\lambda + \mu - \mu \cdot \phi_Y(\gamma)} + \sum_{i=1}^{k-1} \Big(\prod_{l=0}^{i-1} q_l \Big) \Big(\frac{\lambda}{\lambda + \mu - \mu \cdot \phi_Y(\gamma)} \Big)^{i+1} (1 - q_i).$$

In the same manner as for the function F, we can deduce that the ergodicity condition in Eq. (6.20) is necessary and sufficient for F^h to have a fixpoint $\gamma < 1$.

Homogeneous rates and probabilistic phase transitions

Next we assume that additionally the probabilities of a new phase are all equal, that is, $q_i = q$, $i = 0, \ldots, k-2$. This has no further impact on α. However, Eq. (6.26) reduces further to

$$\gamma = \sum_{i=0}^{k-2} q^i \alpha^{i+1}(1-q) + q^{k-1}\alpha^k.$$

The function F^h becomes

$$F^h(\gamma) = \sum_{i=0}^{k-2} q^i \Big(\frac{\lambda}{\lambda+\mu-\mu\cdot\phi_Y(\gamma)}\Big)^{i+1}(1-q) + q^{k-1}\Big(\frac{\lambda}{\lambda+\mu-\mu\cdot\phi_Y(\gamma)}\Big)^{k}.$$

Further, the stationary distribution of the levels $m \geq 1$ has the following product form expression:

$$\pi_{(m,i)} = \pi_{(0,0)} \frac{\lambda(1-\gamma)\gamma^{m-1}(q\alpha)^i}{\mu(1-\phi_Y(\gamma))}, \quad m \geq 1,$$

and

$$\pi_{(0,i)} = \pi_{(0,0)} q^i \left(1 + \frac{(1-\gamma)\phi_Y(\gamma)}{\gamma(1-\phi_Y(\gamma))} \sum_{j=1}^{i} \alpha^j\right), \quad \text{for } i = 1, \cdots, k-1,$$

where $\pi_{(0,0)} = \dfrac{(1-q)\mu(1-\phi_Y(\gamma))}{(1-q^k)(\lambda+\mu(1-\phi_Y(\gamma)))} = \dfrac{(1-q)(1-\alpha)}{1-q^k}$.

Homogeneous rates and deterministic phase transitions: $E_k/M^Y/1$-queue

Taking $q_i = q = 1$, $i = 0, \ldots, k-2$, finally yields an Erlang (k, λ) inter-arrival distribution. Then Eq. (6.26) takes the very simple form:

$$\gamma = \alpha^k. \tag{6.35}$$

The function F^h is simply given by $F^h(\gamma) = \Big(\frac{\lambda}{\lambda+\mu-\mu\cdot\phi_Y(\gamma)}\Big)^k$. Finally, the stationary distribution has a product form (except for level 0) determined by only one factor

$$\pi_{(m,i)} = \pi_{(0,0)} \frac{\lambda(1-\gamma)\alpha^{i+k(m-1))}}{\mu(1-\phi_Y(\gamma))}, \quad m \geq 1,$$

and

$$\pi_{(0,i)} = \pi_{(0,0)} \left(1 + \frac{(1-\gamma)\phi_Y(\gamma)}{\gamma(1-\phi_Y(\gamma))} \sum_{j=1}^{i} \alpha^j\right), \quad \text{for } i = 1, \cdots, k-1,$$

where $\pi_{(0,0)} = \dfrac{1-\alpha}{k}$.

Notice that this is not surprising. The process is then essentially a one-dimensional process with a GI/M/1-structure. This is known to have a product form stationary distribution (cf. Asmussen [22]).

We further note that the $Cox(k)/M^Y/1$-queue with constant rates and probabilistic phase transitions is *not* of the (pure) GI/M/1-type. Indeed, the probabilities are only constant with respect to the phases with index smaller than $k-1$.

Distribution of the number of customers in the queue

By taking the summation over the phases per level we get the following modified geometric distribution for the number of customers in $E_k/M^Y/1$-queue

$$\bar{\pi}_m^k = \begin{cases} \dfrac{\lambda(1-\gamma)^2}{k\mu\cdot(1-\phi_Y(\gamma))}\gamma^{m-1}, & m>0, \\ 1-\dfrac{\lambda(1-\gamma)}{k\mu\cdot(1-\phi_Y(\gamma))}, & m=0, \end{cases} \tag{6.36}$$

where $\bar{\pi}_m^k = \sum_{i=0}^{k-1}\pi_{(m,i)}$, $m\geq 0$.

6.4.2 The $Cox(\infty)/M^Y/1$-queue

Allowing infinitely many phases still fits our framework. We get the natural extensions of formulas Eq. (6.26) and Eq. (6.28) with $k=\infty$. Clearly the expression for α_i as a function of γ is not affected by the amount of phases.

Restricting to the case of homogeneous rates and probabilities, so that $\lambda_i = \lambda$, $q_i = q$, $i = 0,\ldots,$ we get the following results.

The queueing system is ergodic if and only if

$$\lambda\left(\frac{1-q}{q}\right) < \mu\sum_{j=1}^{b} jp_j. \tag{6.37}$$

Then Eq. (6.26) becomes

$$\gamma = \frac{\alpha(1-q)}{1-q\alpha}, \tag{6.38}$$

and so again we obtain

$$F^h(\gamma) = \frac{\lambda(1-q)}{\lambda(1-q) + \mu(1-\phi_Y(\gamma))}.$$

We summarize our results in the next theorem.

Theorem 6.4. *Under the ergodicity condition Eq. (6.37), the $Cox(\infty)/M^Y/1$ queue has a stationary distribution on levels ≥ 1 given by Eq. (6.32) where α and γ can be calculated from Eq. (6.33) and Eq. (6.38). The boundary level can by found by Eq. (6.30), where $\pi_{(0,0)} = (1-q)(1-\alpha)$.*

6.4.3 Numerical analysis

Assume an ergodic $Cox(k)/M^Y/1$ queue of finite order. In order to calculate the solutions $\{\alpha_i, i = 1, \ldots, k-1; \gamma\}$, one can solve Eq. (6.27) directly, e.g. by the Newton-Raphson method, and then determine α_i, $i = 1, \ldots, k-1$, from (6.25).

As γ is a fixpoint of the map F associated with Eq. (6.27), another possibility is to approximate the fixpoint $\gamma < 1$ by selecting γ_0 and by iteratively computing $\gamma_{n+1} = F(\gamma_n)$. It is simply checked that the fixpoint $\gamma = 1$ is not stable, but that the fixpoint of interest, smaller than 1, is a stable fixpoint. Hence we can use the following scheme to approximate the desired values $\alpha_i, i = 1, \ldots, k-1, \gamma$.

Approximation scheme 1

1. Choose $\gamma < 1$;
2. iteratively put $\gamma := F(\gamma)$, till desired convergence; compute α_i from Eq. (6.25), $i = 1, \ldots, k-1$.

In the case of homogeneous rates, we have shown that the level factor γ is the unique fixpoint smaller than 1 of the function F^h . This leads to the following adapted scheme, that we only formulate for the case of homogeneous rates.

Approximation scheme 2 for the case $\lambda_i = \lambda$, $i \leq k-1$ (where $k = \infty$ is allowed)

1. Choose $\gamma < 1$;
2. iteratively put $\gamma := F^h(\gamma)$, until desired convergence, and compute α from Eq. (6.33).

It is outside the scope of the paper to discuss the rate of convergence of the scheme, as well as a detailed stopping criterion. Table 1 below shows the non-surprising property that γ and α are non-increasing in q. Further note that even for a high value of the continuation probability q, γ is already approximately constant starting from around 20 phases.

q	$k=2$		$k=5$		$k=1000$		$k=\infty$	
	γ	α	γ	α	γ	α	γ	α
0.1	0.4168	0.4414	0.4153	0.4411	0.4153	0.4411	0.4153	0.4411
0.2	0.3847	0.4339	0.3788	0.4325	0.3788	0.4325	0.3788	0.4325
0.3	0.3538	0.4272	0.3406	0.4246	0.3406	0.4246	0.3406	0.4246
0.4	0.3239	0.4214	0.3006	0.4173	0.3005	0.4172	0.3005	0.4172
0.5	0.2948	0.4163	0.2585	0.4105	0.2582	0.4104	0.2582	0.4104
0.6	0.2663	0.4117	0.2141	0.4042	0.2134	0.4042	0.2134	0.4042
0.7	0.2385	0.4076	0.1673	0.3986	0.1658	0.3984	0.1658	0.3984
0.8	0.2113	0.4039	0.1176	0.3935	0.1147	0.3932	0.1147	0.3932
0.9	0.1845	0.4006	0.0648	0.3890	0.0598	0.3886	0.0598	0.3886
1	0.1581	0.3976	0.0085	0.3851	≈ 0	0.3846	≈ 0	0.3846

Table 1. $(\lambda, \mu, \gamma, p_1, p_2, p_3) = (0.5, 0.8, 0.35, 0.25, 0.5, 0.25)$.

6.5 Monotonicity properties and relation with the $D/M^Y/1$-queue

6.5.1 Monotonicity properties

Next we will study monotonicity properties for the homogeneous $Cox(k)/M^Y/1$ queues, in the following sense. If $\lambda_i = \lambda$ and $q_i = q$ for $i = 0, \ldots, k-2$, then we denote the corresponding k-order Coxian distribution by $Cox(k, \lambda, q)$.

In the remainder of the paper we assume that the ergodicity condition Eq. (6.20) is satisfied. It is the aim to compare the stationary distribution of the number of customers in the system for $Cox(k, \lambda, q)/M^Y/1$ queues, with different inter-arrival distributions with the same mean inter-arrival times and associsated probabilities q, but with a different amount of phases.

Let λ^*, phase probability q and the number of phases k be given. Then, in order for the mean inter-arrival time to equal $1/\lambda^*$, the parameter λ_k of the homogeneous Coxian distribution has to be equal to:

$$\frac{1}{\lambda^*} = \frac{1}{\lambda_k} \sum_{l=0}^{k-1} q^l = \frac{1-q^k}{\lambda_k(1-q)},$$

that is

$$\lambda_k = \frac{\lambda^*(1-q^k)}{1-q}.$$

Denote the corresponding factors in the stationary distribution (see previous section) of the corresponding QSF process by α_k and γ_k respectively, and denote the stationary distribution of the number of customers in the system by $\bar{\pi}^k$.

It does not seem possible to stochastically compare these queueing systems for a different number of phases: Table 2 below shows that there is a lack of monotonicity in the parameter γ_k, especially for high values of the continuation parameter q.

q	$k=2$		$k=5$		$k=50$		$k=1000$	
	γ	α	γ	α	γ	α	γ	α
0.1	0.4485	0.4734	0.4502	0.4764	0.4502	0.4764	0.4502	0.4764
0.2	0.4439	0.4939	0.4501	0.5057	0.4502	0.5058	0.4502	0.5058
0.3	0.4371	0.5120	0.4494	0.5383	0.4502	0.5391	0.4502	0.5391
0.4	0.4286	0.5283	0.4472	0.5738	0.4502	0.5771	0.4502	0.5771
0.5	0.4189	0.5429	0.4415	0.6111	0.4502	0.6209	0.4502	0.6209
0.6	0.4082	0.5563	0.4300	0.6488	0.4502	0.6718	0.4502	0.6718
0.7	0.3968	0.5685	0.4105	0.6853	0.4502	0.7319	0.4502	0.7319
0.8	0.3848	0.5798	0.3811	0.7197	0.4502	0.8037	0.4502	0.8037
0.9	0.3725	0.5902	0.3407	0.7514	0.4484	0.8905	0.4502	0.8912

Table 2. $(\lambda^*, \mu, \gamma, p_1, p_2, p_3) = (0.5, 0.8, 0.35, 0.25, 0.5, 0.25)$.

However, in the case of a deterministic number of phase transitions, an Erlang queue, there exists more structure regarding the change of γ with respect to a increasing number of phases k.

Monotonicity properties for the $E_k/M^Y/1$-queue

Let us next restrict to the case $q = 1$, in other words, the case of an Erlang inter-arrival distribution. Then the arrival rate in the k-phase system is given by $\lambda_k = \lambda^* k$. Write $\rho = \lambda^*/\mu$.

The following results hold.

Theorem 6.5.

a) The sequence γ_k is strictly decreasing in k. It has limit $\gamma^* = \lim_{k\to\infty} \gamma_k$, which is the unique solution ξ smaller than 1 of the equation below.

$$\xi = e^{-(1-\phi_Y(\xi))/\rho}. \tag{6.39}$$

b) The map $k \mapsto \bar{\pi}_0^k = 1 - \rho(1-\gamma_k)/(1-\phi_Y(\gamma_k))$ is a strictly decreasing function if and only if $P\{Y=1\} < 1$. If $P\{Y=1\} = 1$, i.e. the batch size equals 1 with probability 1, then $\bar{\pi}_0^k = 1-\rho$, for $k = 1, 2, \ldots$

Proof. By combination of Eq. (6.33) and (6.35) we obtain:

$$1/\gamma_k = \left(1 + \frac{(1-\phi_Y(\gamma_k))}{k\rho}\right)^k.$$

Define $g_k(x) := \left(1 + \frac{x}{k}\right)^k, \quad k = 0, 1, \ldots.$

Clearly this function is increasing in x. To show that g_k is also increasing in k, we expand the expression using the binomial formula:

$$\begin{aligned} g_{k+1}(x) &= \sum_{i=0}^{k+1} \binom{k+1}{i} \left(\frac{x}{\rho(k+1)}\right)^i \\ &= 1 + \sum_{i=1}^{k} \frac{(x/\rho)^i}{i!} \cdot \frac{k}{k+1} \cdots \frac{k+2-i}{k+1} + \frac{(x/\rho)^{k+1}}{(k+1)^{k+1}}. \end{aligned} \tag{6.40}$$

On the other hand,

$$\begin{aligned} g_k(x) &= \sum_{i=0}^{k} \binom{k}{i} \left(\frac{x}{\rho(k)}\right)^i \\ &= 1 + \sum_{i=1}^{k} \frac{(x/\rho)^i}{i!} \cdot \frac{k-1}{k} \cdots \frac{k+1-i}{k}. \end{aligned} \tag{6.41}$$

Eq. (6.40) and Eq. (6.41) yield via a term by term comparison that

$$g_{k+1}(x) > g_k(x), \quad \text{for all} \quad x > 0.$$

In other words, $g_{k+1}(1-\phi_Y(\gamma)) > g_k(1-\phi_Y(\gamma))$, with $g_{k+1}(1-\phi_Y(1)) = g_k(1-\phi_Y(1)) = 1$ and $g_{k+1}(1 - \phi_Y(0)) = (1 + \rho^{-1})^{k+1} > (1 + \rho^{-1})^k$. Recall that γ_i is the unique fixpoint of the equation $1/\gamma = g_i(1 - \phi_Y(\gamma))$, with $\gamma_i \in (0, 1)$, $i = k, k + 1$. For $x < \gamma_i$, $1/x > g_i(1 - \phi_Y(x))$ and for $x \in (\gamma_i, 1)$, necessarily $1/x < g_i(1 - \phi_Y(x))$, $i = k, k + 1$. Hence $\gamma_{k+1} < \gamma_k$.

Since γ_k are non-increasing, and bounded below, the sequence has a limit, γ^* say, with $\gamma^* < 1$. This limit solves Eq. (6.39) by the standard limiting argument that $\lim_{k\to\infty}(1 + x/k)^k = e^x$. The function $\gamma \mapsto e^{-(1-\phi_Y(\gamma))}$ is a convex function, with fixpoint s $\gamma = 1$ and $\gamma^* < 1$, derivative larger than 1 at $\gamma = 1$, and positive value at $\gamma = 0$. The result then follows in a standard manner, thus completing the proof of a).

Part b) follows from the fact that

$$\frac{1-\gamma}{1-\phi_Y(\gamma)} = \frac{1}{\sum_{i=1}^{b} p_i \sum_{j=0}^{i-1} \gamma^j}, \tag{6.42}$$

which is strictly increasing in $\gamma < 1$ if and only if $P\{Y = 1\} < 1$. □

Clearly, with increasing k, the variance of the inter-arrival time decreases. Hence, the average server utilisation strictly improves with decreasing variance, in the case of batch sizes Y, with $P\{Y = 1\} < 1$. Whereas, if the batch size equals 1 with probability 1, the average server utilisation is equal to ρ and thus constant.

We can say a little more. We define $L_k = \sum_m m \cdot \bar{\pi}_m^k$ to be the mean number of customers in the system under the stationary distribution. Further, we denote by W_k the expected sojourn time and by V_k the variance. The following comparison result holds.

Theorem 6.6. *The following are true:*

$$L_{k+1} \leq L_k \quad W_{k+1} \leq W_k, \quad \textit{and} \quad V_{k+1} \leq V_k, \quad \textit{for} \quad k = 1, 2, \ldots.$$

Proof. The expectations with respect to $\bar{\pi}_{k+1}$ and $\bar{\pi}_k$ are equal to L_{k+1} and L_k respectively.

It is easy to check that

$$L_k = \frac{\rho}{1 - \phi_Y(\gamma_k)}, \quad \text{for} \quad k = 1, 2, \ldots. \tag{6.43}$$

Since γ_k are non-increasing, then

$$1 - \phi_Y(\gamma_k) \leq 1 - \phi_Y(\gamma_{k+1}).$$

It follows that $L_{k+1} \leq L_k$. Application of Little's formula yields $W_{k+1} \leq W_k$.

From Eq. (6.36) it is clear that the number of customers in the $E_k/M/1$-queue is (almost) geometrically distributed. This implies that $V_k = \rho^2\gamma_k/(1 - \gamma_k)^2$. It is easy to check that this yields $V_{k+1} \leq V_k$.

