Universiteit Leiden

Bachelor's Thesis

Successive Lumping and Lattice Path Counting

Author:
Simon Vroegop

Supervisor:
Dr. F.M. Spieksma

A comparison of two methods for obtaining the stationary probabilities of Quasi-Birth-Death Processes

July 23, 2014

1 Introduction

In this thesis, we will compare two different methods for computing the stationary probabilities of Quasi-Birth-Death processes. The first method is Lattice Path Counting, as described in [1] and [2], the second Successive Lumping, which is explained in detail in [3].
We begin by explaining what a Quasi-Birth-Death process is and why it's so hard to calculate its steady state probabilities. After that, a short description follows of both methods. We will briefly explain how they work and in which cases they can be used. These descriptions are meant as a short introduction to the workings of these methods. A more in depth explanation and more formal proofs can be found in the respective articles in which the methods are introduced.
After that, both methods will be applied to a number of examples to show that the Successive Lumping method is applicable in more situations than Lattice Path Counting.

Readers are expected to have a basic understanding of both discrete- and continuous-time Markov chains.

2 Quasi-Birth-Death Processes

The Quasi-Birth-Death (QBD) process is a generalization of a simple Birth-Death (BD) process. To better understand how a QBD process works, let's first examine a standard BD process. A Birth-Death Process is a continuous-time Markov process with $N \leq \infty$ states and transition rates $q(i,j) = 0$, if $j \notin \{i-1, i+1\}$.
In other words, when in a certain state i, the system can only make a transition to the two adjacent states $i+1$, signifying a *birth*, and $i-1$, signifying a *death*. The transition rates $q(i, i+1)$ and $q(i, i-1)$ are usually denoted by λ_i and μ_i, respectively, so the intensity matrix looks like

$$Q = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & 0 & \cdots \\ \mu_1 & -(\lambda_1+\mu_1) & \lambda_1 & 0 & 0 & \cdots \\ 0 & \mu_2 & -(\lambda_2+\mu_2) & \lambda_2 & 0 & \cdots \\ 0 & 0 & \mu_3 & -(\lambda_3+\mu_3) & \lambda_3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

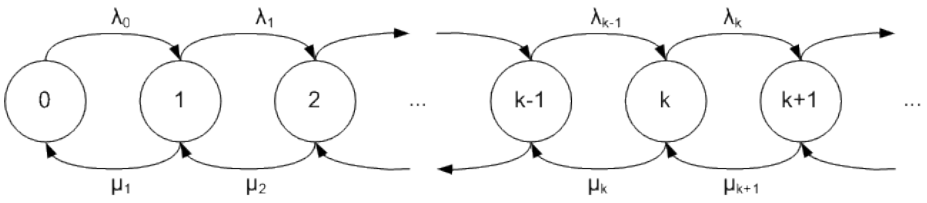

Figure 1: A graphical representation of a Birth-Death process

Due to the simple structure of this kind of process, it's relatively easy to compute the stationary probabilities $\pi_0, \pi_1, ...\pi_N$. By definition of the stationary

distribution, for two adjacent states we should have $\lambda_i \pi_i = \mu_{i+1} \pi_{i+1}$. Also, as we know, the sum of all probabilities in a discrete distribution must be equal to one. Thus to compute the stationary distribution - provided that it exists - we solve the following system of recursive equations.

$$\pi_{i+1} = \frac{\lambda_i}{\mu_{i+1}} \pi_i, \quad \text{for} \quad 0 \leq i \leq N-1 \tag{1}$$

$$\sum_{i=0}^{N} \pi_i = 1 \tag{2}$$

From these equations we can also see that the stationary probabilities can only exist if the model satisfies

$$\sum_{k=0}^{\infty} \left[\prod_{i=0}^{k} \frac{\lambda_i}{\mu_{i+1}} \right] < \infty$$

A Quasi-Birth-Death process is similar to a BD process, only it has a two-dimensional state space. Hence for given constants N and ℓ ($N, \ell \leq \infty$), the states can be written as tuples (n, i), where $n = 0, 1, 2, ...N$ and $i = 0, 1, 2, ..., \ell$. The states from the BD process are now replaced by *levels* of states, within which every transition is permitted. These levels are defined as $L_n = \{(n, 0), (n, 1), \ldots, (n, \ell)\}$, for $0 \leq n \leq \infty$ Transitions between two levels however are only permitted if these levels are adjacent. This means that all QBD processes satisfy $q((n, i)|(n', i')) = 0$ if $|n - n'| > 1$.

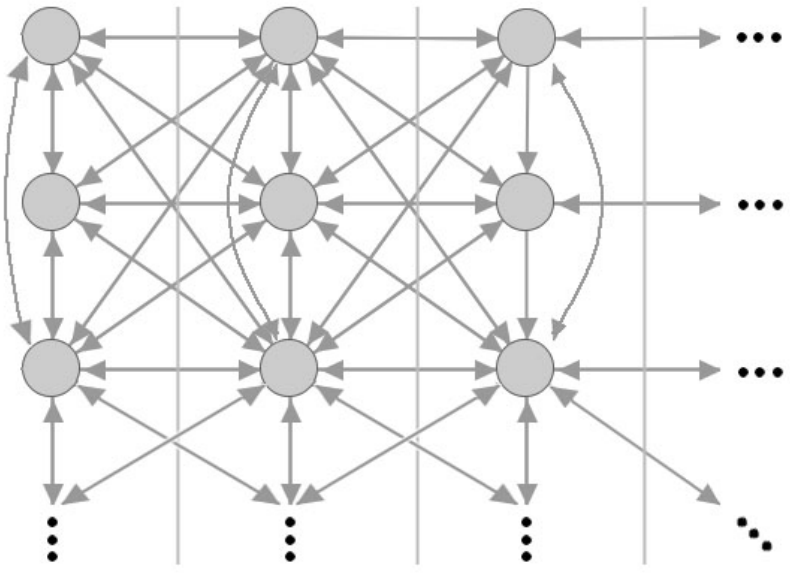

Figure 2: An example of a Quasi-Birth-Death Process

In this paper, we will only examine homogeneous QBD processes, which means that the transition rates are level independent (with the exception of the lowest and - if it exists - highest level). We will also restrict our focus to irreducible and positive recurrent processes; otherwise the stationary distribution does not exist. In these cases, the intensity matrix can be written as

$$Q = \begin{pmatrix} W^0 & U & 0 & 0 & 0 & \cdots \\ D & W & U & 0 & 0 & \cdots \\ 0 & D & W & U & 0 & \cdots \\ 0 & 0 & D & W & U & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

In this matrix, D, W and U are $\ell \times \ell$ submatrices for transitions respectively down one level, within the same level and up one level.

