DEPARTEMENT TOEGEPASTE ECONOMISCHE WETENSCHAPPEN

ONDERZOEKSRAPPORT NR 9818

QUEUE LENGHTS AND WAITING TIMES IN THE TWO-CLASS TWO-SERVER QUEUE WITH NONPREEMPTIVE HETEROGENEOUS PRIORITY STRUCTURES

by

H. LEEMANS

G. DEDENE

Katholieke Universiteit Leuven

Naamsestraat 69, B-3000 Leuven

ONDERZOEKSRAPPORT NR 9818

QUEUE LENGHTS AND WAITING TIMES IN THE TWO-CLASS TWO-SERVER QUEUE WITH NONPREEMPTIVE HETEROGENEOUS PRIORITY STRUCTURES

by

H. LEEMANS
G. DEDENE

D/1998/2376/18

Queue Lengths and Waiting Times in the Two-Class Two-Server Queue with Nonpreemptive Heterogeneous Priority Structures

H. Leemans
G. Dedene

Abstract

Our aim is to analyze a multiserver queue with nonpreemptive heterogeneous priority structures, which arises in the performance evaluation of batch initiator settings in MVS. We use matrix-geometric methods and derive the stationary distribution of queue lengths and waiting times for the Markovian two-class two-server case.

1 Introduction

Priorities arise very naturally in many real life queueing applications. Also, in many of these applications, the system consists of more than one server. Customers typically belong to a specific class and observe the same priority structure on all servers. However, it also happens that the priority structure differs amongst the servers of the same system. Such models arise in batch job processing within the mainframe operating system MVS. Batch jobs are divided in classes based on their resource requirements (e.g. cpu seconds, memory requirements, ...) and they are executed in separate address spaces, called *initiators*. The number of initiators has to be defined by the performance manager; this definition includes a list of classes the initiators are allowed to execute. A simple initiator definition example, with only four initiators and four job classes, is shown below.

```
INITDEF   PARTNUM=4
INIT001   CLASS=AB, START, NAME=I1
INIT002   CLASS=AB, START, NAME=I2
INIT003   CLASS=CD, START, NAME=I3
INIT004   CLASS=DC, START, NAME=I4
```

The order in which the classes are listed imposes a priority structure on the classes. In the example above, class A has priority over class B on initiators I1 and I2; the priority structures on these initiators are *homogeneous*. Initiators I3 and I4 are defined to execute class C and class D jobs. However, their priority structures are *heterogeneous*: on I3, class C has priority over class D, whereas class D has priority over class C on I4. The priorities are *nonpreemptive*.

In practice, we typically find multiple job classes assigned to multiple initiators with a variety of priority structures. In this paper, we restrict ourselves to the simplified system consisting of two servers (S_1 and S_2) and two job classes (A and B) as illustrated in Figure 1. On server S_1, class A has nonpreemptive priority over class B; on server S_2, class B has nonpreemptive priority over class A. Both classes have Poisson arrivals with parameters λ_a and λ_b respectively. Service times are exponentially distributed with average $1/\mu_a$ and $1/\mu_b$ where μ_a may differ from μ_b. When both servers are idle, an arriving job is served by the server which offers the highest priority. Service discipline within each class is FCFS.

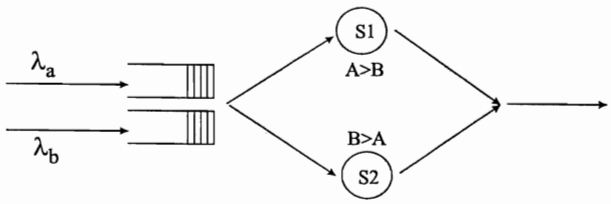

Figure 1: Two-class two-server priority queueing model with heterogeneous priority structures.

We analyze the queues of this model assuming that the system is stable. We therefore require that $\rho_a + \rho_b < 2$, which is the well-known stability condition for a two-class two-server queue without priorities. A proof of this condition may be given using an argument of drift. See Gail et al. [GHT88], as well as Leemans [Lee98] for details.