Next, we need to check the variance for a general batch size distribution:

$$V_k = \frac{\rho(1 - \gamma_k)^2}{1 - \phi_Y(\gamma_k)} \sum_{m \geq 1} m^2 \gamma_k^{m-1} - L_k^2. \tag{6.44}$$

Applying geometric series sum yields

$$\begin{aligned} \sum_{m \geq 1} m^2 \gamma_k^{m-1} &= \gamma_k \sum_{m \geq 1} m(m-1)\gamma_k^{m-2} + \sum_{m \geq 1} m\gamma_k^{m-1} \\ &= \frac{2\gamma_k}{(1 - \gamma_k)^3} + \frac{1}{(1 - \gamma_k)^2}. \end{aligned} \tag{6.45}$$

By combination of Eq. (6.43), Eq. (6.44), and Eq. (6.45) we get:

$$\begin{aligned} V_k &= \frac{\rho(1-\gamma_k)^2}{1-\phi_Y(\gamma_k)}\left(\frac{2\gamma_k}{(1-\gamma_k)^3}+\frac{1}{(1-\gamma_k)^2}\right)-\frac{\rho^2}{(1-\phi_Y(\gamma_k))^2} \\ &= \frac{\rho}{1-\phi_Y(\gamma_k)}\left(\frac{1+\gamma_k}{1-\gamma_k}-\frac{\rho}{1-\phi_Y(\gamma_k)}\right). \end{aligned}$$

We define $h(\gamma) := \dfrac{1+\gamma}{1-\gamma}-\dfrac{\rho}{1-\phi_Y(\gamma)}$. Since $\gamma_k \geq \gamma^*$ and $\gamma_k \downarrow \gamma^*$ by virtue of Theorem 6.5 (a), it is sufficient to show that $h'(\gamma) = \frac{dh(\gamma)}{d\gamma} \geq 0$ for $\gamma \in [\gamma^*, 1)$.

$$h'(\gamma) = \frac{1}{(1-\gamma)^2}\left(2-\rho\left(\frac{1-\gamma}{1-\phi_Y(\gamma)}\right)^2\phi_Y'(\gamma)\right).$$

Applying Eq. (6.42) yields

$$h'(\gamma) = \frac{1}{(1-\gamma)^2}\left(2-\rho\left(\frac{1}{\sum_{j=1}^b p_j\sum_{i=0}^{j-1}\gamma^i}\right)^2\sum_{j=1}^b jp_j\gamma^{j-1}\right).$$

Since $j\gamma^{j-1} \leq \sum_{i=0}^{j-1}\gamma^i$ for $\gamma \in [\gamma^*, 1)$, it follows that

$$h'(\gamma) \geq \frac{1}{(1-\gamma)^2}\left(2-\frac{\rho}{\sum_{j=1}^b p_j\sum_{i=0}^{j-1}\gamma^i}\right).$$

For $\gamma \in [\gamma^*, 1)$,

$$\frac{\rho}{\sum_{j=1}^b p_j\sum_{i=0}^{j-1}\gamma^i} \leq \frac{\rho}{\sum_{j=1}^b p_j\sum_{i=0}^{j-1}(\gamma^*)^i}.$$

So, to show that $h'(\gamma \geq 0)$ for $\gamma \in [\gamma^*, 1)$, it is sufficient to show that

$$2 \geq \frac{\rho}{\sum_{j=1}^b p_j\sum_{i=0}^{j-1}(\gamma^*)^i} = \frac{\rho}{\sum_{j=1}^b p_j\frac{1-(\gamma^*)^j}{1-\gamma^*}},$$

or

$$2(1-\phi_Y(\gamma^*))/\rho \geq 1-\gamma^*. \tag{6.46}$$

By virtue of Theorem 6.5 (a), $\gamma^* \geq 1-(1-\phi_Y(\gamma^*))/\rho$ (since $e^{-x} \geq 1-x$, for $x \geq 0$).

This implies that

$$(1-\phi_Y(\gamma^*))/\rho \geq 1-\gamma^*.$$

So that Eq. (6.46) follows.

Then

$$\frac{\rho}{1-\phi_Y(\gamma_k)}h(\gamma_k) \geq \frac{\rho}{1-\phi_Y(\gamma_{k+1})}h(\gamma_{k+1}).$$

In other words, we have proved that $V_k \geq V_{k+1}$. □

Interestingly enough, for general batch size distribution, the stationary distribution does not stochastically decrease with increasing k. This follows immediately from the fact that $\sum_{m\geq 1}\pi_m^k$ is strictly increasing in k, whereas γ_k is strictly decreasing. However, if the batch size is indentically equal to 1, then the stationary distribution has a stochastically monotonic behaviour as a function of k.

Corollary 6.1. *Suppose that $P\{Y=1\}=1$. The following is true for $k=1,2,\ldots$:*

$$\bar{\pi}_{k+1} \overset{st}{\preceq} \bar{\pi}_k,$$

or equivalently: $\sum_{m\geq M}\bar{\pi}_m^{k+1} \leq \sum_{m\geq M}\bar{\pi}_m^k$, for all $M=0,1,\ldots$.

In this particular case we can also prove that α_k is strictly increasing in k.

Lemma 6.4. *Suppose that $P\{Y=1\}=1$. The sequence of parameters α_k is strictly increasing in k.*

Proof. From Eq. (6.33) we have $\alpha_k = \dfrac{\lambda_k}{\lambda_k+\mu-\mu\gamma_k}$.

We define

$$f_k(x) = \frac{\mu}{\lambda_k}x^{k+1} - (1+\frac{\mu}{\lambda_k})x + 1.$$

Taking the first derivative of f_k yields

$$f_k'(x) = \frac{\mu(k+1)}{\lambda_k}x^k - (1+\frac{\mu}{\lambda_k}).$$

Then $\alpha_k^* = \left(\frac{1+\frac{\lambda_k}{\mu}}{(k+1)}\right)^{1/k}$ is the point where the polynomial $f_k(x)$ has a unique minimum.

Thus

$$\alpha_k \in \left(\frac{1}{1+\frac{\mu}{\lambda_k}}, \alpha_k^*\right).$$

For any $k\geq 1$, we have $\alpha_k^* = \left(\frac{1+\frac{\lambda_k}{\mu}}{(k+1)}\right)^{1/k} = \left(1-\frac{k}{k+1}(1-\rho)\right)^{1/k}$.

We define

$$g(k) = \left(1 - \frac{k}{k+1}(1-\rho)\right)^{1/k}.$$

We will show that $g(k)$ is stricly increasing in k.

$$g'(k) = g(k)\left(-\frac{1}{k^2}\log\left(1 - \frac{k}{k+1}(1-\rho)\right) - \frac{1-\rho}{k(k+1)(1+k\rho)}\right).$$

Since $g(k) = \alpha_k^* > 0$, it is sufficient to show that

$$-\frac{1}{k^2}\log\left(1 - \frac{k}{k+1}(1-\rho)\right) - \frac{1-\rho}{k(k+1)(1+k\rho)} > 0. \tag{6.47}$$

We know that: $\dfrac{1}{k+1} > \dfrac{1}{(k+1)(1+k\rho)}$, for $k \geq 1$.

This implies:

$$e^{-\frac{1-\rho}{(k+1)(1+k\rho)}} > e^{-\frac{1-\rho}{k+1}} > \left(1 - \frac{k}{k+1}(1-\rho)\right)^{1/k}$$
$$-\frac{1}{k^2}\log\left(1 - \frac{k}{k+1}(1-\rho)\right) > \frac{1-\rho}{k(k+1)(1+k\rho)}. \tag{6.48}$$

By combining Eq. (6.47) and (6.48) it is clear that $g(k)$ is stricly increasing in k. This means that $\alpha_k^* < \alpha_{k+1}^*$ and so $\alpha_{k+1}^* \uparrow 1$, for $k \to \infty$. Next, for any positive k and $x \in (0,1)$, the functions f_k and f_{k+1} have 2 intersection points: at 0 and 1. This follows from the relation:

$$f_{k+1}(x) - f_k(x) = \frac{1}{k(k+1)\rho}x(1-x)^2\sum_{i=1}^{k} ix^{i-1}.$$

This also implies that $f_{k+1}(x) > f_k(x)$ for all $x \in (0,1)$. Hence $\alpha_k < \alpha_{k+1} < 1$. In other words we can say that α_k is strictly increasing in k and the proof is complete. □

6.5.2 Comparison of batch service queues with Erlang arrivals versus a deterministic inter-arrival time

Let us assume for the moment that the batch size is identically equal to 1, i.e. $P\{Y = 1\}$. Consider the $D/M/1$-queue with mean inter-arrival time $1/\lambda^*$ and mean service time $1/\mu$, where $\lambda^* = \lambda_k/k$. In this case it is quite well-known that the stationary distribution of

the $E_k/M/1$-queue, with the same mean inter-arrival and mean service time distribution, converges setwise and weakly to the stationary distribution of the $D/M/1$-queue, as $k \to \infty$. We will generalize these results when there are batch services. By virtue of Asmussen [22] and Bhat [27] the stationary distribution $\bar{\pi}_D$ of the $D/M/1$-queue is given by

$$\bar{\pi}_m^D = \begin{cases} (1-\sigma)\rho\sigma^{m-1}, & m > 0 \\ 1-\rho, & m = 0, \end{cases}$$

where σ is the unique root smaller than 1 of

$$\sigma = e^{-(1-\sigma)/\rho},$$

when $\rho < 1$. By virtue of Theorem 6.39, this means that $\sigma = \gamma^* = \lim_{k\to\infty} \gamma_k$, and in particular $\sigma < \gamma_k$. It follows directly that:

$$\bar{\pi}^D \overset{st}{\preceq} \bar{\pi}^{k+1} \overset{st}{\preceq} \bar{\pi}^k, \quad k = 1, 2, \ldots \tag{6.49}$$

Define L_D, W_D, and V_D as the mean number of customers, the expected sojourn time, and the variance of customers of the $D/M/1$-queue respectively, under the stationary distribution. We now derive the result below.

Corollary 6.2. *The following are true:*

i) *Eq. (6.49) holds for all $k = 1, \ldots$, and*

ii) $L_k \downarrow L_D, \quad W_k \downarrow W_D, \quad$ *and* $\quad V_k \downarrow V_D, \quad$ *as* $\quad k \to \infty$, *monotonically.*

Intuitively it seems clear that a similar result holds as well for the $E_k/M^Y/1$- and $D/M^Y/1$-queues, with the same mean inter-arrival times and the same batch service distributions. So far, we have not been able to find any result on (setwise and weak) convergence of the stationary distribution of the $E_k/M^Y/1$-queue to the stationary distribution of the $D/M^Y/1$-queue, although it should be completely similar to convergence results in the case of batch size equal to 1.

General batch size

Suppose now again that $P\{Y = 1\} < 1$.

Theorem 6.7. *For the stationary distribution $\bar{\pi}^D$ of the $D/M^Y/1$-queue the following holds:*

i) $\bar{\pi}_m^k \to \bar{\pi}_m^D$, *for* $m = 0, \ldots$ *and*

ii)

$$\bar{\pi}_m^D = \begin{cases} \dfrac{\rho(1-\sigma)^2}{1-\phi_Y(\sigma)}\sigma^{m-1}, & m > 0, \\ 1 - \dfrac{\rho(1-\sigma)}{1-\phi_Y(\sigma)}, & m = 0, \end{cases} \tag{6.50}$$

where σ is the unique root smaller than 1 of

$$\sigma = e^{-(1-\phi_Y(\sigma))/\rho}, \quad \phi_Y(\sigma) = \sum_{j=1}^{b} p_j \sigma^j. \tag{6.51}$$

Proof. Since the proof is quite standard, we only provide a sketch of the proof.

First, notice that the embedded processes on arrival instants are of the GI/M/1-type considered in [57, pp. 82–86]. As has been derived there, it follows for the corresponding stationary distribution $\bar{\pi}_A^D$ of the $D/M^Y/1$-queue that

$$\bar{\pi}_{A,m}^D = (1-\sigma)\sigma^m,$$

with σ the unique root smaller than 1 of Eq. (6.51). Secondly, by using semi-regeneration, cf. Asmussen [22, Ch. VII.5 and p. 283], we then obtain formula Eq. (6.50) for the stationary distribution of the non-embedded $D/M^Y/1$-queue. The desired convergence properties then follow from the explicit formulae for the respective stationary distributions. □

It directly follows from Theorem 6.7(i) and Theorem 6.6 that the result of Corollary 6.2 holds in the batch service case as well.

Corollary 6.3. *The monotonicity result in Corollary 6.2 (ii) holds in the case of batch service.*

As a consequence, the mean number of customers, expected sojourn time and variance are minimized by deterministic inter-arrival times. This is a well-known result for non-batch systems (cf. Asmussen [22, pp. 336-339]). Results on other performance measures are given in this section as well.

6.6 Appendix

Lemma 6.5. *Let X be a possibly non-conservative, non-explosive, transient, stable Markov process on state space $\boldsymbol{S}$ with q-matrix Q. Then Q^{-1} exists and $Q^{-1} = T$ where the (i,j)-th element of T is defined as follows: $\tau_{i,j} = \int_0^\infty p_{t,(i,j)}dt < \infty$, where $p_{t,(i,j)}$ are the elements of the transition function P_t.*

Proof. A generalization of Lemma 6.1 and the proof is analogous. □

Lemma 6.6. *Let X be a possibly non-conservative, non-explosive, transient, stable Markov process on state space $\boldsymbol{S}$ with q-matrix Q. Let $s \in \boldsymbol{S}$ be given. Consider the transition rate matrix $\tilde{Q}$ on $X\backslash\{s\}$ where the elements are given by:*

$$\tilde{q}_{ij} = q_{ij} + q_{is}\frac{q_{sj}}{q_s},$$

where $q_s = -q_{ss}$. Let q_{ij}^{-1}, $\tilde{q}_{ij}^{-1}$ denote the (i,j)-th elements of matrices Q^{-1} and $\tilde{Q}^{-1}$, respectively. Then the following are true for $i, j \neq s$:

$$\tilde{q}_{ij}^{-1} = q_{ij}^{-1},$$
$$q_{sj}^{-1} = \sum_{r\neq s}\frac{q_{sr}}{q_s}\tilde{q}_{rj}^{-1}.$$

Proof. By Lemma 6.5 the inverse matrices Q^{-1} and $\tilde{Q}^{-1}$ exist, and the entries are equal to the expected sojourn time spent in each state of $\boldsymbol{S}$ and $\boldsymbol{S}_s = \boldsymbol{S}\backslash\{s\}$ respectively. It is convenient to use a representation based on the jump chain on $\boldsymbol{S}$ with transition matrix P^J. The entries of P^J are given by:

$$p_{i,j}^J = \frac{q_{i,j}}{q_i}, \text{ where } q_i = -q_{i,i}, \text{ for } i,j \in \boldsymbol{S}.$$

Clearly P^J is a sub-stochastic matrix, since Q is transient. We denote its n-th iterate by $P^{J,n}$. The 0-th iterate is the identity. It follows from Anderson [19], Proposition (4.1.1) that:

$$\tau_{i,j} = \sum_{n\geq 0} p_{i,j}^{J,n}\frac{1}{q_j}. \tag{6.52}$$

The jump transition probability matrix $\tilde{P}^J$ associated with $\tilde{Q}$ on $\boldsymbol{S}_s$, is given by

$$\tilde{p}_{i,j}^J = p_{i,j}^J + p_{i,s}^J p_{s,j}^J.$$

This is precisely the transition matrix of the jump chain associated with Q, *embedded on* $\boldsymbol{S}_s$. By using Eq. (6.52) and the fact that the (i,j)-th element of $-Q^{-1}$ represents the expected amount of time spent in state j, given a start in state i before absorption outside S, we directly obtain that $-\tilde{Q}^{-1}$ is the expected amount of time spent in $\boldsymbol{S}_s$ before absorption outside this set. This proves the first statement. For the second statement, we invoke Eq. (6.52). The proof is complete by noting that for $j \neq s$:

$$\tau_{s,j} = \sum_{r\neq s} p_{s,r}^J \tau_{r,j} = \sum_{r\neq s}\frac{q_{sr}}{q_s}\tau_{r,s}. \qquad \square$$

CHAPTER 7

Extensions

This chapter is based on work in progress: *Extensions to successive lumping*, cf. [S7], **and on** *Thinning methods.*

7.1 Introduction to Chapter 7

The general aim in this chapter is to show that the successive lumping method can be applied to a larger class of Markov processes than we originally identified in Chapter 2. In that chapter it is shown that by using successive lumping, the stationary distribution can be computed explicitly with respect to a partition that contains an entrance state for each level. We have shown in Chapter 3 how the successive lumping requirement of entrance states induces specific transition structures in QSF processes. Using this structure we can compute the rate matrix R that describes the relation between the stationary distribution of different levels explicitly. However, the requirements for the usage of successive lumping and for the exact computation of the rate matrix are restrictive. In the current chapter we will exploit several ways to relax the entrance state requirement. Besides the concept of exit states, introduced in the previous chapter (Chapter 6), we will consider possibilities to construct entrance states, using the specific transition structure of the Markov process. Furthermore, when the rate matrix R can be computed, it is not always straightforward how to compute the stationary distribution of a Markov process on an infinite sized state space. In Chapter 3 we do provide a procedure to estimate or bound this distribution. In the current chapter we will derive some new structural insights on the rate matrix and how it relates to rate matrices that are computed via different ways. The extensions we will exploit are briefly described below.