In these processes the stationary distribution is the vector $\boldsymbol{\pi}$ that satisfies $\boldsymbol{\pi} \cdot \boldsymbol{Q} = \mathbf{0}$ and $\boldsymbol{\pi} \cdot \mathbf{1} = 1$. Calculating this distribution is, as one might expect, much more complicated than with a BD process.
However, if we partition $\boldsymbol{\pi}$ into subvectors $\boldsymbol{\pi}_i$, where the i'th subvector corresponds with the states of L_i, we can find a relation between the stationary probability vectors of the level sets similarly to the one we found for BD processes:

$$\boldsymbol{\pi}_{k+1} = \boldsymbol{\pi}_k \boldsymbol{R}, \qquad \text{or alternately} \qquad \boldsymbol{\pi}_k = \boldsymbol{\pi}_0 \boldsymbol{R}^k$$

for a certain matrix $\mathbf{R}$, which can be found as follows.
From the definition of the stationary distribution $\boldsymbol{\pi} \cdot \boldsymbol{Q} = \mathbf{0}$ and the assumption $\boldsymbol{\pi}_{k+1} = \boldsymbol{\pi}_k \boldsymbol{R}$, we find

$$\boldsymbol{\pi}_i U + \boldsymbol{\pi}_{i+1} W + \boldsymbol{\pi}_{i+2} D = 0,$$

thus

$$\boldsymbol{\pi}_i U + \boldsymbol{\pi}_i \boldsymbol{R} W + \boldsymbol{\pi}_i \boldsymbol{R}^2 D = 0$$

and

$$\boldsymbol{\pi}_i (U + \boldsymbol{R} W + \boldsymbol{R}^2 D) = 0.$$

From this identity, it can be proven that the matrix $\mathbf{R}$ is the solution of the equation

$$U + \boldsymbol{R} W + \boldsymbol{R}^2 D = \mathbf{0}.$$

Now, the problem that arises is how to explicitly determine the matrix $\mathbf{R}$. Iterative procedures exist, but they only approximate it. Therefore we are interested in methods of calculating $\mathbf{R}$ exactly, without approximations. In the next chapter, we will look at two of these methods.

3 The methods

3.1 Lattice Path Counting

The first method we examine is called *Lattice Path Counting* (LPC).
First, we need the following definition from [1].

Definition 1: Excursion
For an arbitrary level n, an excursion *is defined as the time elapsing from the moment the QBD process leaves an initial state in level n until the time of the first return of the process to level n. The excursion should always leave level n.*

The elements R_{jk} of the matrix R can then be shown (cf. [1] and [5]) to represent the expected time spent in state $(n+1, k)$ during an excursion with initial state (n, j), expressed in the expected time spent in state (n, j).
Note that this property is only applicable if the length of an excursion does not depend on the level n.
In [2], we study a related matrix G, which denotes the probabilities of an excursion, starting from L_{n+1}, visiting level L_n in finite time. The elements G_{jk} represent te probability of the process, when starting in state $(n+1, j)$, to reach level n in the state (n, k). In [2] the relation between R and G is shown to be

$$R = U(-(W + UG))^{-1}$$

Now, if we have an explicit expression for G, we can use that to compute the matrix R.
Due to the immense number of possible excursions, it is usually extremely difficult or even impossible to calculate the elements of G. Therefore, this method can only be used for processes which conform to certain restrictions.

We use the notation $\langle a, b\rangle$ to denote a one-step transition from a state (n, j) to the state $(n + a, j + b)$. The probability of such a step is denoted by $\phi\langle a, b\rangle$. In order to use the LPC method, we can only consider QBD-processes in which the only transitions from non-boundary states $(n > 0, j < \ell)$ are $\langle -1, 0\rangle$, $\langle -1, 1\rangle$, $\langle 0, 1\rangle$, $\langle 1, 1\rangle$ and $\langle 1, 0\rangle$. In the states $(n, \ell), n > 0$ the only transitions that are allowed are $\langle -1, 0\rangle$ and $\langle 1, 0\rangle$. This means that in all states (n, j) with $n > 0$, the process can not move in a downward direction.

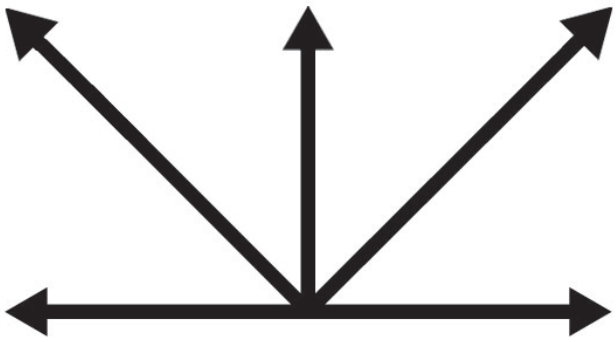

Figure 3: The accepted transitions for using the LPC method

With these restrictions, [2] now gives us a way of calculating the elements of G by determining all possible paths from $(n+1, j)$ to (n, k). Because movement in a downward direction is not allowed in these models, we know that $G_{jk} = 0$ if $k < j$. Also, the transition rates are constant throughout all the states. Therefore, instead of G_{jk}, we can use G_h, where $h = k - j$.
So h is the total amount of steps taken in the upward direction $\langle 0, 1\rangle$ and in the diagonal directions $\langle -1, 1\rangle$ and $\langle 1, 1\rangle$. If we assume the path includes s $\langle -1, 1\rangle$ steps and u $\langle 1, 1\rangle$ steps, then it will include $t = h - s - u$ $\langle 0, 1\rangle$ steps.