Multiserver priority queueing models have been analyzed before. The two-class Markovian multiserver queue with *homogeneous* priority structures

has been solved exactly by Gail, Hantler and Taylor [GHT88, GHT92] and Mitrani and King [MK81] using classical transform methods. Our model differs from those models in that our priority structures are *heterogeneous*. Two-class Markovian queues with *heterogeneous* priority structures have been studied by Fayolle and Iasnogorodski [FI79] (two coupled processors) and Fayolle, King and Mitrani [FKM82] (two-class M/M/c with mixed priorities) using the boundary value approach. Those models differ from our model in that their priorities are *preemptive* and therefore, the state space is two-dimensional. As we shall see, our two-class model with nonpreemptive priorities has a three-dimensional state space, for which the boundary value approach is no longer directly applicable. Moreover, these methods only consider the stationary distribution of queue lengths.

All of these models have later been analyzed with matrix-geometric methods, mainly to illustrate the power and elegance of this method (see Kao and Narayanan [KN91], Miller [Mil92] and Rao and Posner [RP86]). We also apply matrix-geometric methods to analyze the queues of our model and we derive the stationary distribution of queue lengths and waiting times. The remainder of this paper is organized as follows. In Section 2, we define the matrix-geometric model and we show how the joint stationary distribution of queue lengths is obtained. Using these results, we establish in Section 3 an algorithm to obtain the stationary distribution of waiting times. The algorithm is illustrated with with some numerical results.

2 Stationary Distribution of Queue Lengths

Let us denote the state of the system by the tuple (n_a, n_b, x, y), where n_a and n_b respectively represent the number of class A and class B jobs in the system (in the queue or in service). As such, n_a and n_b can take the integer values 0,1,2, ... The indices x and y refer to the class of job that is being served on S_1 and S_2 respectively. Consequently, their values may be A, B or 0; the latter indicates that the respective server is idle. It is necessary to include this information as a third dimension in the state description: the server that becomes idle determines the class that is served next and influences the path that is followed through the chain and therefore also the stationary distribution of queue lengths.

By ordering the states lexicographically, we find that the generator matrix

for this Markov process has the following structure:

$$Q = \begin{bmatrix} B_{00} & B_{01} & & & & \\ B_{10} & B_{11} & B_{12} & & & \\ & B_{21} & A_1 & A_0 & & \\ & & A_2 & A_1 & A_0 & \\ & & & A_2 & A_1 & \ddots \\ & & & & \ddots & \ddots \end{bmatrix}. \tag{1}$$

The entries of Q are infinite blocks, each corresponding with a group of states with the same value of n_a, called a *level*. Formally, the level $\ell(i)$ is defined as $\{(n_a, n_b, x, y) | n_a = i, 0 \leq n_b\}$. Within each level, we find groups of states with the same value of n_b, called the *major phase*. Formally, the major phase $\varpi(j)$ within the level $\ell(i)$ is defined as $\{(n_a, n_b, x, y) | n_a = i, n_b = j\}$. Finally, each major phase has a number of *minor phases*, representing all valid combinations of values for x and y. The minor phases for each major phase within each level are shown in Table 1. It is easy to see that the levels $\ell(0)$ and $\ell(1)$ are different from the levels $\ell(i)$, $i \geq 2$. The levels $\ell(0)$ and $\ell(1)$ are therefore called the *boundary* levels; the levels $\ell(i)$, $i \geq 2$, are the *homogeneous* levels.

level	$\varpi(0)$	$\varpi(1)$	$\varpi(j), j \geq 2$
$\ell(0)$	$(0,0)$	$(0,B),(B,0)$	(B,B)
$\ell(1)$	$(A,0),(0,A)$	$(A,B),(B,A)$	$(A,B),(B,A),(B,B)$
$\ell(i), i \geq 2$	(A,A)	$(A,A),(A,B),(B,A)$	$(A,A),(A,B),(B,A),(B,B)$

Table 1: Number of minor phases for each major phase within each level.