In Section 7.2 we consider QSF processes (see Chapter 3) that contain upward matrices of rank 1 (a column times row structure). In [85] a procedure is developed to compute the rate matrix when the upward (or downward) matrices have rank 1. However, this method is not always explicit, and does not readily extend to infinite state spaces or non-homogeneous processes. Since these QSF processes do not necessarily contain entrance states with respect to the level partition, we will show how to include 'artificial' entrance states in such a way

that the stationary distribution of this newly created process is the same as the one of the original process. As a consequence, all results in all previous chapters (such as efficiently computing the inverse in Chapter 5, etc.) can be readily be extended to processes where originally a partition is identified without entrance states, but where they can be constructed via the method in Section 7.2.

Second, in Section 7.3, we will show that for more complicated non-homogeneous QSF processes, it is possible to compute the rate matrix explicitly and to estimate the steady state distribution with the correct truncation on the highest level.

In Section 7.4, we will show a result for successively lumpable QBD processes: if the downward matrix is non-singular, we can reverse the direction of the rate matrix, see also Chapter 6. Instead of expressing the stationary distribution of lower levels in that of higher ones, we can construct this expression the other way around. We will extend this result and show that it can have an advantage. In this case, it can be easier to compute the stationary distribution than with other rate matrix constructions: we only need to compute the largest eigenvalue of the (inverted) rate matrix, which is the factor associated with the product form of the stationary distribution.

Finally, in Section 7.5, we will introduce a new method of separating transitions by constructing new processes. When doing so, Markov processes that do not meet the restrictions necessary to apply the successive lumping procedure, can meet these restrictions. The 'thinning' method described in this chapter can be applied to any irreducible Markov process, and also to a large class of transient Markov processes. Its usage does not depend on the transition structure. We will show that the combination of thinning and successive lumping can lead to a huge computational benefit.

To our knowledge this procedure has not been described before. Below we describe some existing, comparable methods.

In [52] a 'splitting' procedure is introduced. The main difference is that with splitting, the stationary distribution of the original process is a convex combination of that of the separate processes. However, the constructed processes have the same nonzero transitions as the original one, which is the main reason to introduce thinning; with this method we construct entrance states by eliminating transitions.
In [49] another 'transition eliminating algorithm' is provided, where a given point process is thinned by interactions with a second point process, see also [81]. This procedure depends more heavily on the structure of the process. In [47] transitions are removed for transient Markov processes. The thinning method we propose, is not restricted to Markov processes for which a successive lumping partition has been identified: it can be used on any Markov process.

For each proposed extension and alternative described in the different sections of this chapter, we identify an example that specifically fits the framework.

7.1.1 Preliminaries

In this chapter we consider Markov processes $X(t)$ where the transition rate matrix Q has in general the following structure:

$$Q = \begin{bmatrix} W_0 & U_0 & 0 & 0 & \cdots \\ D_1 & W_1 & U_1 & 0 & \cdots \\ 0 & D_2 & W_2 & U_2 & \cdots \\ 0 & 0 & D_3 & W_3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{7.1}$$

The matrices D_m, W_m and U_m correspond to level m and contain transitions to the next lower level, within the level and to the next higher level respectively. We refer to Chapter 3 for more information and notation on QSF and QBD processes. Throughout this chapter we assume that this q-matrix Q is irreducible and ergodic.
In the remainder of this chapter, the states of $\mathcal{X}$ are specified by tuples (m, i), i.e. $\mathcal{X} = \{(m, i) : m \geq 0, i \geq 1\}$ and m denotes the 'level' of the state and i denotes the 'phase' within the level. Latouche and Ramaswami, [85], have discussed QBD processes where matrix U or D has a column c times row r structure. In this chapter we consider QBD processes, in which the downward matrix D (no level dependency) has a $(c \cdot r)$-structure, containing transitions from level m to level $m + 1$. We will show that these processes are successively lumpable with respect to the level partition and the stationary distribution of the phases within a level has a product form as discussed in [85]. This result can be generalized for Quasi-Skipfree (QSF) processes, as we will show in this chapter.

To understand the scope of this chapter, we repeat the most important concept of successive lumping. Successive lumping is a procedure to calculate the steady state distribution of a Markov process exact and efficiently. To use this procedure, it is important to identify so-called entrance states. A set X has an entrance state i, if one-step transitions from states in X^c to X always end in i. A Markov process is called successive lumpable with respect to the state space partition $S = L_1, L_2, \ldots, L_N$ if set $\underset{\sim}{L}_n := \cup_{j=1}^{n} L_j$ has an entrance state for all $n = 1, 2, \ldots, N - 1$. The number of sets can be finite or infinite.

See Chapter 2 and 3 for more details on successive lumping, a formal definition of entrance states and DES processes. More specifically, we refer to Definition 2.1 and Lemma 3.1 and Lemma 3.2. For the usage of successive lumping to find rate matrices within the matrix analytic framework we refer to Chapter 3.

7.2 QBD processes with downward matrices of rank 1

In this section we consider a QBD process $X(t)$ on state space $\mathcal{X}$ with generator Q of the form described in the previous section. We will show that, when $X(t)$ has a matrix of rank

1 as downward matrices D_m, it can be altered to a successively lumpable Markov process with respect to the same level partition. The rank 1 requirement is equivalent to $D = c \cdot r$, with c a column vector and r a row vector of appropriate length. To satisfy the entrance state requirement we will add an extra entrance state with an arbitrary small sojourn time to each level.

For ease of exposition we assume homogeneity, and thus that the number of phases per level is constant, say equal to ℓ, where $\ell \geq 1$. Secondly, because of this homogeneity we denote: $U_m = U$, $W_m = W$ and $D_m = D$ for all $m \geq 1$. The results presented below extend to the non-homogeneous case, and to the computation of the first rate matrix R_1 that describes the relations between level 1 and 0.

By virtue of [85], the stationary distribution in level m can be expressed in terms of the stationary distribution of the states in level $m-1$ in terms of rate matrix R, where $R = U(-W - UG)^{-1}, G = \mathbf{1} \cdot r$, provided that the downward matrices D have a column times row structure, $c \cdot r$. The column vector identical to 1 is denoted by $\mathbf{1}$. We are going to compare the steady state distribution of $X(t)$ acquired by this procedure with the steady state distribution of the process described below.

We consider a process $X^\epsilon(t)$ on state space $\mathcal{X} \cup \{(m,0)\}_{m=1,2,\ldots}$, where states $(m,0)$ are artificial states, added to each level L_m. This process has the transition rate structure given below, where $\epsilon > 0$.

The within transitions are as follows:

$$w^\epsilon_{(m,i)(m,j)} = \begin{cases} r_j/\epsilon, & (m,i) = (m,0), \\ -1/\epsilon, & (m,i) = (m,j) = (m,0), \\ 0, & (m,j) = (m,0),\ i > 0, \\ w_{ij}, & \text{otherwise}, \end{cases}$$

The downward transitions are:

$$d^\epsilon_{(m,i)(m-1,j)} = \begin{cases} c_i, & (m-1,j) = (m-1,0), \\ 0, & \text{otherwise}, \end{cases}$$

And the upward transitions are:

$$u^\epsilon_{(m,i)(m+1,j)} = \begin{cases} 0, & (m,i) = (m,0), \\ u_{ij}, & \text{otherwise}. \end{cases}$$

Note that many of the transitions in $X^\epsilon(t)$ are the same as those in $X(t)$. The newly constructed state $(m,0)$ is an entrance state of the set $\underset{\sim}{L}_m$ as defined in the previous section. Therefore, we can use successive lumping to construct the rate matrix R^ϵ. Let π_m denote the (unique) stationary distribution of states in level m in process $X(t)$, for $m \geq 0$. Similarly, let π^ϵ_m denote the (unique) stationary distribution of states in level m in process $X^\epsilon(t)$,

for $m \geq 0$. In the theorem below we show that for $\epsilon \to 0$ the steady state probabilities corresponding to $X^\epsilon(t)$ will tend to that of the ones of process $X(t)$.

Theorem 7.1. *The following is true:*

$$\lim_{\epsilon \to 0} \pi^\epsilon_m = (0, \pi_m),$$

Proof. The stationary distribution π^ϵ_m satisfies the following recursion in terms of π^ϵ_{m+1}:

$$\pi^\epsilon_{m+1} = \pi^\epsilon_m R^\epsilon, \quad R^\epsilon = U^\epsilon(-B^\epsilon)^{-1}, \tag{7.2}$$

where $(-B^\epsilon)^{-1}_{ij}$ is the expected time spent in state (m, j) without passing through states in the sub-level set $\underline{L}_{m-1}$, given that the process starts in a state (m, i), for $m \geq 0$ and $i, j \in \{0, 1, \ldots, \ell\}$.

We can readily compute the elements of B^ϵ using successive lumping (Eq. (3.5)) as follows:

$$B^\epsilon = \begin{bmatrix} -1/\epsilon & r/\epsilon \\ U\mathbf{1} & W \end{bmatrix}.$$

The inverse of matrix B^ϵ can be derived using the Woodbury matrix identity, see e.g. [48]:

$$(B^\epsilon)^{-1} = \begin{bmatrix} -1/\epsilon & r/\epsilon \\ U\mathbf{1} & W \end{bmatrix}^{-1} = \begin{bmatrix} -\epsilon + \epsilon r(W + U(\mathbf{1} \cdot r))^{-1} U\mathbf{1} & r(W + U(\mathbf{1} \cdot r))^{-1} \\ \epsilon((W + U(\mathbf{1} \cdot r))^{-1} U\mathbf{1}) & (W + U(\mathbf{1} \cdot r))^{-1} \end{bmatrix}.$$

The matrix $W + U(\mathbf{1} \cdot r)$ has the same structure as the matrix under consideration in [85, Eq. (10)] and that one is known to be non-singular.

Furthermore for R^ϵ:

$$R^\epsilon = -\begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0}' & U \end{bmatrix} \cdot \begin{bmatrix} -\epsilon + \epsilon(rB^{-1}U\mathbf{1}) & rB^{-1} \\ \epsilon(B^{-1}U\mathbf{1}) & B^{-1} \end{bmatrix} = -\begin{bmatrix} 0 & \mathbf{0} \\ \epsilon K & UB^{-1} \end{bmatrix},$$

where $B = W + U(\mathbf{1} \cdot r)$ and $K = B^{-1}U\mathbf{1}$.

Thus in the limit:

$$\lim_{\epsilon \to 0} R^\epsilon = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0}' & -UB^{-1} \end{bmatrix}.$$

In [85, Theorem 2] the rate matrix R of a process with a $c \cdot r$ upward matrix is $R = U(-W - UG)^{-1}$ with $G = \mathbf{1} \cdot r$. This completes the proof. □

It is not required that the number of levels is bounded to the left or to the right. The result given above for processes that have a negative level bound, readily extends to all QBD-processes with $c \cdot r$ downward matrices. Therefore we conclude that to use the successive

lumping method to compute rate matrices, it does not matter whether the downward matrices have a single non-zero column or a $c \cdot r$ structure. In the latter case, we can construct process with downward matrices that have single non-zero columns.

7.2.1 Application: $PH/M/1$-queue

As an example of a queue with $c \cdot r$ downward matrices, we will consider a $PH/M/1$ queue. In this model, the inter-arrival times have a phase-type distribution of order ℓ with parameter λ_i denoting the arrival rate in phase $i \in \{1, \ldots, \ell\}$. At the end of phase i either a new customer arrives with probability p_{i0}, or phase k starts with probability p_{ik}, where $k = i+1, \ldots, \ell$ and $i \in \{1, \ldots, \ell\}$. A new customer starts in phase i with probability r_i upon arrival. Service times are exponential with rate μ.

The $PH/M/1$ queueing system can be formulated as a QBD process $X(t)$ on the state space $\mathcal{X} = \{\ldots, L_{-1}, L_0\}$, where $L_m = \{(m,1), ..., (m,\ell)\}$ for all $m \geq 0$. State (m,i) denotes that there are $-m$ customers in the system and that the arriving customer is in the i-th phase of its arrival.

The transition rate matrix Q for this model is given in Eq. (7.1) with $\ell \times \ell$ sub-matrices $U = \mu I$, $W = W_0 - \mu I$,

$$W_0 = \begin{bmatrix} -\lambda_1 & p_{12}\lambda_1 & p_{13}\lambda_1 & \cdots & p_{1\ell}\lambda_1 \\ 0 & -\lambda_2 & p_{23}\lambda_2 & \cdots & p_{2\ell}\lambda_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -\lambda_{\ell-1} & p_{\ell-1,\ell}\lambda_{\ell-1} \\ 0 & 0 & \cdots & 0 & -\lambda_\ell \end{bmatrix},$$

$$D = \begin{bmatrix} p_{10}\lambda_1 r_1 & p_{10}\lambda_1 r_2 & \cdots & p_{10}\lambda_1 r_\ell \\ p_{20}\lambda_2 r_1 & p_{20}\lambda_2 r_2 & \cdots & p_{20}\lambda_2 r_\ell \\ \vdots & \vdots & \ddots & \vdots \\ p_{\ell 0}\lambda_\ell r_1 & p_{l0}\lambda_\ell r_2 & \cdots & p_{\ell 0}\lambda_\ell r_\ell \end{bmatrix},$$

where $p_{i0} + \sum_{j=i+1}^{\ell} p_{ij} = 1$ and $\sum_{i=1}^{\ell} r_i = 1$.

Note that D has a $c \cdot r$ structure, with

$$c = \begin{pmatrix} p_{10}\lambda_1 \\ p_{20}\lambda_2 \\ \vdots \\ p_{\ell 0}\lambda_\ell \end{pmatrix} \quad \text{and } r = (r_1, r_2, \cdots, r_\ell).$$

The $PH/M/1$ queueing model therefore fits the requirements proposed in this section, nec-

essary to use the successive lumping approach to find the rate matrix R. We construct this rate matrix R using Equation (7.2), wherein:

$$R = \lim_{\epsilon \to 0} R^{\epsilon} = U^{\epsilon}(-B^{\epsilon}).$$

In the above we have:

$$B^{\epsilon} = \begin{bmatrix} -1/\epsilon & r/\epsilon \\ \mu\mathbf{1} & W \end{bmatrix}.$$

and the elements of U^{ϵ} can be computed by Eq. (7.2).

7.3 QSF processes

We can extend some of the results presented in Section 7.2 to a more general setting, the setting of QSF processes. In this section we will again assume that the downward matrix D has a $(c \cdot r)$-structure. Without loss of generality, we may assume that the process is skip free to the right (see Chapter 6), unbounded or bounded in both the negative and the positive level direction. The QSF structure implies that the jump rates are not allowed to cross more than one level in the downward direction. For a clear notation, we assume that the process is homogeneous. The infinitesimal generator Q is given as follows:

$$Q = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \iddots \\ \cdots & D & W & U^1 & U^2 & U^3 & \cdots \\ \cdots & 0 & D & W & U^1 & U^2 & \cdots \\ \cdots & 0 & 0 & D & W & U^1 & \cdots \\ \cdots & 0 & 0 & 0 & D & W & \cdots \\ \iddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

where the $l \times l$ sub-matrices D, W and U^s $(s = 1, 2, \ldots)$ represent the transition rates to the next lower level, the same level, and the s-th higher level respectively.

Interestingly, Theorem 7.1 can be similarly applied to QSF processes, where it was originally presented for QBD processes. The fact that levels can be 'skipped' in one direction, does not change the artificial construction of entrance states. We will not prove this statement here, since it is analogous to the original proof, but with more tedious notation.

7.3.1 Application: $PH/M^Y/1$-queue

We consider a queueing model in which customers arrive within a maximum of l exponentially distributed phases. This model is a generalisation of the $PH/M/1$-queue, described in Section 7.2.1. However, in the current model, the number of simultaneous served jobs occurs

according to a distribution induced by random variable Y. Customers are served in a batch of size j, where $1 \leq j \leq b$ with probability p_j. The state space description and the matrices D and W are equal to their respective counterparts in the $PH/M/1$ queue. The matrices U^j are of size $l \times l$, and defined as: $U^j = p_j \mu I$ for $j = 1, 2, \ldots b$. Since there is a level bound in the upward direction (no customers in the system), we define $U^{k'}$ as the upward matrix from level $-k$ to level zero, as follows: $U^{k'} = \sum_{i=k}^{b} p_i \mu I$.

Because D has a $c \cdot r$ structure, we can analyse this process as a DES process, combining the results Chapter 3 and Section 7.2 to find the rate matrices.

7.4 Non-singular rate matrices

In this section we will discuss a solution procedure to find the steady state distribution from the rate matrices when the QBD process is unbounded in the negative direction. In this section we will assume that the levels are of finite size.

When a QBD process with a $c \cdot r$ downward matrix has no negative level bound, we can not readily compute the stationary distribution using the rate matrices R_i. Using these rate matrices we can express the stationary distribution of a certain level in that of a lower level, but since there is no finite lowest level, we can not normalize without using a state space truncation. We propose a straight forward approach to circumvent this problem by reversing the direction of the rate matrices; instead of using the rate matrices we are interested in using their inverse. In that way, if the distribution on the highest level is known, we can compute the entire distribution. To find the stationary distribution on the highest level is still hard, although the rate matrices are directly computable. However, when the process is level homogeneous, it is formulated in [85] that the stationary distribution on the highest level can be computed using that the levels satisfy a product form solution. In this section we will show that the factor associated with this product form is an eigenvalue of R, and compare the results to rate matrices that are computed differently.

We consider a homogeneous QBD process $X(t)$ on levels $0, -1, -2, \ldots$. We use this negative level description to be able to interpreted $X(t)$ as a DES process, i.e. that there are entrance states in the downward direction. As stated above, the results regarding the construction of the inverse rate matrices also apply to non-homogeneous processes, by indexing the matrices W, U and D such that these matrices are level dependent.