We will first only consider the path in horizontal direction. For this purpose, the diagonal steps $\langle -1, 1\rangle$ and $\langle 1, 1\rangle$ will be considered horizontal steps. We denote the total amount of horizontal steps $\langle 1, 0\rangle$ with m. Then the amount of $\langle -1, 0\rangle$ steps is equal to $m + 1$. We also know that $m \geq \max\{u, s - 1\}$.
The process only reaches level L_n in the last step. That means that the first $2m$ horizontal steps are equivalent to a *Bernoulli Excursion*, a random walk on the x-axis, starting at $x = 0$ that contains m steps in both directions, but never crosses the y-axis. From [6] we know that for given m, the number of possible Bernoulli excursions containing $2m$ steps is given by the mth Catalan number:

$$\frac{1}{m+1}\binom{2m}{m}.$$

If we revert back to the original counting problem, including vertical steps, we now have the tools to calculate the total number of paths from $(n+1, j)$ to $(n, j+h)$.
By $L_h(s, u, m)$ we denote the number of paths from $(n+1, j)$ to $(n, j+h)$ that contain m steps to the right, u of which are diagonal $\langle 1, 1\rangle$ steps, and $m + 1$ steps to the left, s of which are the diagonal steps $\langle -1, 1\rangle$. This means that the number of $\langle 1, 0\rangle$ an $\langle -1, 0\rangle$ steps are equal to $m - u$ and $m + 1 - s$ respectively. The paths contain h steps to the top, including the diagonal steps, which brings the number of vertical $\langle 0, 1\rangle$ steps to $t = h - s - u$. The total number of steps in these paths is $v = (m - u) + u + (m + 1 - s) + s + t = 2m + t + 1$.

Lemma 1
The value of $L_h(s, u, m)$ can be calculated as follows.

$$L_h(s, u, m) = \frac{1}{m+1}\binom{2m}{m}\binom{m+1}{s}\binom{m}{u}\binom{2m+t}{t}$$

Proof
When only considering the horizontal direction, the paths form a Bernoulli excursion with $2m$ steps, followed by one step to the left. The number of possible Bernoulli excursions with $2m$ steps is

$$\frac{1}{m+1}\binom{2m}{m}.$$

When considering the problem in all directions, we know that out of the $m + 1$ steps to the left, s are diagonal $\langle -1, 1\rangle$ steps and out of the m steps to the right, u are $\langle 1, 1\rangle$ steps. Therefore the number of possible Bernoulli excursions must be multiplied by

$$\binom{m+1}{s}\binom{m}{u}.$$

Now we only need to consider the vertical $\langle 0,1\rangle$ steps. The total number of steps is $2m+t+1$ and each of these steps can be a $\langle 0,1\rangle$ step, except for the last step, which must go from L_{n+1} to L_n. Thus, out of $2m+t$ steps, t are vertical. So we multiply by

$$\binom{2m+t}{t}$$

to arrive at the expression for $L_h(s,u,m)$.
For the probability of these paths $P_h(s,u,m)$ we can find

$$P_h(s,u,m) = \phi\langle -1,1\rangle^s \phi\langle 0,1\rangle^{h-s-u} \phi\langle 1,1\rangle^u \phi\langle 1,0\rangle^{m-u} \phi\langle -1,0\rangle^{m+1-s}$$

Now that we know an expression for $L_h(s,u,m)$, we can finally calculate the elements of G.

Theorem 1
For $h = 0,1,\ldots,$ we have

$$G_h = \sum_{s=0}^{h} \sum_{u=0}^{h-s} \sum_{m=\max\{u,s-1\}}^{\infty} L_h(s,u,m) P_h(s,u,m)$$

Using this, we can now calculate the rate matrix with which we can obtain the stationary distribution.

3.2 Successive Lumping

Another method to obtain the stationary probabilities of QBD processes is called *Successive Lumping* (SL) and is described in [3].
The main idea of this method is to partition the state space $\mathcal{X}$ into a sequence of level sets $\mathcal{D} := \{D_0, D_1, \ldots, D_M\}$ with $M \leq \infty$. For the sake of notational convenience, we then relabel the states of every of these sets D_m to $\{(m,1),(m,2),\ldots,(m,\ell_m)\}$ for certain $\ell_m \leq \infty$. We then compute the stationary distribution of the QBD process when limited to the state space D_0. We then use these stationary probabilities to determine the stationary distribution of the process when limited to the state space $D_0 \cup D_1$ and so forth.
Of course, this isn't always possible, because there are generally too many possible transitions between two adjacent subsets of the partition. That's where the concept of *entrance states* comes in.

Definition 2: Entrance State
Given a Markov chain on the state space $\mathcal{X}$ with intensity matrix Q, a subset $\mathcal{Y} \subset \mathcal{X}$ has an entrance state $\varepsilon \in \mathcal{Y}$ *if and only if*

$$q(m,j|m',j') = 0 \text{ for all } (m',j') \notin \mathcal{Y} \text{ and } (m,j) \neq \varepsilon.$$

To put it in words: a subset $\mathcal{Y}$ of the state space has an entrance state ε if and only if ε is the only state in $\mathcal{Y}$ that can be directly reached from states in $\mathcal{X}\backslash\mathcal{Y}$. All other states in the subset can only be reached from within the subset.
The existence of specific entrance states in a QBD process greatly reduces the number of possible transitions between certain adjacent levels. Therefore, we can use the method described above, if the process conforms to the following definition, described in [3].