The matrix Q is the generator of a *Quasi-Birth-and-Death* (QBD) process. Transitions are only allowed between neighboring levels, or between adjacent major phases within the same level. The matrices B_{01}, B_{12} and A_0 describe the rates at which the process moves to a higher level; B_{10}, B_{21} and A_2 represent transition rates to a lower level; B_{00}, B_{11} and A_1 indicate the transitions within the level. The structure of the matrices A_0, A_1 and A_2 is shown below. Empty positions indicate that no transition is possible between the corresponding major phases. The block $\triangle^*$ denotes an appropriately sized diagonal matrix, the elements of which are such that the row sums of Q equal zero.

$$A_0 = \begin{bmatrix} L_{A0} & & & \\ & L_{A1} & & \\ & & L_A & \\ & & & \ddots \end{bmatrix}, \qquad A_1 = \begin{bmatrix} \triangle^* & L_{B0} & & & \\ M_{B0} & \triangle^* & L_{B1} & & \\ & M_{B1} & \triangle^* & L_B & \\ & & M_B & \triangle^* & \ddots \\ & & & \ddots & \ddots \end{bmatrix},$$

$$A_2 = \begin{bmatrix} M_{A0} & & & \\ & M_{A1} & & \\ & & M_A & \\ & & & \ddots \end{bmatrix},$$

with

$$L_{A0} = \lambda_a, \qquad L_{A1} = \lambda_a I_3, \qquad L_A = \lambda_a I_4,$$

$$L_{B0} = \begin{bmatrix} \lambda_b & \cdot & \cdot \end{bmatrix}, \quad L_{B1} = \begin{bmatrix} \lambda_b & \cdot & \cdot & \cdot \\ \cdot & \lambda_b & \cdot & \cdot \\ \cdot & \cdot & \lambda_b & \cdot \end{bmatrix}, \quad L_B = \lambda_b I_4,$$

$$M_{A0} = 2\mu_a, \qquad M_{A1} = \begin{bmatrix} \mu_a & \mu_a & \cdot \\ \cdot & \mu_a & \cdot \\ \cdot & \cdot & \mu_a \end{bmatrix}, \quad M_A = \begin{bmatrix} \mu_a & \mu_a & \cdot & \cdot \\ \cdot & \mu_a & \cdot & \cdot \\ \cdot & \cdot & \cdot & \mu_a \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix},$$

$$M_{B0} = \begin{bmatrix} \cdot \\ \mu_b \\ \mu_b \end{bmatrix}, \quad M_{B1} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \mu_b & \cdot \\ \mu_b & \cdot & \cdot \\ \cdot & \mu_b & \mu_b \end{bmatrix}, \quad M_B = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \mu_b & \cdot & \cdot \\ \mu_b & \cdot & \cdot & \cdot \\ \cdot & \mu_b & \cdot & \mu_b \end{bmatrix}.$$

The states in the subblocks of A_0, A_1 and A_2 are ordered lexicographically, as in Table 1. Zero elements in these subblocks are indicated by a dot. The structure of the boundary matrices (B_{00}, B_{01}, B_{10}, B_{11}, B_{12} and B_{21}) is more messy and is therefore omitted here; details may be found in [Lee98].

Neuts [Neu81] has shown that the invariant probability vector $\boldsymbol{\pi}$ of a QBD has a matrix-geometric form. For our model, it means that

$$\boldsymbol{\pi}_i = \boldsymbol{\pi}_2 R^{i-2}, \qquad i \geq 2, \tag{2}$$

where $\boldsymbol{\pi}_i$ is the subvector of $\boldsymbol{\pi}$ corresponding to the level $\ell(i)$; it is in turn composed of subvectors $\boldsymbol{\pi}_{i,j}$ corresponding to each major phase $\varpi(j)$ and containing one entry for each minor phase. R is the *rate matrix*; it is the minimal nonnegative solution to the matrix-quadratic equation

$$A_0 + RA_1 + R^2 A_2 = 0.$$

The boundary probability vectors $\boldsymbol{\pi}_0$, $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ are defined by solving

$$\begin{aligned}
\boldsymbol{\pi}_0 B_{00} + \boldsymbol{\pi}_1 B_{10} &= \mathbf{0}, \\
\boldsymbol{\pi}_0 B_{01} + \boldsymbol{\pi}_1 B_{11} + \boldsymbol{\pi}_2 B_{21} &= \mathbf{0}, \\
\boldsymbol{\pi}_1 B_{12} + \boldsymbol{\pi}_2 (A_1 + R A_2) &= \mathbf{0}, \\
\boldsymbol{\pi}_0 \mathbf{1} + \boldsymbol{\pi}_1 \mathbf{1} + \boldsymbol{\pi}_2 (I - R)^{-1} \mathbf{1} &= 1.
\end{aligned}$$

Several algorithms have been developed to determine the rate matrix of a QBD, the most efficient one so far being the algorithm LR (the Logarithmic Reduction algorithm, see Latouche and Ramaswami [LR93]). The algorithms apply for finite as well as for infinite A-matrices, but a numerical implementation requires them to be finite. We shall therefore have to truncate on the major phase in each level, more specifically, we shall not allow the number of class B jobs in the system to exceed M. The effect of this truncation on the results of the analysis is discussed in Leemans [Lee98] and in Leemans and Dedene [LD98].