Instead of the equation $\pi_{m+1} = \pi_m R$ we will investigate its inverse expression, for the levels with $m \leq -2$, provided that R is invertible:

$$\pi_m = \pi_{m+1} R^{-1}. \tag{7.3}$$

When matrix D is the product of a column and a row vector, we can compute the rate matrix $\bar{R}$ as is described in [85]. In that paper a different notation is used and in discrete time. It is

a rate matrix in the 'inverse direction', i.e. it can substitute R^{-1} in Eq. (7.3). This matrix is computed as follows:

$$\bar{R} = D(-W - \eta U)^{-1}. \tag{7.4}$$

In that paper it is argued that η is a solution of Equation (7.5) below:

$$g(\eta) = \eta, \tag{7.5}$$

where $g(\eta) = \xi c$ with $\xi = -r(W + \eta U)^{-1}$. Therefore the following holds for $m \leq -1$:

$$\pi_m = \pi_{m+1}\bar{R}, \quad \bar{R} = D(-W - \eta U)^{-1}.$$

Secondly, in that paper it is shown that for $m \leq -2$:

$$\pi_m = \pi_{-2}\eta^{-m-2}. \tag{7.6}$$

By virtue of Theorem 7.1 and Eq. (7.2) and when $\epsilon \to 0$, it is true that the following rate matrix solves Eq. (7.3), when U is non-singular (recall that $g = \mathbf{1} \cdot r$):

$$\pi_{m+1} = \pi_m R^{-1}, \quad R^{-1} = (-W - UG)U^{-1}. \tag{7.7}$$

To find the stationary distribution using this inverse rate matrix we do need to compute the relative stationary distribution on level 0. If the QBD process is level homogeneous, this can be done using η and normalisation (cf. [85]).

However, when the process has a non-homogeneous structure, or if the only quantity of interest is the rate matrix, then Eq. (7.7) provides a much faster procedure to derive the relation between different levels, since this rate matrix can be computed explicitly, without finding η.

Nevertheless, if we do want to compute the stationary distribution, the rate matrix computed as in Eq. (7.7) possesses an interesting property. In the theorem below we prove that the fixed points of Eq. (7.5) are in fact equivalent with the eigenvalues of matrix R^{-1}.

Theorem 7.2. *A fixed point of Eq. (7.5) is an eigenvalue of R^{-1} and ξ is the corresponding eigenvector.*

Proof. We will go into detail on the computation of η and relate it to the rate matrix computation done with successive lumping (Eq. (7.7)). We derive:

$$\begin{aligned}
g(\eta) - \eta &= -r(W + \eta U)^{-1}c - \eta \\
&= r(W + \eta U)^{-1}(W + U)\mathbf{1} - \eta \\
&= r(W + \eta U)^{-1}(W + \eta U)\mathbf{1} + r(W + \eta U)^{-1}(U - \eta U)\mathbf{1} - \eta \\
&= 1 - \eta + (1 - \eta)r(W + \eta U)^{-1}U\mathbf{1} \\
&= (1 - \eta)(1 + r(W + \eta U)^{-1}U\mathbf{1}),
\end{aligned}$$

where the second equality is true by $(W+U)\mathbf{1} = -c$, since the row sum of every row is 0, and the fourth equality since $r\mathbf{1} = 1$ by its definition.

Next, we will consider $(1 + r(W+\eta U)^{-1}U\mathbf{1})$. We introduce the factor $K_\eta = \frac{\det(U)}{\det(W+\eta U)}$, and write:

$$\begin{aligned} 1 + r(W+\eta U)^{-1}U\mathbf{1} &= K_\eta \frac{\det(W+\eta U)}{\det(U)}(1 + r(W+\eta U)^{-1}U\mathbf{1}) \\ &= K_\eta \frac{\det(W+\eta U + U\mathbf{1}r)}{\det(U)}. \end{aligned}$$

This holds by a variant of the well-known Sylvester's Determinant Theorem (originally published as [99] without proof, for a proof see for example [15]). From this theorem we derive the following general relations between the determinants of matrices:

$$\begin{aligned} \det(I + AB) &= \det(I + BA), \\ \det(X + kr) &= \det(X)(1 + rX^{-1}k), \\ \det(X + AB) &= \det(X)\det(I + BX^{-1}A) = \det(X)\det(I + ABX^{-1}), \end{aligned}$$

where A, B, X, k, and r are $m \times n$, $n \times m$, $m \times m$, $m \times 1$, and $1 \times m$ matrices, respectively. Matrix I is the identity matrix of appropriate dimension.

Using the same theorem, we next derive:

$$\begin{aligned} \frac{\det(W+\eta U + U\mathbf{1}r)}{\det(U)} &= \eta \frac{\det\left(U + \frac{W+UG}{\eta}\right)}{\det(U)} \\ &= \det\left(\eta I + (W+UG)U^{-1}\right). \end{aligned}$$

We can conclude that:

$$g(\eta) - \eta = (1-\eta)K_\eta \chi_R,$$

with χ_R the characteristic polynomial of matrix R^{-1}. This shows us that the solutions of Eq. (7.5) are the eigenvalues of matrix R^{-1}, as well as $\eta = 1$, which is a solution, but not an eigenvalue of R^{-1}. It is straightforward to check that ξ is an eigenvector. □

Note that a solution to the fixed point equation can not be computed easily. By using R instead of $\bar{R}$, we are able to compute η as an eigenvalue of the rate matrix R^{-1}. This matrix is independent of the parameter η. It is numerically easier (more tools exists) to find the eigenvalue of a matrix than to find the fixed point of a certain equation.

Therefore we suggest the following approach for four different representations of Markov process $X(t)$, regarding the computation of the rate matrix and the stationary distribution.

1. When $X(t)$ is non-homogeneous and U_n is non-singular, the rate matrix can be computed explicitly via Eq. (7.7) per level.

2. When $X(t)$ is non-homogeneous and U_n is singular, the rate matrices can be computed via Eq. (7.4), where η is the solution of Eq. (7.5) per level.

3. When $X(t)$ is homogeneous and U is non-singular, the rate matrix can be computed explicitly via Eq. (7.7) and the eigenvalue η of R^{-1} can be used to compute the stationary distribution.

4. When $X(t)$ is homogeneous and U is singular, the rate matrix can be computed indirectly by Eq. (7.4), where η is the solution of Eq. (7.5). Then using η, we can compute the stationary distribution by Eq. (7.6).

7.4.1 Application: $M/M/s/s$ retrial queue with impatient customers

As an example of why the above straightforward computation of the rate matrix is beneficial, we consider a queueing system with s servers and a buffer capacity of size s. Customers arrive according to a Poisson process with rate λ and upon encountering an available server, are served with rate μ. If a customers encounters that all servers are occupied, he will join an infinite capacity orbit. After an exponentially distributed time with rate β each of the orbit customers will retry to go to an empty server w.p. q or leave the system w.p. $1-q$, independently of the occupancy rate of the servers. We will use a two dimensional state space description where state (n, j) denotes that the number of customers in orbit equals $-n$ ($n = 0, -1, \ldots$) and that j ($j = 0, 1, \ldots, s$) servers are busy. Then the q-matrix is given as follows:

$$Q = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots \\ \cdots & W_{-3} & U_{-3} & 0 & 0 \\ \cdots & D & W_{-2} & U_{-2} & 0 \\ \cdots & 0 & D & W_{-1} & U_{-1} \\ \cdots & 0 & 0 & D & W_0 \end{bmatrix},$$

where the $(s+1) \times (s+1)$ sub-matrices D_n, W_n, $(n = 0, 1, 2, \ldots)$, and U represent the transition rates to the n-th lower level, the the n-th same level, and the next higher level respectively. These submatrices have the following form, where 0 denotes a zero-matrix of the proper dimension:

$$W_0 = \begin{bmatrix} -\lambda & \lambda & 0 & \cdots & 0 & 0 \\ \mu & -(\lambda+\mu) & \lambda & \cdots & 0 & 0 \\ 0 & 2\mu & -(\lambda+2\mu) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -(\lambda+(s-1)\mu) & \lambda \\ 0 & 0 & 0 & \cdots & s\mu & -(\lambda+s\mu) \end{bmatrix},$$

for $n \geq 1$:

$$W_{-n}=\begin{bmatrix} -(\lambda+n\beta) & \lambda & \cdots & 0 & 0 \\ \mu & -(\lambda+\mu+n\beta) & \cdots & 0 & 0 \\ 0 & 2\mu & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & s\mu & -(\lambda+s\mu+n(1-q)\beta) \end{bmatrix},$$

and

$$U_{-n} = \begin{bmatrix} n(1-q)\beta & nq\beta & 0 & \cdots & 0 & 0 \\ 0 & n(1-q)\beta & nq\beta & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & n(1-q)\beta & nq\beta \\ 0 & 0 & 0 & \cdots & 0 & n(1-q)\beta \end{bmatrix},$$

$$D = \begin{bmatrix} 0 & \ldots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & 0 \\ 0 & \ldots & 0 & \lambda \end{bmatrix}.$$

It is clear that matrix D has a $(c \cdot r)$-structure and U_{-n} is invertible for all $n \geq -1$, where

$$c = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \lambda \end{bmatrix}, \quad r = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}.$$

Let $\gamma := \frac{-q}{1-q}$. It is easy to check that the inverse of U_{-n} can be expressed as follows:

$$U_{-n}^{-1} = \frac{1}{n(1-q)\beta} \begin{bmatrix} 1 & \gamma & \gamma^2 & \cdots & \gamma^s \\ 0 & 1 & \gamma & \cdots & \gamma^{s-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

Since U_{-n} is non-singular for all $n \geq 1$, we can compute the rate matrix R_n^{-1} as:

$$R_{-n}^{-1} = (-W_{-n} - U_{-n}G)U_{-(n+1)}^{-1}, \text{ and thus:}$$

$$R_{-n}^{-1} = \begin{bmatrix} \lambda+n\beta & -\lambda & \cdots & 0 & -n\beta \\ -\mu & \lambda+\mu+n\beta & \cdots & 0 & -n\beta \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda+(s-1)\mu+n\beta & -\lambda-n\beta \\ 0 & 0 & \cdots & -s\mu & \lambda+s\mu+nq\beta \end{bmatrix} \cdot U_{-(n+1)}^{-1} \cdot$$

Using this approach we have directly described a relation between the stationary distribution of the different levels, that identify the number of customers in the orbit.

7.5 Thinning

When we consider a Markov process and find a certain partition $\mathcal{D}$ that satisfies the entrance state property described in Chapter 2, we can use successive lumping to compute the stationary distribution. However, consider for example the Markov process $X(t)$ depicted in Figure 7.1. We would like to perform successive lumping with respect to partition $\mathcal{D} = \{D_0, D_1, D_2, D_3\}$.

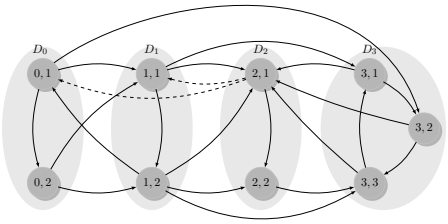


Figure 7.1: Transition diagram of Markov process $X(t)$

.

The two dashed transitions together violate the entrance state property of set $D_0 \cup D_1$: it is possible to enter this set via $(0,1)$ or $(1,1)$. Note that if one of those two transitions was absent, then $(0,1)$ or $(1,1)$ would be the entrance state of this set.

Below we will first describe the method we will propose to 'thin' *any* Markov process in words. An arbitrary subset S containing one-step transitions with their origin in a fixed state x is removed and a new process is constructed, without these transitions. Then we construct another process that contains the transitions in S, but does not include the one-step transitions from x that do not belong to set S. From the stationary distributions of these two newly constructed processes we compute the stationary distribution of $X(t)$. We will refer to this procedure as *thinning*, since the number of one step transitions is reduced in each of the separate Markov processes. The thinning method has no requirements concerning the

transition structure at all; it is possible to perform it on any irreducible Markov process. Also, it is not restricted to a division of one-step transitions into two sets; thinning allows to divide the transitions in an arbitrary number of sets.
In this section we first describe the thinning method in detail for an arbitrary Markov process, and then perform it on the Markov process $X(t)$ depicted in Figure 7.1 as an example.

7.5.1 Procedure

We consider an irreducible Markov process $X(t)$ with q-matrix Q on state space $\mathcal{X} = \{0, 1, 2, \ldots, M\}$, where M is finite or infinite. Note that this is an arbitrary Markov process, in general different from the one in the previous section(s). Without loss of generality we choose a state 0 and partition the one-step transitions leaving that state into T sets $S^1, \ldots, S^T$. We define $X^n(t)$ as the Markov process on $\mathcal{X}$ containing all transitions of $X(t)$, but *without* the transitions in $\cup_{m \neq n} S^m$. To make this statement rigorous, let Q^n, the generator of $X^n(t)$, be defined as follows:

$$q^n(i,j) := \begin{cases} q(0,0) + \sum_{(0,j) \notin S^n} q(0,j), & \text{if } (i,j) = (0,0), \\ 0, & \text{if } (i,j) = (0,j) \notin S^n, \\ q(i,j), & \text{otherwise.} \end{cases}$$

Note that Q^n is a non-transient generator matrix of the continuous Markov process $X^n(t)$. We let π^n denote the the steady state probability of this process. Furthermore, we let $\pi_i := \{\pi^1(i), \ldots, \pi^T(i)\}$ denote the vector containing the stationary probabilities of state i in all T different processes. Also, the vector with as its elements $\{\pi^1(i)^{-1}, \ldots, \pi^T(i)^{-1}\}$ will be denoted as $(\pi_i)^{-1}$. It is important to note the following.

Remark 7.1. The stationary probability of state 0 in process $X^n(t)$, $\pi^j(0)$, can not be zero: since $X(t)$ is irreducible, and only transitions *leaving* 0 are removed in $X^j(t)$ we can conclude that $\pi_j(0)$ still belongs to a recurrent class. Other states might get transient.

We now state the following theorem that provides a method to construct the stationary distribution $X(t)$ from the distributions of its 'thinned' processes. Let $\mathbf{1}$ be a vector identical to 1, and let $x \cdot y$ denote the inner product of vectors x and y.

Theorem 7.3. *The following is true for the stationary distribution π of $X(t)$, as a function of the stationary distributions of its T thinned processes $X^1, \ldots, X^T$.*

$$\pi(0) = \frac{1}{1 - T + (\pi_0)^{-1} \cdot \mathbf{1}},$$
$$\pi(i) = \frac{(\pi_0)^{-1} \cdot \pi_i}{1 - T + (\pi_0)^{-1} \cdot \mathbf{1}}, \text{ for } i = 1, 2, \ldots, M.$$

Proof. We will show that the vector π constructed as above is indeed the stationary distribution of $X(t)$. First, we note that both the numerator and the denominator are positive, which implies that $\pi(i) \geq 0$ for all i. Second, it is straightforward to check that $\pi \cdot \mathbf{1} = 1$, as follows:

$$\begin{aligned}\sum_{i=0}^{M} \pi(i) &= \frac{1 + \sum_{i=1}^{M} (\pi_0)^{-1} \cdot \pi_i}{1 - T + (\pi_0)^{-1} \cdot \mathbf{1}} \\ &= \frac{1 + (\pi_0)^{-1} \cdot (\sum_{i=1}^{M} \pi_i)}{1 - T + (\pi_0)^{-1} \cdot \mathbf{1}} \\ &= \frac{1 - T + (\pi_0)^{-1} \cdot \mathbf{1}}{1 - T + (\pi_0)^{-1} \cdot \mathbf{1}} = 1.\end{aligned}$$

Let $K^n := Q - Q^n$ and let B_0 denote the top row of a matrix B (where $B \in \{Q, Q^n, K^n\}$). Note that K_0^n is the only row of K^n containing non-zero elements. All other rows of K^n are identical to zero. We note the following relation between K_0^n and Q_0:

$$\sum_{n=1}^{T} K_0^n = \sum_{n=1}^{T} (Q_0 - Q_0^n) = TQ_0 - Q_0 = (T-1)Q_0. \tag{7.8}$$

Next, we will prove that $\pi Q = 0$, using that by their definition $\pi^n Q^n = 0$ for all n. First, note that $\pi Q = 0$ is equivalent to the following, by its construction in the theorem:

$$\left(1, (\pi_0)^{-1} \cdot \pi_1, (\pi_0)^{-1} \cdot \pi_2, \ldots, (\pi_0)^{-1} \cdot \pi_M\right) Q = 0.$$

We show that the above is true as follows, where δ is a vector of length M identical to zero but with a 1 as its first entry:

$$\begin{aligned}\left(1, (\pi_0)^{-1} \cdot \pi_1, \ldots, (\pi_0)^{-1} \cdot \pi_M\right) Q &= \left(\left((\pi_0)^{-1} \cdot \pi_0, \ldots, (\pi_0)^{-1} \cdot \pi_M\right) - (T-1)\delta\right) Q \\ &= \sum_{n=1}^{T} \frac{1}{\pi^n(0)} \pi^n (Q^n + K^n) - (T-1)Q_0 \\ &= \sum_{n=1}^{T} \frac{1}{\pi^n(0)} \pi^n K^n - \sum_{n=1}^{T} K_0^n \\ &= \sum_{n=1}^{T} \frac{1}{\pi^n(0)} \pi^n(0) \delta K^n - \sum_{n=1}^{T} K_0^n \\ &= \sum_{n=1}^{T} K_0^n - \sum_{n=1}^{T} K_0^n = 0.\end{aligned}$$

The third equality is true by Equation (7.8), and the proof is complete. □

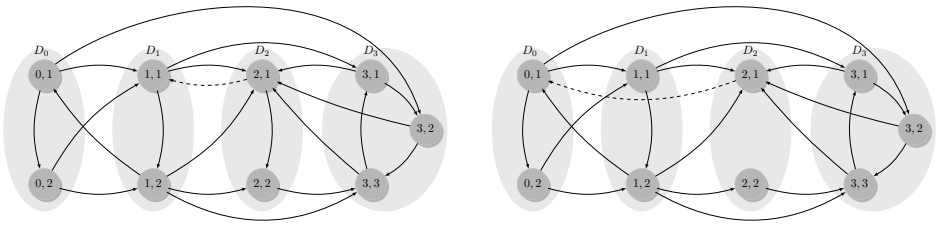
It is possible to apply this procedure repeatedly and to 'thin' the transitions around multiple states. However, this needs to be done sequentially; first we construct processes by thinning transitions around any state 0. Then we can consider *each* of these processes separately and possibly repeat the thinning around another state in all of these processes.