Definition 3: Successive Lumpability
A Markov chain $X(t)$ is called successively lumpable *with respect to a partition $\mathcal{D} = \{D_0, \ldots, D_M\}$ if and only if the sets $\underset{\sim}{D}_m := \bigcup_{i=0}^{m} D_i$ have an entrance state for every $m = 0, 1, \ldots M$. The entrance state for a set $\underset{\sim}{D}_m$ is denoted as $(m, \varepsilon_m(\mathcal{D}))$.*

When we now look at a QBD process that is successively lumpable, we can construct the Markov chain $Z_0(t)$, with intensity matrix U_0, which represents the process restricted to the state space D_0.
Because the process $X(t)$ is positive recurrent, we know that every time the process leaves D_0, the probability of it returning to that set is equal to 1. We also know that D_0 contains an entrance state $(0, \varepsilon_0(\mathcal{D}))$, so every transition from outside of D_0 to D_0 will be a transition to $(0, \varepsilon_0(\mathcal{D}))$. Therefore, if we restrict our attention to the state space D_0, every transition that leaves the space, can be seen as a transition to $(0, \varepsilon_0(\mathcal{D})$. Then, the elements of the intensity matrix U_{D_0} are

$$u_{D_0}(0,j|0,i) = \begin{cases} q(0,\varepsilon_0(\mathcal{D})|0,i) + \sum\limits_{(k',j')\notin D_0} q(k',j'|0,i), & \text{if } j = \varepsilon_0(\mathcal{D}), \\ q(0,j|0,i) & \text{otherwise.} \end{cases}$$

Because this state space is much smaller than $\mathcal{X}$, it's relatively simple to determine the stationary probabilities of this process using conventional methods. We shall denote these by $v_{D_0}(0,i)$.
We now 'lump' the states of D_0 together to form an artificial state which we call $(1,0)$.
We define $\Delta_m := (m,0) \cup D_m$ for $m = 1,2,\dots M$ and we examine the process on the state space $\Delta_1 \cup D_2 \cup \cdots \cup D_M$. Similarly to the previous Markov chain we constructed, $Z_0(t)$, we now construct the chain $Z_1(t)$ on the state space Δ_1. Just like with Z_0, we can simply calculate the steady state probabilities $v_{\Delta_1}(1,i)$ of this process. The way to interpret the elements $v_{\Delta_1}(1,i)$ for $i = 1,2,\dots,\ell_1$ is as follows. If the state space of the process $X(t)$ would have consisted only of D_0 and D_1, then $v_{\Delta_1}(1,i)$ would be the stationary probability of the state $(1,i)$. Also, $v_{\Delta_1}(1,0)$ would then equal the sum of the stationary probabilities of all the states $(0,i) \in D_0$. We could find the steady state probability of an individual state $(0,j) \in D_0$ using $\pi(0,j) = v_{D_0}(0,j)v_{\Delta_1}(1,0)$.
However, if the state space does not consists of only D_0 and D_1, we need additional lumping steps to compute the stationary distribution. Therefore we lump the states $(1,i), i = 0,1,2,\dots,\ell_1$ together to form the artificial state $(2,0)$, form a Markov process on the state space Δ_2 and calculate the values of $v_{\Delta_2}(2,i)$ for $i = 0,1,2,\dots,\ell_2$.
We continue like this, constructing a sequence of Markov chains $Z_m(t)$, for $m \leq M$ with state space Δ_m and intensity matrix U_{Δ_m} whose elements are given by

$$u_{\Delta_m}(m,j|m,i) = \begin{cases} q(m,j|m,i) + \sum\limits_{(k',j') \notin \Delta_m} q(k',j'|m,i), & \text{if } j = \varepsilon_m(\mathcal{D}), \\ q(m,j|m,i) & \text{otherwise.} \end{cases}$$

For every chain $Z_m(t)$ we can calculate the stationary probabilities $v_{\Delta_m}(m,i)$. After we have found the steady state distributions of all the processes Z_m, with $m = 0,1,\dots,M$, we can use the following theorem to calculate the stationary distribution of the whole process $X(t)$.

Theorem 2
If $X_0(t)$ is successively lumpable with $|\mathcal{X}| < \infty$, then

$$\pi(m,j) = v_{\Delta_m}(m,j) \prod_{k=m+1}^{M} v_{\Delta_k}(k,0), \text{ for all } (m,j) \in \mathcal{X}$$

A formal proof for this theorem can be found in [3], but an intuitive way to show this is the following: $v_{\Delta_M}(M,0)$ is the sum of the stationary probabilities of all the states in $\mathcal{X} \backslash D_M$. Then $v_{\Delta_M}(M,0)v_{\Delta_{M-1}}(M-1,0)$ is the sum of the stationary probabilities of all the states in $\mathcal{X} \backslash (D_M \cup D_{M-1})$. And so forth.

Note that this theorem only holds if the state space is finite. In that case we can find an explicit solution for the steady state distribution. If the state space is infinite, we might not be able to find an explicit answer. However, by calculating $v_{\Delta_m}(m,0)$ for every m, we do have a lot of information about the proportions between the stationary distribution of D_{m-1} and that of D_m, comparable to finding the rate matrix using the LPC method.

4 Examples

4.1 Example 1: Two-Station Tandem Queue

The first model is the two-station tandem queue from [1]. In this model we have two service stations sharing a single server. Customers join the queue at station 1 according to a Poisson Process with rate λ_1. Upon completing service, they will join the queue at station 2, which additionally has its own stream of arrivals, with Poisson rate λ_2. There is also a third Poisson Process which yields a customer at both queues with arrival rate λ_3. Customers at station 1 and 2 are serviced with rate μ_1 and μ_2 respectively. The single server moves between stations, giving priority to customers at station 1, thus only offering service at station 2 when station 1 is empty.[1] Station 2 has a finite capacity of size ℓ. Customers arriving at station 2 immediately leave the system, if they find the queue at full capacity.
The states in this process are written as (n, j), where n and j denote the number of customers waiting at station 1 and 2 respectively.