For example, suppose we want to perform thinning on a Markov process $X(t)$ around a state 0 and around a state 1, where we thin the transitions around 0 in two sets S^0 and S^1. To compute the stationary distribution of $X^1(t)$ we thin the transitions from 1 in classes V^1 and V^2 and create two processes in this way. The same has to be done for process $X^2(t)$, thus to compute the stationary distribution of $X(t)$ we consider 4 separate, thinned processes: one with transitions in $S^1 \cup V^1$, one with $S^1 \cup V^2$, one with $S^2 \cup V^1$ and one with transitions in $S^2 \cup V^2$, each one also containing all other transitions.

The thinning procedure by itself does not lead to a computational benefit. Instead of computing the stationary distribution of one Markov process, multiple Markov processes have to be considered. The method is in many cases a great addition to successive lumping, as we will show below. Also it can be used to analyse Markov processes of which a substantial number of states that will get transient by thinning transitions: the number of non-transient states reduces, and therefore computing the stationary distribution for each thinned process separately is faster.

7.5.2 Example

We take a closer look at the example provided in Figure 7.1. We apply the thinning procedure by creating two processes: $X^1(t)$ that contains the same transitions of $X(t)$ but not $(2,1) \to (0,1)$ and $X^2(t)$ that contains the transitions in $X(t)$ but not $(2,1) \to (1,1)$ and $(2,1) \to (1,2)$. The processes $X^1(t)$ and $X^2(t)$ are depicted in Figure 7.2 (a) and (b).



(a) Transition diagram of $X^1(t)$

(b) Transition diagram of $X^2(t)$

Figure 7.2: The two 'thinned' processes.

From the pictures we can derive that the set $\cup_{i=1}^{n} D_0$ has an entrance state for $n = 1, 2, 3$ in both processes. In $X^1(t)$ the set $D_0 \cup D_1$ has $(1, 1)$ as its entrance state and in $X^2(t)$ that entrance state is $(0, 1)$. Thus, we compute the stationary distributions of process $X^1(t)$ and $X^2(t)$ with successive lumping, and then use Theorem 7.3 to construct the stationary distribution of process $X(t)$.

CHAPTER 8

Shortest expected delay routing

This chapter is based on work in progress: *Shortest expected delay routing with arbitrary service rates*, [S8].

8.1 Introduction to Chapter 8

In this chapter we will thoroughly describe an application that fits the level dependent QSF process framework. In fact, we will examine a system that can be modelled as a DES process, introduced in Chapter 3.

Specifically, we consider a queueing system with two servers. Herein, one of the two servers has a higher service rate than the other one. We will denote the ratio between the rate of the faster server 2 and that of server 1 by $c > 1$. Both servers have exponentially distributed service times and an individual queue. Customers join the waiting line of the queue with the minimum expected delay: in our definition, expected delay includes the expected time of their own service. We will refer to this system as the 'Shortest Expected Delay'-routing model, or SED-routing. For example, in computer science this type of routing is used extensively in parallel computing.

SED-routing is an extension of 'Join the Shortest Queue'-routing (JSQ), wherein both servers work at the same speed. To construct the fastest routing policy, we would preferably want to take the workload of single jobs into account, however in many systems this information is not available. In that case, SED-routing provides an alternative when the server rates are known. In [93] (and references therein) it is argued that although SED-routing seems to be a natural choice, it does not minimize the mean stationary waiting time in general.

The analysis of the stationary behaviour of SED-routing is in general hard, and very little results regarding this specific model exists: most literature available handles JSQ-routing. In [41] an analysis has been conducted to establish structural results regarding the generating function of the stationary distribution. In [13], an SED-routing model is discussed where

the servers operate according to an Erlang distribution. The SED-routing model under consideration in [93] requires the ratio c between the two server speeds to be rational. In that paper, the compensation approach [14] is used to compute the stationary distribution. If c is an integer, a 3-dimensional state space description is necessary to satisfy the requirements of this approach. To allow a service rate ratio that is rational but not an integer, it is claimed in the paper that a 4-dimensional approach is necessary.

Extending the approach of [93] to irrational values of c might be possible by estimating the steady state distribution using a limiting sequence of rationals. However, we conjecture that there might be some technical difficulties in doing so: a small change in c might lead in certain states to different routing policies for an arriving job. Therefore constructing a converging limiting sequence is not straightforward at all, since the sequence of stationary distributions will be effected by these changes in routing. Also, it might be hard to produce numerical results when using that approach.

In contrast to the above, the assumption that c is irrational actually simplifies the analysis in this chapter. The situation that the expected delay is equal for both queues can not occur in this case: there are no states of the system in which a customer can choose between the waiting queues. The number of transitions reduces, and the model is easier to analyse. However, it is not a strictly necessary condition for the solution procedure provided in that paper. It is possible to construct a slightly different level partition when c is rational or when it is an integer. Since a solution method exists for those two cases, cf. [93], we will only provide a solution procedure when c is irrational.

In this chapter we will model SED-routing as a 2-dimensional Markov process. We identify a partition of the state space that allows the use of successive lumping (Chapter 2 and 3) to compute the stationary distribution. First we will formally define the model in Section 8.2, then we will show how successive lumping can be applied in Section 8.3. We will conclude with a numerical study in Section 8.5, using the truncated state space of Section 8.4.

8.2 Model description

Below we will formalize the system dynamics. Customers (jobs) arrive according to a Poisson process with rate λ and join the waiting line of server 1 or server 2. Server 1 has an exponentially distributed serving speed of rate μ_1 and server 2 one of rate μ_2. In the analysis below we describe the relation of the two rates by $\mu_2 = c\mu_1$ where $c > 1$ and irrational. Without loss of generality we will from now on assume that $\mu_1 = 1$ and $\mu_2 = c$. Customers join (are assigned to) the waiting line that has the shortest expected waiting time, including their own expected service time.

The expected waiting time for a customer joining queue 1 when that queue is of length i equals $i + 1$. The expected waiting time in queue 2 when it is of length j equals $(j + 1)/c$.

In SED-routing, arriving customers will join queue 1 when $c(i+1) < j+1$, and queue 2 when $c(i+1) > j+1$. Note that equality cannot occur, since c is irrational.

In [35] it is argued that the system is stable when:

$$\rho = \frac{\lambda}{\mu_1 + \mu_2} = \frac{\lambda}{1+c} < 1. \tag{8.1}$$

We will assume that this ergodicity assumption (8.1) holds for the remainder of this chapter.

We model this queueing system with SED-routing as a Markov process $X(t)$ on a two dimensional state space $\mathcal{X} := \{(i,j) | i,j \geq 0\}$ where state (i,j) denotes the condition of the system with i customers in queue 1 and j customers in queue 2. The transitions are specified by generator matrix Q where its $(i,j),(k,l)$-th entry, denoted by $q_{(i,j),(k,l)}$, contains the transition from (i,j) to (k,l). As a result of the system dynamics described above, it is straightforward to check that these entries have the following form when $(i,j) \neq (k,l)$:

$$q_{(i,j),(k,l)} = \begin{cases} 1, & \text{if } (k,l) = (i-1,j), \\ c, & \text{if } (k,l) = (i,j-1), \\ \lambda, & \text{if } (k,l) = (i+1,j) \text{ and } j+1 > c(i+1). \\ \lambda, & \text{if } (k,l) = (i,j+1) \text{ and } j+1 < c(i+1). \\ 0, & \text{otherwise.} \end{cases} \tag{8.2}$$

The diagonal elements of Q are as follows:

$$q_{(i,j),(i,j)} = \begin{cases} -\lambda & \text{if } i = j = 0. \\ -\lambda - c, & \text{if } i = 0 \text{ and } j > 0. \\ -\lambda - 1, & \text{if } j = 0 \text{ and } i > 0. \\ -\lambda - 1 - c, & \text{otherwise.} \end{cases} \tag{8.3}$$

A transition diagram for the specific case of $c = \pi/2$ is displayed in Figure 8.1. The arrows represent the non-zero one-step transitions. The number of customers in queue 1 is depicted as the x-variable, the number of customers in queue 2 as the y-variable. This figure points out that arriving customers join the waiting line of the faster server 2 when the system is in a state *below* the line $j = c(i+1) - 1$, and join the waiting line of the slower server 1 when the system is in a state *above* that line.

8.3 Successive lumping

We want to compute the stationary distribution of the system above using successive lumping. To do so, we first construct a specific level partition $\mathcal{L}$ of the states in $\mathcal{X}$. Next, we will

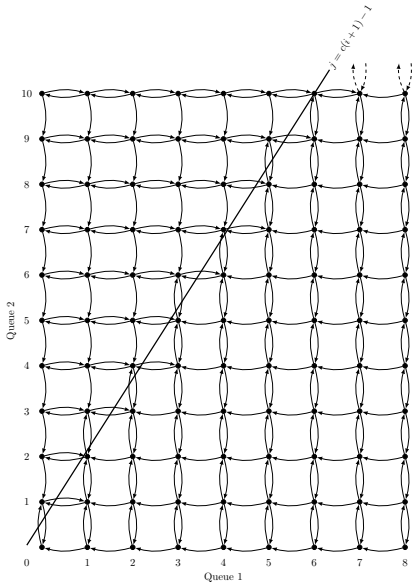


Figure 8.1: Transition Diagram of SED-routing with $c = \pi/2$.

describe how the matrices that contain the transitions between the various levels of this level partition are set up. When these matrices are formally defined, we will show that $X(t)$ is a DES process with respect to $\mathcal{L}$ and construct the rate matrices, in line with Algorithm 3.1 of Chapter 3.

8.3.1 Establishing the level partition

We will describe how to identify a specific level partition of the states in state space $\mathcal{X}$. To do so, we first introduce the model specific concept of *horizontal* and *vertical* sets. As the name suggests, a vertical set has a vertical 'shape' in the grid of Figure 8.1. In a vertical set the first entry, i.e. the number of customers in queue 1, is constant. Analogously we define a horizontal set: in such a set the number of customers in queue 2 is constant. Later we will show how these sets relate to a level partition. We formalize the description of vertical sets V_i and horizontal sets H_j in the definition below by defining their sizes.

Definition 8.1. For $i = 1, 2, \ldots,$, we define vertical set V_i as:

$$V_i := \{(i, 0), (i, 1), \ldots, (i, m)\},$$

where $m = \lfloor c \cdot i \rfloor$.

For $j = 1, 2, \ldots$, we define horizontal set H_j as:

$$H_j := \{(0, j), (1, j), \ldots, (n, j)\},$$

where $n = \lfloor j/c \rfloor$.

To show that a partition in these horizontal and vertical sets is exhaustive in the state space $\mathcal{X}$, without state $(0, 0)$, we state the following lemma.

Lemma 8.1. *When c is irrational, the construction in Definition 8.1 assigns every state in $\mathcal{X}\backslash\{(0, 0)\}$ to precisely one set, horizontal or vertical.*

Proof. We will show that all states $(i, j) \in \mathcal{X}\backslash\{(0, 0)\}$ belong to at most one set and at least one set. It is clear that by construction all states belong to at most one vertical set V_i and to at most one horizontal set H_j. Suppose that state $(i, j) \in \mathcal{X}\backslash\{(0, 0)\}$ belongs to both vertical set V_i and to horizontal set H_j. Then $j \leq \lfloor c \cdot i \rfloor < c \cdot i$ and $i \leq \lfloor j/c \rfloor < j/c$, which is a contradiction. The strict inequalities are true since c is irrational. Thus (i, j) belongs to at most one set.

Suppose that $(i, j) \in \mathcal{X}\backslash\{(0, 0)\}$ is not an element of either V_i or H_j, then by the same reasoning as above; $j \geq \lceil c \cdot i \rceil > c \cdot i$ and $i \geq \lceil j/c \rceil > j/c$, which also leads to a contradiction, thus (i, j) belongs to at least one set. The proof is complete. □

Since state (i, j) belongs to V_i or to H_j we state the following as a result of the lemma above for all $(i, j) \in \mathcal{X}\backslash\{(0, 0)\}$:

$$(i, j) \in \begin{cases} V_i, & \text{if } j \leq \lfloor c \cdot i \rfloor, \\ H_j, & \text{otherwise.} \end{cases}$$

We want to stress out that the size of the sets V_i and H_i depends on the value of service rate c. Figure 8.2 shows the transition diagram of SED-routing with $c = \pi/2$ as was displayed in Figure 8.1, but with the addition of vertical and horizontal sets, visible by a grey background. Figure 8.3 also shows SED-routing, but now with $c = \pi$; the arrival process is different for this parameter value of c. Note that the horizontal and vertical sets have different sizes than in Figure 8.2.

Next we define an ordering in the vertical and horizontal sets. To do so, we introduce *levels* L_n, with $n = 0, 1, 2, \ldots$. These levels will represent either a vertical or a horizontal set, following the allocation in the definition below.

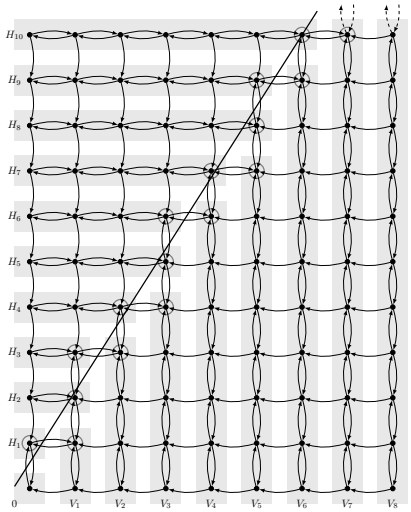


Figure 8.2: Transition diagram with horizontal and vertical sets for $c = \pi/2$.

Definition 8.2. We define the levels $L_n,\ n = 0, 1, 2, \ldots$ as follows:

$$L_n = \begin{cases} \{(0,0)\}, & \text{if } n = 0, \\ V_{\lfloor \frac{n+1}{c+1} \rfloor}, & \text{if } \lfloor \frac{n+1}{c+1} \rfloor > \lfloor \frac{n}{c+1} \rfloor, \\ H_{\{n - \lfloor \frac{n}{c+1} \rfloor\}}, & \text{otherwise.} \end{cases} \tag{8.4}$$

In the lemma below we show that by using this allocation, all sets are assigned to a level, and vice versa. This implies that all states of $\mathcal{X}$ are assigned to a level.

Lemma 8.2. *The allocation in Definition 8.2 assigns every set to precisely one level.*

Proof. It is easy to check that we can construct the unique inverse allocation of the one described in Eq. (8.4) as follows:

$$V_i = L_{\{i + \lfloor c \cdot i \rfloor\}},$$
$$H_j = L_{\{j + \lfloor j/c \rfloor\}}.$$

This allocation is well defined and proves that every set is assigned to precisely one level. □

Note that with the allocation described in Definition 8.2 the vertical sets are allocated to levels in a strictly increasing manner: a higher indexed vertical set implies a higher indexed level. The same relation holds for horizontal sets. For the specific example of $c = \pi/2$ the

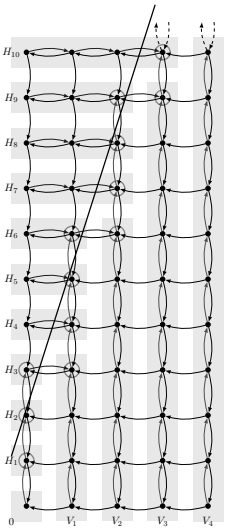


Figure 8.3: Transition diagram with horizontal and vertical sets for $c = \pi$.

level-ordering of the sets is (Figure 8.2):

$$L_0, L_1, L_2, \ldots = \{(0,0)\}, H_0, H_1, V_1, H_2, H_3, V_2, H_4, V_3, H_5, H_6, V_4, \ldots$$

Since all states are assigned to precisely one level, we denote $\mathcal{L} := \{L_0, L_1, L_2, \ldots\}$ as the exhaustive level partition.

8.3.2 Defining the transition sub-matrices

In this section we will show what the shape of generator matrix Q is, when using the ordering provided by level partition $\mathcal{L}$. We already identified the separate elements of Q in Eq. (8.2) and (8.3), but not their ordering within Q. We will order the states according to the levels, defined in the previous section. Within a level the states are ordered from lower to higher index as follows. If level L_n is vertical set V_i, then its states are ordered as $(i,0), (i,1), \ldots (i,m)$ and if level L_n is horizontal set H_i, then its states are ordered as $(0,j), (1,j), \ldots (n,j)$.

In general, the matrix Q will have the form displayed in Eq. (8.5), according to level partition $\mathcal{L}$. Herein, 0 denotes a matrix identical to zero of the appropriate dimension. The matrices W_n, U_n and $D_{n,m}$ denote the matrices containing transitions within, to higher and to lower levels respectively.