The $(\ell \times \ell)$ submatrices of Q in this model would be:

$$U = \begin{pmatrix} \lambda_1 & \lambda_3 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_1 & \lambda_3 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_1 & \lambda_3 & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \ddots & 0 \\ 0 & 0 & \ldots & 0 & \lambda_1 & \lambda_3 \\ 0 & 0 & \cdots & 0 & 0 & \lambda_1 + \lambda_3 \end{pmatrix} \quad D = \begin{pmatrix} 0 & \mu_1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \mu_1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \mu_1 & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \ddots & 0 \\ 0 & 0 & \ldots & 0 & 0 & \mu_1 \\ 0 & 0 & \cdots & 0 & 0 & \mu_1 \end{pmatrix}$$

$$W^0 = \begin{pmatrix} \Delta_0 & \lambda_2 & 0 & 0 & \cdots & 0 \\ \mu_2 & \Delta_0 & \lambda_2 & 0 & \cdots & 0 \\ 0 & \mu_2 & \Delta_0 & \lambda_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ldots & \mu_2 & \Delta_0 & \lambda_2 \\ 0 & 0 & \cdots & 0 & \mu_2 & \Delta_0 \end{pmatrix} \quad W = \begin{pmatrix} \Delta & \lambda_2 & 0 & 0 & \cdots & 0 \\ 0 & \Delta & \lambda_2 & 0 & \cdots & 0 \\ 0 & 0 & \Delta & \lambda_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ldots & 0 & \Delta & \lambda_2 \\ 0 & 0 & \cdots & 0 & 0 & \Delta \end{pmatrix}$$

With the diagonal elements Δ and Δ_0 we denote the negative sum of all elements of the respective rows they are in. This ensures that the row sums of the transition matrix Q are all equal to 0.

In the figure below we've drawn a visual representation of all possible transitions. In this graph we can more clearly see that the possible transitions in non-boundary states are the diagonal steps $\langle -1, 1\rangle$ and $\langle 1, 1\rangle$, the vertical step $\langle 0, 1\rangle$, and the horizontal step $\langle 1, 0\rangle$. Also, in the states (n, ℓ) the only transitions are the horizontal steps $\langle -1, 0\rangle$ and $\langle 1, 0\rangle$. As we know from the previous chapter, these transitions conform to the restrictions needed to use the LPC method. Therefore in this case it is possible to use this method.

[1] Note that in order for the process to retain the Markov property, i.e. being memoryless, when servicing a customer at station 2, as soon as a customer arrives at station 1, the server must immediately abandon his service of that customer and move to station 1.

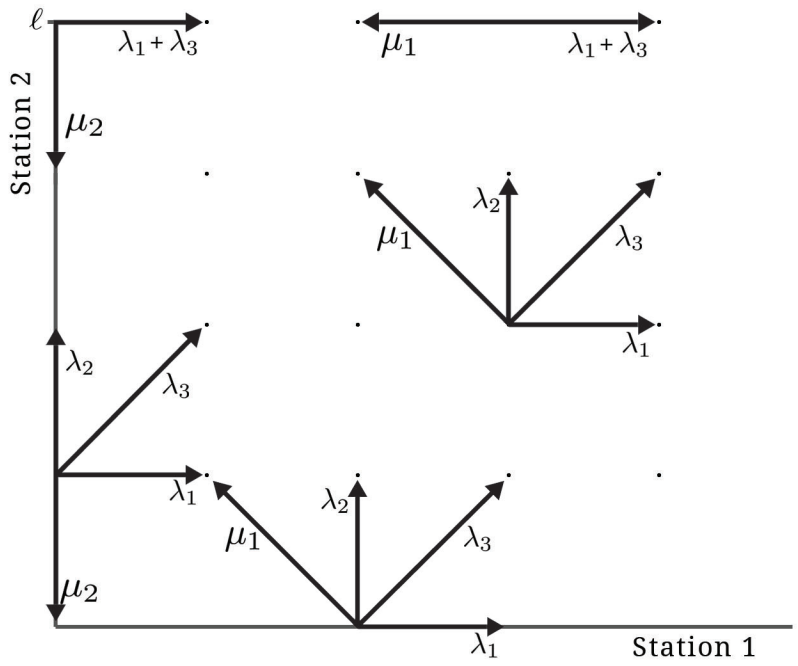

Figure 4: The transition graph for the two-station tandem queue model

We recall that the prerequisite for using Successive Lumping is the existence of entrance states for a certain partition $\mathcal{D}$.
We can show that this is true, if we use the following partition.

$$D_m := \{(0,m),(1,m),(2,m),\dots\} \text{ for } m = 0,1,\dots,\ell$$

These are the horizontal rows of states. Because the only downward transitions in this model are the $\langle 0,-1\rangle$ steps from the states where $n = 0$, the states $(0,j)$ are now the entrance states of the levels D_j. This can be seen more clearly in the figure below.
Furthermore, because the amount of lumping steps is finite, we can easily obtain an explicit expression for the stationary probability of any state, whereas by using LPC we have only found the matrix R and consequently determining an explicit solution will be much harder.

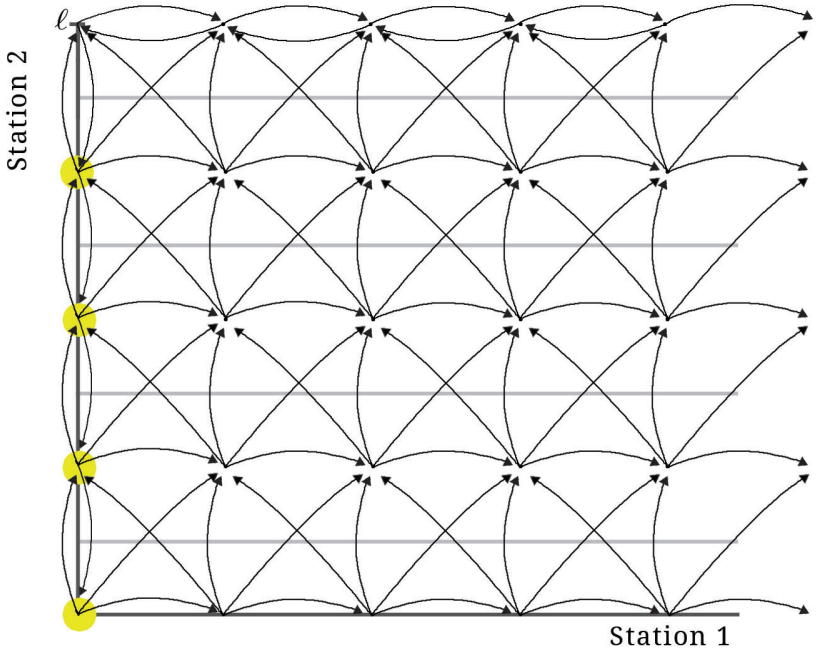

Figure 5: Level partitioning of the two-station tandem queue model so that every level has an entrance state