We will describe the shape of the sub-matrices of Q containing positive transitions. These

structures are directly derived from the definition of the SED-routing model, the provided level partition $\mathcal{L}$ and the elements of Q.

$$Q = \begin{bmatrix} W_0 & U_0 & 0 & 0 & 0 & 0 & \cdots \\ D_{1,0} & W_1 & U_1 & 0 & 0 & 0 & \cdots \\ D_{2,0} & D_{2,1} & W_2 & U_2 & 0 & 0 & \cdots \\ D_{3,0} & D_{3,1} & D_{3,2} & W_3 & U_3 & 0 & \cdots \\ D_{4,0} & D_{4,1} & D_{4,2} & D_{4,3} & W_4 & U_4 & \cdots \\ D_{5,0} & D_{5,1} & D_{5,2} & D_{5,3} & D_{5,4} & W_5 & \cdots \end{bmatrix}. \tag{8.5}$$

We first describe the matrices that contain transitions from states to other states *within* the same level. It is straightforward to check that these matrices W_n are of size $|L_n| \times |L_n|$ and have the following form, for $n = 1, 2, \ldots$:

$$W_n = \begin{bmatrix} -\lambda - \mu_{i(n)} & \lambda & \ldots & 0 & 0 \\ \mu_{j(n)} & s & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & s & \lambda \\ 0 & 0 & \ldots & \mu_{j(n)} & s \end{bmatrix}.$$

Herein, the service rates $\mu_{i(n)}$ and $\mu_{j(n)}$ depend on the value of n, that uniquely defines whether L_n is a horizontal or a vertical set, see Definition 8.2. These functions are defined as follows:

$$\begin{aligned} i(n) &= \begin{cases} 1, & \text{if } L_n = V_k \text{ for some } k, \\ 2, & \text{if } L_n = H_k \text{ for some } k, \end{cases} \\ j(n) &= \begin{cases} 1, & \text{if } L_n = H_k \text{ for some } k, \\ 2, & \text{if } L_n = V_k \text{ for some } k. \end{cases} \end{aligned} \tag{8.6}$$

For completion, note that $W_0 = [-\lambda]$.

Next, we define the upward matrices U_n containing transitions from states in level n to level $n+1$. Figure 8.2 shows that transitions to a higher level are sparse. Every level n has precisely one state (the one with the highest index) that has a positive one-step probability to a state (again the one with the highest index) in level $n+1$. The transition rate associated with this transition is λ.

The up matrices are therefore of size $|L_n| \times |L_{n+1}|$ and have the following structure, in accordance with Eq. (8.2) and (8.3):

$$U_n = \begin{bmatrix} \ddots & \vdots & \vdots \\ \ldots & 0 & 0 \\ \ldots & 0 & \lambda \end{bmatrix}.$$

The matrices that are the most difficult to describe are the ones containing transitions to levels with a lower index.

Recall the matrices $D_{n,m}$, containing the transitions from states in level L_n to states in level L_m, with $m < n$. These matrices are of size $|L_n| \times |L_m|$ and can have different representations, depending on whether L_n and L_m correspond to horizontal or vertical sets.

When taking a closer look at the transition diagrams for a given value of c (Figure 8.2 and Figure 8.3) we observe the following. From each vertical set, there are downward transitions to the previous, lower indexed vertical set. The same statement can be made for horizontal sets. Besides downward transitions to sets of the same type, there are transitions to sets of the other type. These dynamics are as follows. From horizontal sets there are either no transitions to vertical sets or except at most a single transition to a single state. From a vertical set, there can be several downward transitions to horizontal sets: a single transition to each horizontal set with an higher index than the previous vertical set. As the above shows, it is hard to efficiently describe how the downward transitions behave. This behaviour heavily depends on the value of c. It is completely and correctly described by Eq. (8.7) below. To do so, we first need to define the following matrices $X_{n,m}$ and $Y_{n,m}$ that are specific instances of downward matrices, all of size $|L_n| \times |L_m|$. The identity matrix of size $|L_m|$ is denoted by $I_{|L_m|}$:

$$X_{n,m} = \mu_{i(n)} \begin{bmatrix} I_{|L_m|} \\ \hline \mathbf{O}_{(|L_n|-|L_m|)\times|L_m|} \end{bmatrix},$$

and

$$Y_{n,m} = \begin{bmatrix} \ddots & \vdots & \vdots \\ \cdots & 0 & 0 \\ \cdots & 0 & \mu_{i(n)} \\ \hline \mathbf{O}_{(n-m-1)\times|L_m|} \end{bmatrix},$$

where $\mathbf{O}_{a\times b}$ is a zero-matrix of size $a \times b$. In the statements of the matrices above, $i(n) = 1$ if $L_n = V_k$ and $i(n) = 2$ if $L_n = H_k$ for some k as in Eq. (8.6).

Let $n(V_k)$ denote the index of the level corresponding to a vertical set V_k. The downward matrices $D_{n,m}$ are defined as follows.

$$D_{n,m} = \begin{cases} X_{n,m}, & \text{if } L_n = H_k \text{ and } L_m = H_{k-1}, \\ X_{n,m}, & \text{if } L_n = V_k \text{ and } L_m = V_{k-1}, \\ Y_{n,m}, & \text{if } L_n = H_k \text{ and } L_{n-1} = V_l, \\ Y_{n,m}, & \text{if } L_n = V_k \text{ and both } L_m = H_l \text{ and } n(V_{k-1}) < m, \\ \mathbf{O}_{|L_n|\times|L_m|}, & \text{otherwise.} \end{cases} \tag{8.7}$$

Corollary 8.1. *Let $M := \lceil c \rceil + 1$. For every level n there are at most M non-zero downward matrices $D_{n,m}$.*

Proof. From a horizontal set there are either 1 or 2 non-zero downward matrices by Eq. (8.7). There are either $M-1$ or M non-zero downward matrices from a vertical set $L_n = V_k$: one to the previous vertical set $V_k - 1$ and to all of the $n - n(V_{k-1})$ horizontal sets in between these two. This number is at most $\lceil c \rceil$. □

For the specific case that $c = \pi/2$, Q is as in Eq. (8.8). As described, a fixed choice of c defines which downward matrices are identical to 0. The shape of the matrices not equal to zero is uniquely determined from the results above.

$$Q = \begin{bmatrix} W_0 & U_0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ D_{1,0} & W_1 & U_1 & 0 & 0 & 0 & 0 & \cdots \\ D_{2,0} & D_{2,1} & W_2 & U_2 & 0 & 0 & 0 & \cdots \\ 0 & D_{3,1} & D_{3,2} & W_3 & U_3 & 0 & 0 & \cdots \\ 0 & 0 & 0 & D_{4,3} & W_4 & U_4 & 0 & \cdots \\ 0 & 0 & D_{5,2} & D_{5,3} & D_{5,4} & W_5 & U_5 & \cdots \\ 0 & 0 & 0 & 0 & D_{6,4} & D_{6,5} & W_6 & \cdots \end{bmatrix}. \tag{8.8}$$

In Figure 8.4 we have displayed the transition diagram of an SED-routing model with $c = \pi/2$, but sorted according to the levels. The states with the lowest indeces are displayed on top of each level, the states with the highest indeces at the bottom. This diagram undoubtedly indicates that upward transitions only occur from one state in each level. The within transitions follow a birth-and-death structure. The downward transitions on the other hand behave irregularly and depend on whether a level belongs to a vertical or horizontal set, and if its predecessors are horizontal or vertical sets.

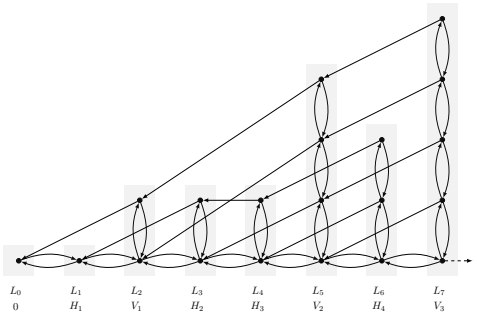


Figure 8.4: Transition Diagram sorted by levels for $c = \pi/2$.

Now that we have identified the level partition $\mathcal{L}$ and constructed the matrices describing the transitions between levels, we will go into more detail on the specific structure induced by this partition. In the lemma below we show that the super-level sets of $\mathcal{L}$ (see Definition 3.2) contain entrance states, according to their introduction given in Definition 2.1.

Lemma 8.3. *The set $\cup_{n \geq N} L_n$ has an entrance state for all N.*

Proof. See Lemma 3.1 of Chapter 3, with this difference that we consider QSF process to the right, instead of to the left: in the SED-routing model, entrance states exists for super-level sets instead of for sub-level sets. This is induced by the structure of the upward matrices; they have a single non-zero column. □

When taking a closer look at Figure 8.2 or Figure 8.3, it is straightforward to check that the entrance states are the encircled states. It is not possible to enter a level in any other state than the entrance state, if the process is in a state belonging to the lower indexed finite subset of the state space.

Using the above lemma concerning entrance states, we formally state that the SED-routing model is successively lumpable with respect to $\mathcal{L}$ in the theorem below.

Theorem 8.1. *The Markov process $X(t)$ is successively lumpable with respect to $\mathcal{L}$.*

Proof. A direct result of Lemma 8.3 combined with the work of Chapter 2 and 3, specifically Lemma 3.2. Indeed, $X(t)$ is a DES process (with a negative level numbering), according to its definition in Chapter 3. For more details we refer to those chapters. □

8.3.3 Constructing the rate matrices

In this section we will describe how to compute the rate matrices that describe the relation between the stationary distributions of the different levels.

Since we derived that $X(t)$ is successively lumpable with respect to $\mathcal{L}$ we can compute the rate matrices according to (because of notation change, slightly modified versions of) Eq. (3.10) and (3.11). There are some differences in usage to the formulas given in that section. One of the differences with those calculations is that we consider a process that is skip free to the right instead of to the left. Therefore the role of the downward and upward matrices interchanges. Second, there are at most M non-zero down matrices per level. Hence, instead of constructing rate matrices that describe a relation between the stationary distribution of the current level n and all higher levels, these matrices describe a relation between the stationary distribution of level n and that of M higher levels. Finally, the entrance states of the super-level sets are the states with the highest index, instead of the state with the lowest index.

Specifically, let π_n be the vector of size $|L_n|$ that denotes this stationary distribution of the states belonging to level n and a vector $\widetilde{\pi}_{n+k} := \{\pi_{n+k}, \pi_{n+k+1}, \ldots, \pi_{n+k+M-1}\}$. Then matrix R_n^k describes their relation as follows:

$$\pi_n = \widetilde{\pi}_{n+k} R_n^k.$$

Below we will construct the rate matrices R_n^k of size $(\sum_{m=0}^{M-1} |L_{n+k+m}|) \times |L_n|$ by describing them in terms of the matrices U, W and $D_{n,m}$, derived in the previous section.

First, the one-step rate matrices R_n^1 have the following form (see Eq. (3.11)):

$$R_n^1 = -A_n (B_n)^{-1}.$$

Matrix B_n is of size $|L_n| \times |L_n|$ and is constructed as follows:

$$B_n = W_n + \mu_{3-i(n)} \mathbf{1} \delta,$$

where $i(n)$ is as in Eq. (8.6). Furthermore, $\mathbf{1}$ denotes a column vector identically equal to 1 and $\delta = (0, \ldots, 0, 1)$, both of appropriate size. Matrix A_n is of size $|\sum_{m=1}^{M} L_{n+m}| \times |L_n|$ and defined as follows:

$$A_n = \begin{bmatrix} D_{n+1,n} + \sum_{m=0}^{n-1} D_{n+1,m} \mathbf{1} \delta \\ \vdots \\ D_{n+M,n} + \sum_{m=0}^{n-1} D_{n+M,m} \mathbf{1} \delta \end{bmatrix}$$

The inverse of B_n can be computed efficiently using the algorithm provided in Chapter 5 and takes $\mathcal{O}(|L_n|^2)$ arithmetical steps.

When the 'one-step' rate matrices R_n^1 are known, we can compute the rate matrices corresponding to larger steps, $k \geq 2$ for all n as follows:

$$R_n^k = \left[R_{n+k-1}^1 | \tilde{I}_{n+k} \right] \hat{R}_n^{k-1},$$

where $\tilde{I}_{n+k}$ is an identity matrix of size $\sum_{m=0}^{M-1} |L_{n+k+m}|$ and:

$$\hat{R}_n^{k-1} = \begin{bmatrix} R_n^{k-1} \\ \mathbf{O}_{|L_{n+M+k-1}| \times |L_n|} \end{bmatrix}.$$

This calculation corresponds to Eq. (3.10), but with changed direction and corrected for the fact that M is a finite number.

8.4 Truncation method

The rate matrices constructed in the previous section contain the relation between the stationary distribution of a certain level with its M subsequent higher levels. However, the number of levels is unbounded, i.e. there is no level with a maximum index. Since there is no maximum level, then we can not normalize to compute the stationary distribution, but need to apply a truncation method to do so.

Formally stated, we define a truncated process $X^N(t)$ on state space $\mathcal{X}^N = \{L_0, \ldots, L_N\}$. The transition structure Q^N of $X^N(t)$ is identical to matrix Q belonging to $X(t)$, unless described otherwise. All sub-matrices of Q^N are denoted with a superscript N.

When we perform a state space truncation at level N, there is only a single transition in the original process that has his origin in $\mathcal{X}^N$ and goes to a state in $\mathcal{X}\backslash\mathcal{X}^N$. This transition has rate λ.

The truncation method that we will apply works as follows: we 'remove' the upward matrix U_N and keep the remaining structure in tact. To make the process stable, we define $W_N^N = W_N$, except for the lower left element on the diagonal, i.e.:

$$W_N^N = W_N + \begin{bmatrix} 0 & \ldots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \ddots & 0 & 0 \\ 0 & \ldots & 0 & \lambda \end{bmatrix}. \tag{8.9}$$

Using this truncation makes Q^N a stable transition matrix, and $X^N(t)$ non-transient. This is in line with the truncation method of Remark 3.6; we refer to that remark and the remainder of Section 3.3.1 for more information on state space truncations for DES processes.

Note that the rate matrix computation of the previous section does not change. Only a slight adjustment needs to be made to take into account that for the highest indexed levels, there are not M higher levels anymore. So now: $\widetilde{\pi}_{n+k} = \{\pi_{n+k}, \pi_{n+k+1}, \ldots, \pi_{\max(n+k+M-1,N)}\}$, and the rate matrices change accordingly. Also, we emphasize that the computation of the rate matrices described in the previous section is recursive and can be readily extended for higher truncation levels.

There are various other possibilities to truncate the state space: one could think of methods that exploit the birth-and-death structure of the levels outside $\mathcal{X}^N$ to estimate the return probabilities to states in $\mathcal{X}^N$. Doing so, many variations are possible to make Q^N stable. One could for example consider only all positive return probabilities to level N or also to level $N-1$ if this is a set in the other orientation (vertical or horizontal).

However as numerical results will show in the next section, the stationary distribution converges (rapidly) when using the straightforward truncation method as described above in Eq. (8.9).

8.5 Numerical Analysis

In this section we will analyse and compare numerical results of several instances of the SED-routing model. To do a reasonable analysis we use a finite truncation at level N, described in the previous section and we construct the rate matrices as in Section 8.3.3.

First we will compute the stationary distribution for relatively small state spaces, induced by a low truncation level. Specifically, we consider a system where we have used parameter values $c = \pi/2$ and $\lambda = 1.5$, and truncation level $N = 10$. The results are provided in Table 8.1 below. The lines in Table 8.1 represent the boundaries of the various levels, identical to the level partition displayed in Figure 8.2. For example, the number in the lower left corner (0.2621) represents the stationary probability that the system is empty. Note that the system spends a lot of time in and near the 'entrance states' (see Lemma 8.3). This is to be expected, since a certain drift exists towards those states, as Figure 8.4 shows.

0.0001	0.0003	0.0009	**0.0033**	0.0027
0.0003	0.0007	0.0022	**0.0063**	0.0012
0.0018	0.0051	**0.0169**	**0.0107**	0.0006
0.0166	**0.0482**	**0.0314**	0.0051	0.0003
0.0371	**0.0879**	0.0158	0.0025	0.0001
0.1927	**0.1403**	0.0085	0.0013	0.0001
0.2621	0.0904	0.0057	0.0008	0.0000

Table 8.1: Stationary distribution with $c = \pi/2$, $\lambda = 1.5$ and $N = 10$

Second, in Table 8.2 we have repeated the computation but with a higher truncation level: in this case $N = 15$. It stands out that the stationary distribution corresponding to this system with a higher truncation level is very similar to the one where $N = 10$, even with these small state spaces. Convergence is fast, especially when the system is not heavily loaded, as is the case with these parameter values.

Next, we will consider truncations at (much) higher levels. For ease of exposition, we will only display the average queue lengths EL_1 and EL_2 for different parameter instances. Table 8.3 displays the results when server 2 operates with rate $c = \pi/2$ for various truncation levels and arrival rates. This table shows that the stationary distribution converges for all arrival rates. As expected, in heavier traffic (when λ is close to $(1 + c)$) this convergence is

0.0000	0.0000	0.0000	0.0000	0.0001	0.0002	0.0002
0.0000	0.0000	0.0000	0.0000	0.0001	0.0004	0.0001
0.0000	0.0000	0.0001	0.0003	0.0011	0.0007	0.0000
0.0001	0.0003	0.0010	0.0033	0.0021	0.0003	0.0000
0.0003	0.0008	0.0022	0.0062	0.0011	0.0002	0.0000
0.0018	0.0051	0.0168	0.0106	0.0005	0.0001	0.0000
0.0166	0.0480	0.0312	0.0051	0.0003	0.0000	0.0000
0.0370	0.0876	0.0158	0.0024	0.0001	0.0000	0.0000
0.1920	0.1398	0.0085	0.0013	0.0001	0.0000	0.0000
0.2612	0.0901	0.0057	0.0008	0.0000	0.0000	0.0000

Table 8.2: Stationary distribution with $c = \pi/2$, $\lambda = 1.5$ and $N = 15$

slower; but the table shows that the convergence does happen. Therefore we conclude that the truncation method defined in Section 8.4 works: the stationary distribution converges in N. Interestingly, the ratio between the waiting times becomes c in heavier traffic. This implies that in that case the expected waiting time for an arriving customer is equal for both queues. This also confirms the results in [42], where it is shown that SED-routing is asymptotically optimal in the heavy traffic limit.