4.2 Longest Queue Priority

The longest queue priority model, as found in [2] and [4] consists of two stations. Each queue has a steady incoming stream of customers, arriving according to a Poisson process with rates λ_1 and λ_2 for station 1 and station 2 respectively. The system has one server, with service rate μ, who moves between the stations, only providing service for the station with the longest queue. If the queues are the same length, one is chosen at random with equal probabilities.

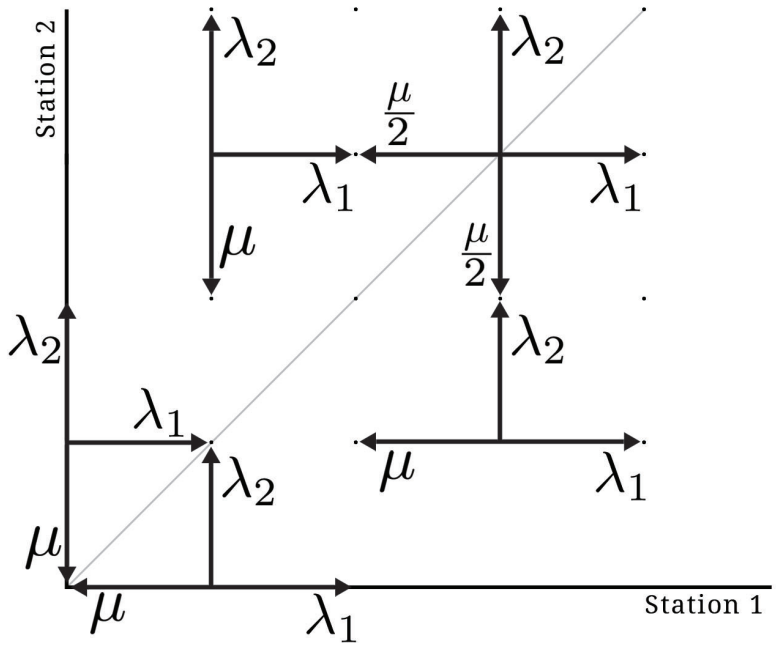

Figure 6: The transition graph for the longest queue priority model

At first glance, it's not possible to solve this model with the LPC method. After all, looking at Figure 6, we can see that from many states, there is a transition in the downward direction. However, with a small adjustment, we can transform the transition graph into one without downward transitions. Originally for a tuple (n, j), n denotes the length of the first queue and j the length of the second. Instead, j will now denote the length of the shortest queue and n the difference between the length of the two. This has two drawbacks. Firstly, this only works if $\lambda_1 = \lambda_2$. Secondly, we lose some information about how many people are waiting for each of the stations. We know the length of the longest and shortest queue, but have no way of knowing which is which.
However, as we can see in the figure below, we now have obtained a process where the only downward transitions are those starting in the states $(0, j)$. Therefore we can now use the LPC method to obtain the stationary probabilities.

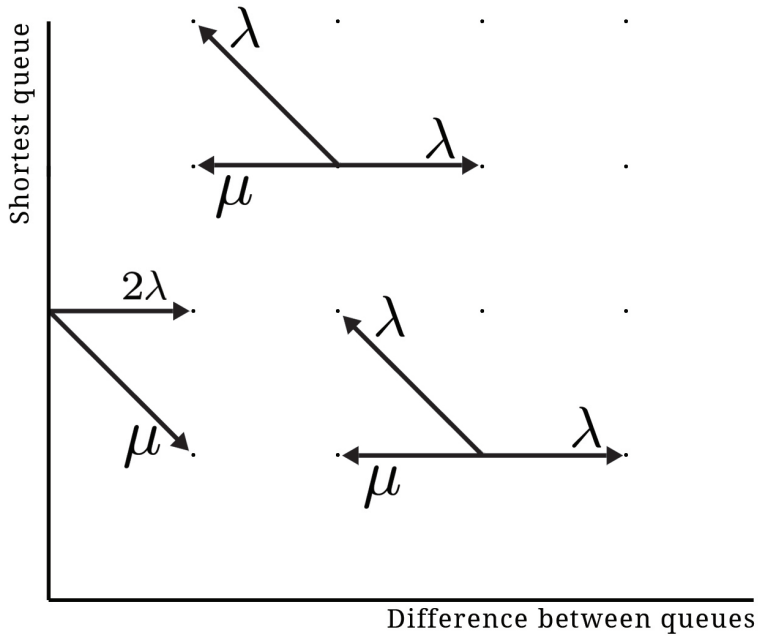

Figure 7: Transitions of the longest queue priority model after relabeling the axes

It also seems at first that the model is not lumpable. A partition in which each level has an entrance state is not readily apparent. However, in [4] such a partition is found and is shown to be

$$\begin{aligned} D_m := \quad & \{(m,m)\} \cup \{(m-1,m),(m-1,m+1),(m-1,m+2),\dots\} \\ \cup \quad & \{(m,m-1),(m+1,m-1),(m+2,m-1),\dots\} \end{aligned}$$

In the figure below, we see this partition visualized. We can see that the diagonal elements (m,m) are the entrance states of the sets $\underset{\sim}{D}_m$. So we can conclude that the model is successively lumpable. Additionally, unlike the LPC method, we can even use SL in this model if $\lambda_1 \neq \lambda_2$, and also we retain all information about which queue is which.

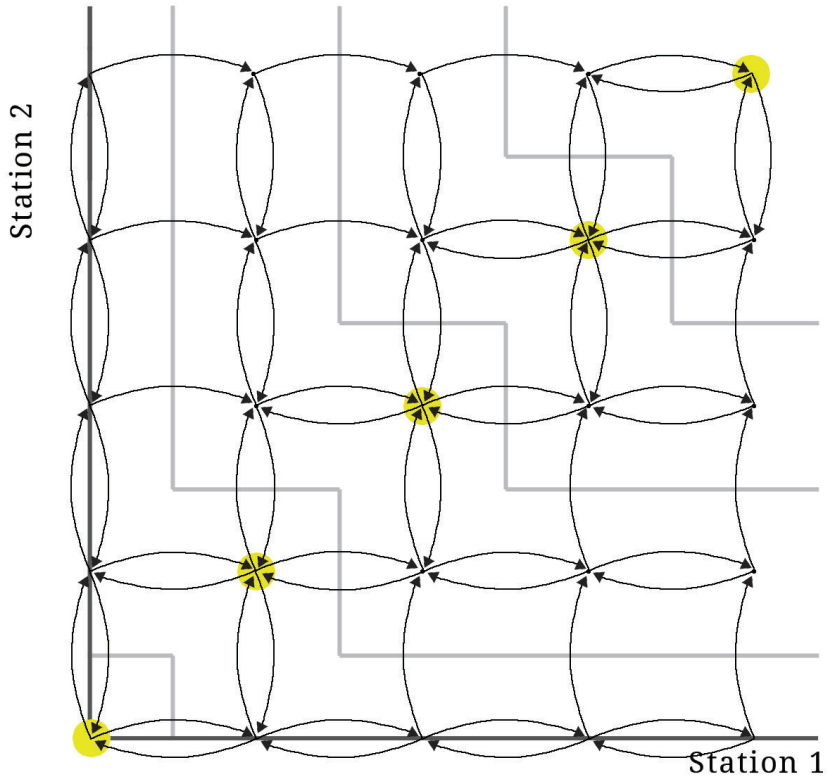

Figure 8: Level partitioning of the longest queue priority model with the entrance states marked yellow

4.3 Join the Shortest Queue

This last example is called the Join the Shortest Queue model. It contains two stations, each with one server, with service rates μ_1 and μ_2. There is one arrival stream, with rate λ. Upon arrival, a customer joins the queue that is shortest at that moment. If both queues are the same length, a queue will be picked at random.
This means that from states under the diagonal the possible transitions are $\langle 0, 1\rangle$, $\langle 0, -1\rangle$ and $\langle -1, 0\rangle$, from states above the diagonal $\langle 1, 0\rangle$, $\langle 0, -1\rangle$ and $\langle -1, 0\rangle$ are possible, and from the states on the diagonal we can take the steps $\langle 0, 1\rangle$, $\langle 0, -1\rangle$, $\langle 1, 0\rangle$ and $\langle -1, 0\rangle$. These transitions are drawn in the figure below.
We can see very clearly that the LPC method is not applicable when the model looks like this.

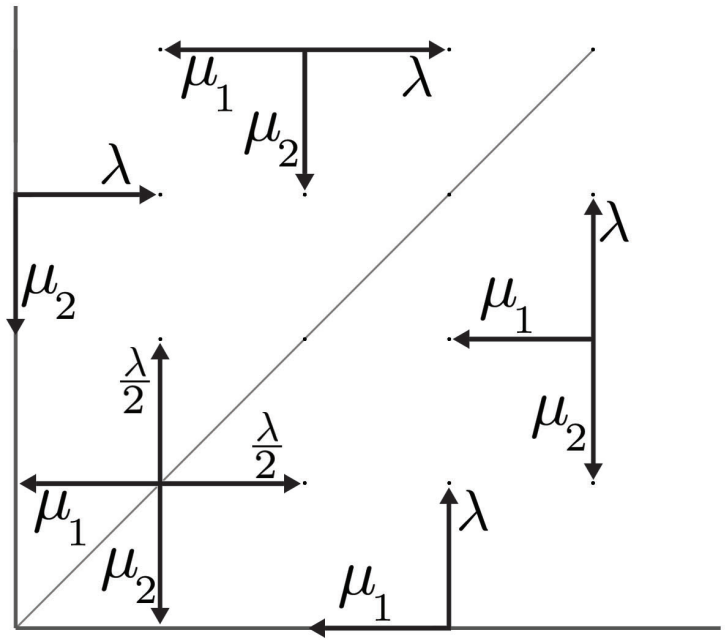

Figure 9: The transition graph for the join-the-shortest-queue model

This graph looks very similar to that of the longest queue priority model. It would seem that using the same adjustments as we did in that case would make it possible to use the LPC method, just as it did before.
So again, we denote by n the difference in length between the two queues and by j the length of the shortest queue. Then we draw the transition graph for the states (n, j).

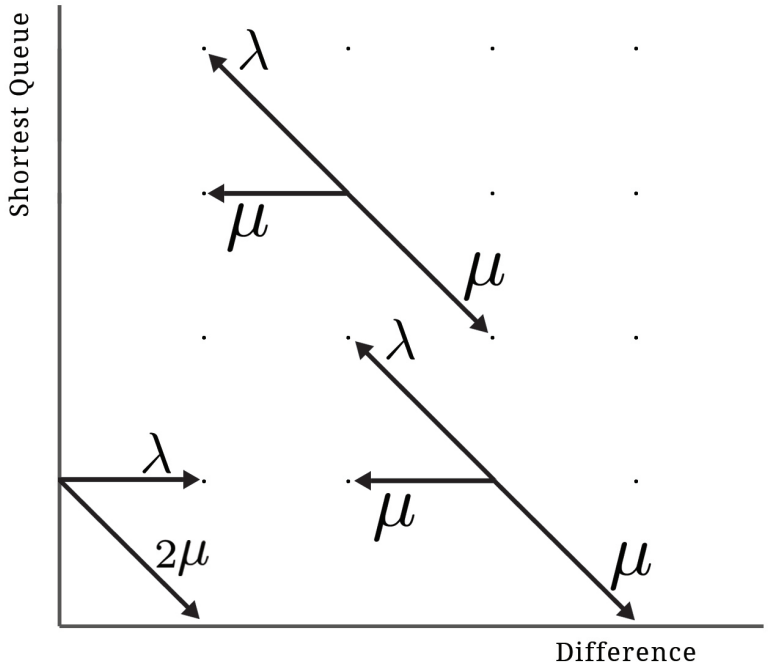

Figure 10: Transition graph of the join-the-shortest-queue model after relabeling the axes

We see that unlike in the case of the longest queue priority model, relabeling the axes does not enable us to use Lattice Path Counting on this model. There are still downward transitions in non-boundary states. In fact, there is no way to adjust this problem to allow the LPC method to be used.