For a comparison, we have computed the stationary distribution of the SED-routing model when $c = \pi$. These results are displayed in Table 8.4. Again, the results show that the stationary distribution converges, but slower when the system is heavily loaded. We have chosen the arrival rates such that the load of the system corresponds to the loads used in Table 8.3. When taking a closer look at for example $\lambda = 4\frac{\pi+1}{\pi+2}$, the system with $c = \pi$ has the same load as when $\lambda = 2$ and $c = \pi/2$. The total expected queue length does not differ by much (4.176 and 4.3669), but the distribution over the two queues is very different. The average queue length of queue 1 when $c = \pi/2$ is almost twice the size of the length of the same queue when $c = \pi$, although they operate with the same rate. The average queue length of queue 2 is longer (by a factor 1.5) if $c = \pi$, but the corresponding server has twice the rate of the other model.

This leads to the conclusion that the waiting time in a system with 2 servers that have very different service rates is lower than when the two servers operate with (almost) the same speed. This effect is due to the stability of the system: arriving customers are served immediately by the faster server when the system is empty. This occurs frequently, especially when the system is not heavily loaded. In heavy traffic this effect diminishes; when the workload is the same, the average waiting time for a customer is independent of the service speed ratio c.

	$N = 50$	$N = 100$	$N = 200$
$\lambda = 1.5$	$EL_1 = 0.6605$ $EL_2 = 1.1965$	$EL_1 = 0.6605$ $EL_2 = 1.1965$	$EL_1 = 0.6605$ $EL_2 = 1.1965$
$\lambda = 2$	$EL_1 = 1.5586$ $EL_2 = 2.6172$	$EL_1 = 1.5587$ $EL_2 = 2.6173$	$EL_1 = 1.5587$ $EL_2 = 2.6173$
$\lambda = 2.25$	$EL_1 = 2.9583$ $EL_2 = 4.7959$	$EL_1 = 2.9829$ $EL_2 = 4.8342$	$EL_1 = 2.9830$ $EL_2 = 4.8343$
$\lambda = 2.5$	$EL_1 = 7.6543$ $EL_2 = 12.1688$	$EL_1 = 11.5219$ $EL_2 = 18.2226$	$EL_1 = 13.7665$ $EL_2 = 21.7494$

Table 8.3: Results for $c = \pi/2$

	$N = 50$	$N = 100$	$N = 200$
$\lambda = 3\frac{\pi+1}{\pi+2}$	$EL_1 = 0.3004$ $EL_2 = 1.6256$	$EL_1 = 0.3004$ $EL_2 = 1.6256$	$EL_1 = 0.3004$ $EL_2 = 1.6256$
$\lambda = 4\frac{\pi+1}{\pi+2}$	$EL_1 = 0.8803$ $EL_2 = 3.4866$	$EL_1 = 0.8804$ $EL_2 = 3.4866$	$EL_1 = 0.8804$ $EL_2 = 3.4867$
$\lambda = 4.5\frac{\pi+1}{\pi+2}$	$EL_1 = 1.7714$ $EL_2 = 6.2606$	$EL_1 = 1.7873$ $EL_2 = 6.3104$	$EL_1 = 1.7873$ $EL_2 = 6.3105$
$\lambda = 5\frac{\pi+1}{\pi+2}$	$EL_1 = 4.7005$ $EL_2 = 15.3979$	$EL_1 = 7.1122$ $EL_2 = 22.9800$	$EL_1 = 8.5176$ $EL_2 = 27.3956$

Table 8.4: Results for $c = \pi$

Bibliography

Self-references

[S1] Katehakis, M. N. and Smit, L. C. "On computing optimal (Q,r) replenishment policies under quantity discounts". *Annals of Operations Research*, 200(1):279–298, 2012.

[S2] Katehakis, M. N. and Smit, L. C. "A successive lumping procedure for a class of Markov chains". *Probability in the Engineering and Informational Sciences*, 26(4):483–508, 2012.

[S3] Katehakis, M. N., Smit, L. C., and Spieksma, F. M. "DES and RES processes and their explicit solutions". *Probability in the Engineering and Informational Sciences*, 29(02):191–217, 2015.

[S4] Katehakis, M. N., Smit, L. C., and Spieksma, F. M. "A comparative analysis of the successive lumping and the lattice path counting algorithms". *Journal of Applied Probability*, 53(1):106–120, 2016.

[S5] Katehakis, M. N., Smit, L. C., and Spieksma, F. M. "On the solution to a system of equations arising in stochastic processes". *Submitted to Mathematics of Operations Research*, 2016.

[S6] Ertiningsih, D., Katehakis, M. N., Smit, L. C., and Spieksma, F. M. "Level product form QSF processes and an analysis of queues with Coxian interarrival distribution". *Accepted at Naval Research Logistics*, 2015.

[S7] Ertiningsih, D., Smit, L. C., and Spieksma, F. M. "Extensions to successive lumping". In preparation, 2016.

[S8] Katehakis, M. N., Smit, L. C., and Spieksma, F. M. "Shortest expected delay routing with arbitrary service rates". In preparation, 2016.

References

[9] Adan, I. J. B. F., Kapodistria, S., and van Leeuwaarden, J. S. H. "Erlang arrivals joining the shorter queue". *Queueing Systems*, 74(2-3):273–302, 2013.

[10] Adan, I. J. B. F., van de Waarsenburg, W. A., and Wessels, J. "Analyzing $E_k|E_r|c$ queues". *European Journal of Operational Research*, 92(1):112–124, 1996.

[11] Adan, I. J. B. F., Boxma, O. J., Kapodistria, S., and Kulkarni, V. G. "The shorter queue polling model". *Annals of Operations Research*, pages 1–34, 2013.

[12] Adan, I. J. B. F., Economou, A., and Kapodistria, S. "Synchronized reneging in queueing systems with vacations". *Queueing Systems*, 62(1-2):1–33, 2009.

[13] Adan, I. J. B. F. and Wessels, J. "Shortest expected delay routing for Erlang servers". *Queueing systems*, 23(1-4):77–105, 1996.

[14] Adan, I. J. B. F., Wessels, J., and Zijm, W. H. M. "A compensation approach for two-dimensional Markov processes". *Advances in Applied Probability*, pages 783–817, 1993.

[15] Akritas, A. G., Akritas, E. K., and Malaschonok, G. I. "Various proofs of Sylvester's (determinant) identity". *Mathematics and Computers in Simulation*, 42(4):585–593, 1996.

[16] Alexanderian, A. "On continuous dependence of roots of polynomials on coefficients". *Retrieved from http://users.ices.utexas.edu/∼alen/articles/polyroots.pdf*, 2013.

[17] Ammar, G. S. "Classical foundations of algorithms for solving positive definite Toeplitz equations". *Calcolo*, 33(1-2):99–113, 1996.

[18] Ammar, G. S. and Gragg, W. B. "Superfast solution of real positive definite Toeplitz systems". *SIAM Journal on Matrix Analysis and Applications*, 9(1):61–76, 1988.

[19] Anderson, W. J. *Continuous-time Markov Chains: An Applications-oriented Approach*, volume 7. Springer-Verlag, New York, NY, 1991.

[20] Artalejo, J. R., Economou, A., and Lopez-Herrero, M. J. "The maximum number of infected individuals in SIS epidemic models: Computational techniques and quasi-stationary distributions". *Journal of Computational and Applied Mathematics*, 233(10):2563–2574, 2010.

[21] Artalejo, J. R. and Gómez-Corral, A. "Retrial queueing systems". *Mathematical and Computer Modelling*, 30(3-4):xiii–xv, 1999.

[22] Asmussen, S. *Applied Probability and Queues*, volume 2. Springer-Verlag, New York, NY, 2003.

[23] Asmussen, S., Nerman, O., and Olsson, M. "Fitting phase-type distributions via the EM algorithm". *Scandinavian Journal of Statistics*, pages 419–441, 1996.

[24] Baer, N., Boucherie, R. J., and van Ommeren, J. K. "The PH/PH/1 multi-threshold queue". In *Analytical and Stochastic Modeling Techniques and Applications*, pages 95–109. Springer, 2014.

[25] Barrett, W. W. and Feinsilver, P. J. "Inverses of banded matrices". *Linear Algebra and its Applications*, 41:111–130, 1981.

[26] Ben-Israel, A. and Greville, T. N. E. *Generalized inverses: theory and applications*, volume 15. Springer Science & Business Media, 2003.

[27] Bhat, U. N. *An Introduction to Queueing Theory: Modeling and Analysis in Applications*. Springer Science & Business Media, 2008.

[28] Bhulai, S. "On the value function of the $M|Cox(r)|1$ queue". *Journal of Applied Probability*, 43(2):363–376, 2006.

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

[30] Bini, D. A. and Meini, B. "On the solution of a nonlinear matrix equation arising in queueing problems". *SIAM Journal on Matrix Analysis and Applications*, 17(4):906–926, 1996.

[31] Bini, D. A., Meini, B., Steffé, S., and Van Houdt, B. "Structured markov chains solver: software tools". In *Proceeding from the 2006 workshop on Tools for solving structured Markov chains*, page 14, Pisa, Italy, 2006. ACM.

[32] Böhm, W., Krinik, A., and Mohanty, S. G. "The combinatorics of birth-death processes and applications to queues". *Queueing Systems*, 26(3-4):255–267, 1997.

[33] Bright, L. and Taylor, P. G. "Calculating the equilibrium distribution in level dependent quasi-birth-and-death processes". *Stochastic Models*, 11(3):497–525, 1995.

[34] Brown, M., Peköz, E. A., and Ross, S. M. "Some results for skip-free random walk". *Probability in the Engineering and Informational Sciences*, 24(04):491–507, 2010.

[35] Coombs-Reyes, J. D. "Customer allocation policies in a two server network: stability and exact asymptotics". *PhD Thesis*, 2003.

[36] Derman, C., Lieberman, G. J., and Ross, S. M. "On the optimal assignment of servers and a repairman". *Journal of Applied Probability*, pages 577–581, 1980.

[37] Eisenblätter, A., Wessäly, R., Martin, A., Fügenschuh, A., Wegel, O., Koch, T., Achterberg, T., and Koster, A. "Modelling feasible network configurations for UMTS". In *Telecommunications Network Design and Management*. Springer, United States, 2003.

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

[39] Feinberg, B. N. and Chui, S. S. "A method to calculate steady state distributions of large Markov chains by aggregating states". *Operations Research*, 35(2):282–290, 1987.

[40] Flajolet, P. and Guillemin, F. "The formal theory of birth-and-death processes, lattice path combinatorics and continued fractions". *Advances in Applied Probability*, 32(3):750–778, 2000.

[41] Flatto, L. and McKean, H. P. "Two queues in parallel". *Communications on Pure and Applied Mathematics*, 30(2):255–263, 1977.

[42] Foschini, G. J. "On heavy traffic diffusion analysis and dynamic routing in packet switched networks". *Computer Performance*, pages 499–513, 1977.

[43] Frostig, E. "Jointly optimal allocation of a repairman and optimal control of service rate for machine repairman problem". *European Journal of Operational Research*, 116(2):274–280, 1999.

[44] Gaver, D. P., Jacobs, P. A., and Latouche, G. "Finite birth-and-death models in randomly changing environments". *Advances in Applied Probability*, pages 715–731, 1984.

[45] Gillent, F. and Latouche, G. "Semi-explicit solutions for M/PH/1-like queuing systems". *European Journal of Operational Research*, 13(2):151–160, 1983.

[46] Grassmann, W. K. "Real eigenvalues of certain tridiagonal matrix polynomials, with queueing applications". *Linear Algebra and its Applications*, 342(1):93–106, 2002.

[47] Gross, D. and Miller, D. R. "The randomization technique as a modeling tool and solution procedure for transient Markov processes". *Operations Research*, 32(2):343–361, 1984.

[48] Hager, W. W. "Updating the inverse of a matrix". *SIAM review*, 31(2):221–239, 1989.

[49] He, Q. M. and Neuts, M. F. "Markov chains with marked transitions". *Stochastic Processes and their Applications*, 74(1):37–52, 1998.

[50] Heinig, G. and Rost, K. *Algebraic Methods for Toeplitz-like Matrices and Operators*. Springer, Basel, Switserland, 1984.

[51] Hooghiemstra, G. and Koole, G. "On the convergence of the power series algorithm". *Performance Evaluation*, 42(1):21–39, 2000.

[52] Hordijk, A. and Spieksma, F. "Constrained admission control to a queueing system". *Advances in Applied Probability*, 21(2):409–431, 1989.

[53] Hordijk, A. and Spieksma, F. "On ergodicity and recurrence properties of a Markov chain with an application to an open Jackson network". *Advances in Applied Probability*, 24(2):343–376, 1992.

[54] Ikebe, Y. "On inverses of Hessenberg matrices". *Linear Algebra and its Applications*, 24:93–97, 1979.

[55] Janssen, H. K. "On the nonequilibrium phase transition in reaction-diffusion systems with an absorbing stationary state". *Zeitschrift für Physik B Condensed Matter*, 42(2):151–154, 1981.

[56] Kapodistria, S. "The M/M/1 queue with synchronized abandonments". *Queueing Systems*, 68(1):79–109, 2011.

[57] Karlin, S. *A First Course in Stochastic Processes*. Academic Press INC., London, 1966.

[58] Katehakis, M. N. and Derman, C. "Optimal repair allocation in a series system". *Mathematics of Operations Research*, 9(4):615–623, 1984.

[59] Katehakis, M. N. and Veinott Jr., A. F. "The multi-armed bandit problem: decomposition and computation". *Mathematics of Operations Research*, 12(2):262–268, 1987.

[60] Katehakis, M. N. and Melolidakis, C. "On the optimal maintenance of systems and control of arrivals in queues". *Stochastic Analysis and Applications*, 13(2):137–164, 1995.

[61] Katehakis, M. N. and Derman, C. "On the maintenance of systems composed of highly reliable components". *Management Science*, 35(5):551–560, 1989.

[62] Katehakis, M. N. and Melolidakis, C. "Dynamic repair allocation for a K out of N system maintained by distinguishable repairmen". *Probability in the Engineering and Informational Sciences*, 2:51–62, 1 1988.

[63] Kemeny, J. G. and Snell, J. L. *Finite Markov Chains*. D. van Nostrand Company, inc., Princeton, N.J., 1960.

[64] Kharoufeh, J. P. *Level-Dependent Quasi-Birth-and-Death Processes*. John Wiley & Sons, Inc., Hoboken, NJ, 2011.

[65] Kılıç, E. and Stanica, P. "The inverse of banded matrices". *Journal of Computational and Applied Mathematics*, 237(1):126–135, 2013.

[66] Kim, D. S. and Smith, R. L. "An exact aggregation algorithm for a special class of Markov chains". *Technical Report*, 1989.

[67] Kim, D. S. and Smith, R. L. "An exact aggregation/disaggregation algorithm for mandatory set decomposable Markov chains". In *Numerical Solution of Markov Chains*, pages 89–104. 1990.

[68] Koole, G. and Spieksma, F. "On deviation matrices for birth–death processes". *Probability in the Engineering and Informational Sciences*, 15(2):239–258, 2001.

[69] Latouche, G. and Ramaswami, V. "A logarithmic reduction algorithm for quasi-birth-death processes". *Journal of Applied Probability*, pages 650–674, 1993.

[70] Latouche, G. and Ramaswami, V. *Introduction to Matrix Analytic Methods in Stochastic Modeling*, volume 5. ASA-SIAM Series on Statistics and Applied Probability. SIAM, Philadelphia, PA., 1999.

[71] Li, H. B., Huang, T. Z., Liu, X. P., and Li, H. "On the inverses of general tridiagonal matrices". *Linear Algebra and its Applications*, 433(5):965–983, 2010.

[72] Liu, D. and Zhao, Y. Q. "Determination of explicit solutions for a general class of Markov processes". In *Matrix-Analytic Methods in Stochastic Models*, pages 343–358. Dekker, Basel, 1996.

[73] Mallik, R. K. "The inverse of a tridiagonal matrix". *Linear Algebra and its Applications*, 325(1):109–139, 2001.

[74] Martinsson, P. G., Rokhlin, V., and Tygert, M. "A fast algorithm for the inversion of general Toeplitz matrices". *Computers & Mathematics with Applications*, 50(5):741–752, 2005.

[75] Meurant, G. "A review on the inverse of symmetric tridiagonal and block tridiagonal matrices". *SIAM Journal on Matrix Analysis and Applications*, 13(3):707–728, 1992.

[76] Meyer, C. D. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

[77] Miranker, W. L. and Pan, V. Y. "Methods of aggregation". *Linear Algebra and its Applications*, 29:231–257, 1980.

[78] Mohanty, S. G. *Lattice Path Counting and Applications*. Academic Press, New York, NY, 1979.

[79] Mohanty, S. G. and Panny, W. "A discrete-time analogue of the M/M/1 queue and the transient solution: A geometric approach". *Sankhyā: The Indian Journal of Statistics, Series A*, pages 364–370, 1990.

[80] Neuts, M. F. *Matrix-Geometric Solutions in Stochastic Models: an Algorithmic Approach*. Courier Dover Publications, 1981.

[81] Neuts, M. F. "The burstiness of point processes". *Stochastic Models*, 9(3):445–466, 1993.

[82] Pérez, J. and Van Houdt, B. "Quasi-birth-and-death processes with restricted transitions and its applications". *Performance Evaluation*, 68(2):126–141, 2011.

[83] Perros, H. G. *Queueing Networks with Blocking*. Oxford University Press, New York, NY., 1994.

[84] Ramaswami, V. "A stable recursion for the steady state vector in Markov chains of M/G/1 type". *Stochastic Models*, 4:183–188, 1988.