Now we want to determine if this model is lumpable.
As noted above, the transition graph is very similar to that of the previous example, only mirrored. Therefore, a good guess for a partition containing entrance states at every level would be to flip the one from figure 8 around. We define the partition as:

$$\begin{aligned} D_m := \quad & \{(m,m)\} \cup \{(m+1,m),(m+1,m-1),(m+1,m-2),\dots(m+1,0)\} \\ \cup \quad & \{(m,m+1),(m-1,m+1),(m-2,m+1),\dots(0,m+1)\} \end{aligned}$$

If we define the partition like this, the transition graph looks as in the figure below

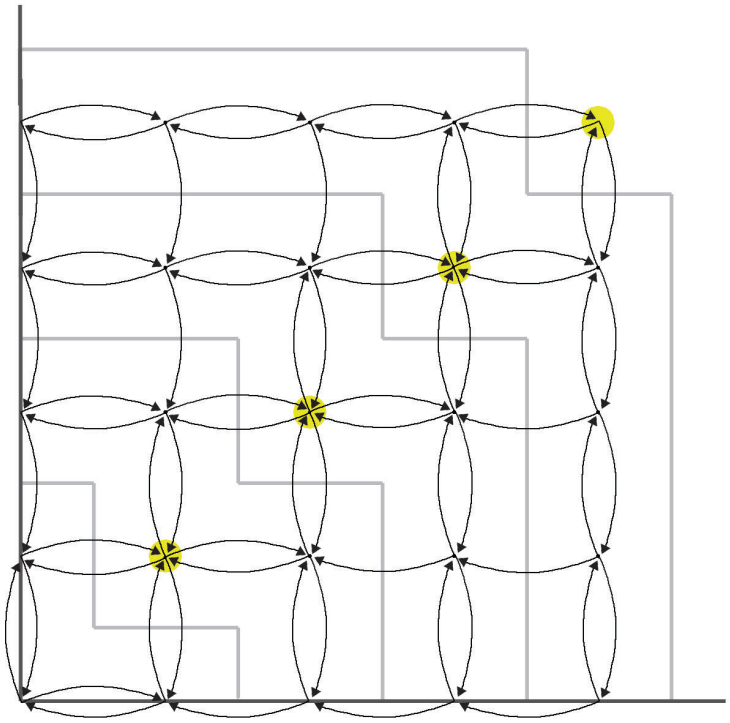

Figure 11: Level partitioning of the join-the-shortest-queue model so that every level has an entrance state (marked yellow)

As we can see, the diagonal states are entrance states of their respective levels. However, we do notice one difference from the previous examples. Namely, the states (m, m) aren't entrance states to the sets $\underset{\sim}{D}_m = \bigcup_{i=0}^{m} D_i$, but rather to the sets $\widetilde{D}_m = \bigcup_{i=m}^{\infty} D_i$.
This means that we can't use Successive Lumping to find an explicit solution for the stationary distribution. However, we can still find the relation between the stationary distributions of each level, which still gives us insight in the steady state probabilities of the process.

These last two models illustrate another advantage of using Successive Lumping rather than Lattice Path Counting: SL doesn't require all level sets to be the same size.
In the Longest Queue Priority-mode, the first level contained only one state, whereas all other levels had infinitely many. In the Join the Shortest Queue-model, all levels contained finitely many states, but no two level sets had the same amount.
This illustrates the versatility of the Successive Lumping method, as opposed to Lattice Path Counting, which requires every level to be the same size.

5 Conclusion

In the above examples we demonstrated that Successive Lumping can be used in more situations than Lattice Path Counting.
The first model proved to be no problem for either method. Both LPC and SL were viable options and the model required no further adjustments.
In the second model, as soon as we found a suitable partition of the states, we had no problems using SL. The LPC method however, could only be used if the transition rates of the incoming streams were the same. Using that method also required us to throw away information about which queue was which.
In the third example we found a model that could not be solved using LPC. Although we couldn't find an explicit solution using SL, the method could provide us with the proportions between the different partition levels, which gives us some information about the steady state probabilities.

We also demonstrated Successive Lumping's versatility, applying it to both finite and infinite level sets, and even when all levels had a different size.

In short, while LPC is a good method to determine the rate matrix in some QBD processes, it would seem that the Successive Lumping method is applicable in more situations and gives us more information about the stationary distributions.

References

[1] J.S.H. van Leeuwaarden & E.M.M. Winands, *Quasi-Birth-Death processes with an explicit rate matrix*. Stochastic Models vol. 22, 2006.

[2] J.S.H. van Leeuwaarden, M.S. Squillante & E.M.M. Winands, *Quasi-Birth-And-Death Processes, Lattice Path Counting, And Hypergeometric Functions*. Journal of Applied Probability vol. 46, 2009.

[3] M. N. Katehakis & L. C. Smit, *A Successive Lumping Procedure for a Class of Markov Chains*. Probability in the Engineering and Informational Sciences vol. 26, 2012

[4] M. N. Katehakis, Laurens C. Smit & F.M. Spieksma, *Successive Lumping versus Lattice Path Counting in Queueing*.

[5] G. Latouche & V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling, 1987*.

[6] L. Takács, *A Bernoulli Excursion and it Various Applications*. Advances in Applied Probability vol. 23, 1991