[85] Ramaswami, V. and Latouche, G. "A general class of Markov processes with explicit matrix-geometric solutions". *Operations-Research-Spektrum*, 8(4):209–218, 1986.

[86] Ramaswami, V. and Lucantoni, D. M. "Algorithms for the multi-server queue with phase type service". *Stochastic Models*, 1(3):393–417, 1985.

[87] Righter, R. "Optimal policies for scheduling repairs and allocating heterogeneous servers". *Journal of Applied Probability*, 33(2):536–547, 1996.

[88] Riska, A. and Smirni, E. "M/G/1-type Markov processes: A tutorial". In *Performance Evaluation of Complex Systems: Techniques and Tools*, volume 2459, pages 36–63. Springer, 2002.

[89] Ross, S. M. *Applied Probability Models with Optimization Applications*. Holden-Day, Inc., SF, USA, 1970.

[90] Ross, S. M. *Stochastic Processes*. John Wiley and Sons, NY., 1996.

[91] Ross, S. M. *Introduction to Stochastic Dynamic Programming*. Academic press, Inc., NY,USA, 1983.

[92] Schweitzer, P. J., Puterman, M. L., and Kindle, W. L. "Iterative aggregation-disaggregation procedures for discounted semi-Markov reward processes". *Operations Research*, 33(3):589–605, 1985.

[93] Selen, J., Adan, I. J. B. F., Kapodistria, S., and van Leeuwaarden, J. S. H. "Steady-state analysis of shortest expected delay routing". *arXiv:1509.03535v2*, 2015.

[94] Seneta, E. "Computing the stationary distribution for infinite Markov chains". *Linear Algebra and its Applications*, 34:259–267, 1980.

[95] Seneta, E. *Non-negative Matrices and Markov Chains*. Springer-Verlag, New York, NY, 2^{nd} edition, 1980.

[96] Sonin, I. M. "Optimal stopping of Markov chains and three abstract optimization problems". *An International Journal of Probability and Stochastic Processes*, 83(4-6):405–414, 2011.

[97] Spieksma, F. M. and Tweedie, R. L. "Strengthening ergodicity to geometric ergodicity for Markov chains". *Stochastic Models*, 10(1):45–74, 1994.

[98] Spitzer, F. *Principles of Random Walk*, volume 34. Springer-Verlag, New York, NY, 2001.

[99] Sylvester, J. J. "XXXVII. on the relation between the minor determinants of linearly equivalent quadratic functions". *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1(4):295–305, 1851.

[100] Szpankowski, W. "Stability conditions for multidimensional queueing systems with computer applications". *Operations Research*, 36(6):944–957, 1988.

[101] Tong, H., Faloutsos, C., and Pan, J. Y. "Fast random walk with restart and its applications". In *ICDM '06*, pages 613–622. IEEE, 2006.

[102] Tweedie, R. L. "The calculation of limit probabilities for denumerable Markov processes from infinitesimal properties". *Journal of Applied Probability*, pages 84–99, 1973.

[103] Tweedie, R. L. "Criteria for ergodicity, exponential ergodicity and strong ergodicity of Markov processes". *Journal of Applied Probability*, 18(1):122–130, 1981.

[104] Ulukus, M. Y., Güllü, R., and Örmeci, L. "Admission and termination control of a two class loss system". *Stochastic Models*, 27(1):2–25, 2011.

[105] Ungureanu, V., Melamed, B., Katehakis, M. N., and Bradford, P. G. "Deferred assignment scheduling in cluster-based servers". *Cluster Computing*, 9(2):57–65, 2006.

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

[107] van Leeuwaarden, J. S. H., Squillante, M. S., and Winands, E. M. M. "Quasi-birth-and-death processes, lattice path counting, and hypergeometric functions". *Journal of Applied Probability*, 46(2):507–520, 2009.

[108] van Leeuwaarden, J. S. H. and Winands, E. M. M. "Quasi-birth-and-death processes with an explicit rate matrix". *Stochastic Models*, 22(1):77–98, 2006.

[109] Varga, R. S. *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1963.

[110] Varga, R. S. "On recurring theorems on diagonal dominance". *Linear Algebra and its Applications*, 13(1):1–9, 1976.

[111] Veinott Jr., A. F. "The optimal inventory policy for batch ordering". *Operations Research*, 13(3):424–432, 1965.

[112] Vere-Jones, D. "Ergodic properties of nonnegative matrices I". *Pacific Journal of Mathematics*, 22(2):361–386, 1967.

[113] Vlasiou, M., Zhang, J., and Zwart, B. "Insensitivity of proportional fairness in critically loaded bandwidth sharing networks". *arXiv preprint arXiv:1411.4841*, 2014.

[114] Williams, V. V. "Multiplying matrices faster than Coppersmith-Winograd". In *Proceedings of the 44th symposium on Theory of Computing*, pages 887–898, New York, NY, 2012. ACM.

[115] Woodbury, M. "Inverting modified matrices". *Memorandum Report, Statistical Research Group*, 42, 1950.

[116] Yap, V. B. "Similar states in continuous-time Markov chains". *Journal of Applied Probability*, 46(2):497–506, 2009.

[117] Zhang, Q., Mi, N., Riska, A., and Smirni, E. "Performance-guided load (un)balancing under autocorrelated flows". *IEEE Transactions on Parallel and Distributed Systems*, 19(5):652–665, 2008.

[118] Zhao, Y. Q. and Grassmann, W. K. "Queueing analysis of a jockeying model". *Operations Research*, 43(3):520–529, 1995.

[119] Zheng, Y. S. and Zipkin, P. "A queueing model to analyze the value of centralized inventory information". *Operations Research*, 38(2):296–307, 1990.

Samenvatting

Dit proefschrift behandelt netwerken met veel toestanden. De processen die we bekijken, hebben een bepaalde dynamiek op deze netwerken en hebben met name toepassingen in de wachttijdtheorie. Ook zijn er veel toepassingen die hun oorsprong kennen in de informatica en voorraadmanagement.

Een manier om een dergelijk systeem te beschrijven is een Markovketen. In het bijzonder modelleert een Markovketen een proces dat zich door een aantal toestanden beweegt en stapsgewijs overgangen vertoont van de ene naar een andere of naar dezelfde toestand. Als het systeem zich in een bepaalde toestand bevindt, hangt het toekomstige gedrag van dit systeem alleen maar af van de huidige toestand en niet van de manier waarop deze toestand tot stand is gekomen. In elke toestand verblijft het systeem een bepaalde stochastische tijd en na die periode springt het volgens een bepaalde kansverdeling naar een andere toestand.

We zijn met name geïnteresseerd in het exact bepalen van de stationaire verdeling van grootschalige Markovketens. De stationaire verdeling van een Markovketen bevat de kansen dat het systeem zich in een bepaalde toestand bevindt op een willekeurig moment in tijd. Anders gezegd, deze verdeling geeft aan welk percentage van de tijd het systeem zich in elke toestand bevindt. In het algemeen is het niet eenvoudig om de stationaire verdeling op een efficiënte manier exact te berekenen als het aantal toestanden van het netwerk groot is, mogelijk zelfs oneindig. Wanneer deze stationaire verdeling bekend is, kunnen verschillende eigenschappen van de beschouwde toepassing bepaald worden. Zo kan in een wachttijdmodel de gemiddelde wachttijd berekend worden en in een voorraadmodel de gemiddelde tijd dat er geen producten op voorraad zijn.

In Hoofdstuk 2 introduceren we de *herhaaldelijk-samenvoegings* (successive lumping) methode. Hiervoor is het noodzakelijk een partitie van de toestanden te maken in verschillende verzamelingen. Wanneer één van deze verzamelingen een zogenaamde ingangstoestand heeft, is de herhaaldelijk-samenvoegings methode een manier om de stationaire verdeling alleen voor toestanden in deze verzameling te berekenen. Een verzameling S heeft een ingangstoestand wanneer het volgende geldt: om een toestand in S te bereiken wanneer het systeem zich in een toestand buiten deze verzameling S bevindt, moet het systeem zich tussendoor enige tijd in de ingangstoestand bevinden. Dus in andere woorden: overgangskansen van buiten S naar een toestand in S gaan altijd via de ingangstoestand.
Nadat de stationaire verdeling is bepaald van de toestanden in een verzameling, voegen we deze samen tot een enkele toestand en herhalen we de procedure. In dit hoofdstuk geven we

de exacte formules om de stationaire verdeling te berekenen volgens deze methode en we bewijzen de correctheid hiervan.

Een bekende klasse van oplossingsmethodes om de stationaire verdeling van een Markovketen te bepalen is de klasse van de matrix-analytische methodes. Hiervoor moeten de overgangskansen van de Markovketen aan een specifieke structuur voldoen. Het idee van deze methodes is om de stationaire verdeling van een verzameling toestanden uit te drukken in die van een andere. Hierbij wordt gebruik gemaakt van een zogenaamde 'rate matrix', aangegeven met de letter R. In het algemeen is het lastig om deze matrix exact te bepalen. In Hoofdstuk 3 laten we zien dat herhaaldelijk-samenvoegings ook binnen deze aanpak toegepast kan worden om efficiënt R te berekenen.

Naast de specifieke overgangsstructuur wordt er binnen de matrix-analytische methodes over het algemeen van uit gegaan dat de transitieovergangen hetzelfde zijn per set: er is sprake van homogene transities. Dit is geen eis voor de herhaaldelijk-samenvoegings methode. Verder geven we een bovengrens voor de stationaire kansen op een deelverzameling van de toestanden, wanneer slechts een deel van de keten wordt beschouwd. Ook geven we enkele voorbeelden van specifieke toepassingen die binnen deze klasse vallen.

Er bestaan een aantal andere oplossingsmethodes om de bovengenoemde rate matrix R exact te bepalen. Deze methodes maken bijvoorbeeld gebruik van extra structuur van de transities. Eén van deze procedures telt het aantal mogelijke paden van een bepaalde toestand naar een andere toestand. Het is dan echter wel noodzakelijk dat de toestanden van de Markovketen als een raamwerk weergegeven kunnen worden en dat transities alleen naar buren mogelijk zijn. Daarnaast is het niet toegestaan om stappen naar het oosten in dit raamwerk te nemen, behalve op één van de randen. In Hoofdstuk 4 geven we aan hoe de klasse van problemen waarop deze methode toepasbaar is, verschilt van de klasse van problemen waarvoor de herhaaldelijk-samenvoegings methode gebruikt kan worden. Het blijkt dat de laatstgenoemde methode breder toepasbaar is, met name omdat er dan ook transities mogelijk zijn die toestanden overslaan. Ook kunnen de overgangskansen en verblijftijd per toestand verschillen. Beide methodes zitten echter wel (impliciet) vast aan de aanwezigheid van ingangstoestanden.

Een onderdeel van de herhaaldelijk-samenvoegings methode is het inverteren van een aantal grote matrices. Dit zijn operaties die in het algemeen veel tijd in beslag nemen, omdat voor een exact antwoord een groot aantal bewerkingen moet worden gedaan. Bij de herhaaldelijk-samenvoegings methode komt een bepaalde structuur van de niet-nul elementen in de matrix vaak naar voren. In Hoofdstuk 5 beschrijven we een algoritme dat deze structuur gebruikt om op een efficiënte manier een exacte inverse te berekenen. Naast het feit dat deze methode snel werkt, is het resultaat exact en onafhankelijk van de grootte van de matrix. Verder beschrijven we in dit hoofdstuk bepaalde eigenschappen van de eigenwaardes van deze matrix. Deze eigenwaarden hebben een negatief reëel deel, en we vermoeden zelfs dat het complexe deel nul is. Het blijkt dat deze eigenwaarden direct af te leiden te zijn uit die van een andere, gemakkelijker te analyseren matrix.

Zoals gezegd, om gebruik te maken van de herhaaldelijk-samenvoegings methode identificeren we een partitie van de toestanden in verzamelingen, zodanig dat de overgangskansen tussen verzamelingen voldoen aan een bepaalde structuur: er zijn 'ingangstoestanden' aanwezig. In Hoofdstuk 6 gaan we dieper in op een bepaald soort processen met een specifieke structuur van de overgangskansen tussen toestanden binnen een verzameling. We laten zien hoe je gebruik kan maken van 'uitgangstoestanden' en hoe deze equivalent zijn met de aanwezigheid van een ingangstoestand voor een iets gemodificeerde partitie. We voeren een numerieke analyse uit en leiden met behulp van de herhaaldelijk-samenvoegings methode af dat de stationaire verdeling van de toestanden aan een productvorm voldoet. Dit houdt in dat we de stationaire kans van een toestand kunnen uitdrukken als het product van een bepaalde factor en de stationaire kans van een andere toestand. Bovendien laten we zien hoe deze factor nauwkeurig geschat kan worden.

De herhaaldelijk-samenvoegings methode eist dat elke verzameling in het bezit is van een ingangstoestand. In Hoofdstuk 7 bekijken we een aantal manieren om deels onder deze voorwaarde uit te komen. Zo bekijken we systemen waar we zelf ingangstoestanden aan kunnen toevoegen. Ook bekijken we een andere, specifieke manier om ingangstoestanden te creëren door de transities vanuit een specifieke toestand s te verdelen in verschillende verzamelingen. Bij elk van deze verzamelingen maken een we een nieuwe Markovketen. Het oorspronkelijke model kan vanuit deze losse ketens terug worden geconstrueerd. Deze methode hebben we 'thinning' genoemd en wordt ook in dit hoofdstuk beschreven.

Het bepalen van een partitie die de herhaaldelijk-samenvoegings methode toestaat is in het algemeen een moeilijk probleem. Er zijn veel modellen waar deze partitie direct zichtbaar is, maar er zijn ook veel toepassingen waar deze helemaal niet voor de hand liggend is. In Hoofdstuk 8 bekijken we zo'n toepassing, geïnspireerd door een model uit de informatica. In dit model komen taken aan bij een processor met 2 verschillende wachtrijen behorende bij verschillende servers. De aankomende taak sluit aan in de rij waarin de verwachte verblijftijd op dat moment minimaal is. Bij dit model werken de servers met onderling verschillende snelheden, wat de analyse van dit model gecompliceerd maakt. Door gebruik te maken van de herhaaldelijk-samenvoegings methode kunnen we een aantal conclusies met betrekking tot de vorm van de stationaire verdeling trekken en hieruit de gemiddelde rijlengtes afleiden.

Acknowledgements

This thesis is the result of four years of research done as a PhD student, both in Leiden and in New Jersey, USA. My work has been funded both by Leiden University and the Rutgers University. A result from my master thesis work has been published as [S1], and is not included in this thesis. Chapter 2 and 3 of this thesis are used in a different form in my PhD thesis handed in at Rutgers University in April 2014. Chapter 6 and 7 are partially joint work with Dwi Ertiningsih (among others) and will also be used in her doctoral dissertation. Chapter 4, 5 and 8 are not all published yet, and partly contain ongoing work.

During the last four years I have felt the support of many people. It feels just to mention some of them, although I know that opinions differ on whether it is appropriate to do so in the acknowledgements of a thesis. Therefore, the person that I am most grateful to, is clearly missing in the lines below.

First of all, I am grateful to the members of the committee for a careful review of this thesis. Michael, I cannot thank you enough for convincing me to pursue a PhD degree. Although even when we were living in the same timezone our working hours were almost completely distinct, you still found a lot of time to supervise me and to do research.

The Operations Research group in Leiden is not very big, and I hope it will expand again in the upcoming years. Dwi, Herman and Jan-Pieter, it has been great to have you as colleagues and roommates in 217. Compared to Dwi and Herman, my PhD life has been easy. Dwi, I have a lot of respect for you, leaving everything behind to pursue a PhD degree. Herman, commuting every day and raising kids besides doing research does not leave a lot of time for other things. But for some reason you never showed any signs of stress. Jan-Pieter, you brought new life to the OR group and gave me valuable advice about this thesis. Unfortunately we (I) did not have the time to start new projects, but who knows maybe later?

The establishment of this thesis has been a bureaucratic nightmare. My sincere gratitude goes out to Luz Kosar and the other supporting staff in Rutgers for helping me addressing all these issues and to Martin Lübke for giving me such a flexible contract in Leiden.

Finally, a word to Geesje. You think that you were not of any help during the last four years. That is not true. You put things in perspective and supported me when I was absent for all those months. I can not wait for the 28th of May!

Curriculum Vitae

Laurens Smit werd geboren op 20 maart 1986 te Leiden. In 2004 heeft hij zijn VWO-diploma gehaald aan het Stedelijk Gymnasium Leiden. Daarna is Laurens direct begonnen met de studie Wiskunde aan de Universiteit Leiden. In 2008 heeft hij zijn bachelordiploma behaald. Aansluitend is hij begonnen met de master Applied Mathematics, eveneens aan de Universiteit Leiden. Als onderdeel van deze master heeft Laurens aan Rutgers University in Newark gewerkt aan zijn afstudeerscriptie onder begeleiding van prof. Michael Katehakis en dr. Flora Spieksma vanuit de Universiteit Leiden. Deze scriptie is getiteld 'Efficient Computations for a Class of Markov Chains and Related Applications' en werd in 2011 met succes verdedigd.

Vervolgens is Laurens begonnen aan een promotieonderzoek aan de Universiteit Leiden, eveneens onder begeleiding van dr. Spieksma. Gedurende de looptijd van dit promotieonderzoek, heeft hij een aantal semesters doorgebracht aan Rutgers University, om daar onder begeleiding van prof. Katehakis in Mei 2014 het PhD-programma in Management Science af te ronden met de thesis getiteld 'On Successive Lumping of Large Scale Systems'.

Het onderzoek van Laurens werd in 2013 beloond met de 'Dean's Award for Ph.D. Student Research' van Rutgers University. Eveneens in 2013 won hij met [S3] de 'INFORMS New Jersey Chapter Contest', een jaarlijkse competitie voor Operation Research Students in New Jersey.