If we define $P(i,j)$ as the limiting probability that $n_a = i$ and $n_b = j$, we have that $P(i,j) = \boldsymbol{\pi}_i \mathbf{z}_{i,j}$, where $\mathbf{z}_{i,j}$ is a column vector of the same size as the corresponding row vector $\boldsymbol{\pi}_i$, with ones on the positions corresponding to the major phase $\varpi(j)$ and with zeroes elsewhere. In practice, we have to define $\mathbf{z}_{i,j}$ for $i = 0, 1, 2$ only, because the matrix-geometric property (2) allows us to write $P(i,j) = \boldsymbol{\pi}_2 R^{i-2} \mathbf{z}_{2,j}$, for $i \geq 2$.

The joint stationary distribution is thus completely determined as soon as the rate matrix and the boundary probability vectors have been defined. It is now also straightforward to derive the marginal stationary distributions and their moments. We omit the details here, since they are not needed for the derivation of the waiting time distribution in the next section.

3 Stationary Distribution of Waiting Times

The waiting time is defined as the time between the arrival of a unit to the queue and the moment at which it enters service. Ramaswami and Lucantoni present in [RL85] an efficient algorithm for the derivation of the complementary distribution function of stationary waiting times in phase-type queues and QBD processes; their algorithm is based on the technique of *tagging* and *randomization*. We adapt this algorithm here to account for the paths through the messy boundary states and we establish a set of recursion formulas for the complementary waiting times.

Consider a tagged job of class A entering the system. Its waiting time is first of all determined by the number of class A jobs waiting upon its arrival.

All further class A arrivals have no influence on its waiting time; therefore we set $\lambda_a = 0$. Consequently, the matrices A_0, B_{01} and B_{12}, representing transitions to a higher level, become zero matrices.

Waiting is over if the tagged job enters service; the states in which this occurs are the *absorbing* states and the waiting time is simply the time until *absorption*. Some of the states in the boundary levels $\ell(0)$ and $\ell(1)$ are absorbing states, while other states are non-absorbing. The absorbing states in $\ell(0)$ and $\ell(1)$ are aggregated into one absorbing state $\star$:

$$\star = \{(0,0,0,0),(0,1,0,B),(0,1,B,0),(1,0,A,0),(1,0,0,A)\}.$$

If the tagged job, for example, finds only class B jobs and no class A jobs present, which means that the process starts in $\ell(0)$, then still, the class A job must wait until one of the class B jobs leaves since priorities are nonpreemptive. Or, if on the other hand, the process starts in $\ell(1)$ with no class B jobs present, the tagged class A job is immediately served (zero waiting time). We emphasize that, here, the level indicates the number of jobs of class A ahead of the tagged job, not the total number of class A jobs as in the previous section. The probability of starting in $\star$ is then given by $\Pr(\star) = \boldsymbol{\pi}_{0,0}+\boldsymbol{\pi}_{0,1}\mathbf{1}+\boldsymbol{\pi}_{1,0}\mathbf{1}$; this is the stationary probability $W(0)$ that there is no waiting at all. The absorbing states in $\star$ are removed from $\ell(0)$ and $\ell(1)$ and from their probability vectors $\boldsymbol{\pi}_0$ and $\boldsymbol{\pi}_1$, so that $\tilde{\boldsymbol{\pi}}_0 = (\boldsymbol{\pi}_{0,2}, \boldsymbol{\pi}_{0,3}, \dots)$ and $\tilde{\boldsymbol{\pi}}_1 = (\boldsymbol{\pi}_{1,1}, \boldsymbol{\pi}_{1,2}, \dots)$.

The generator for this *service process* is now given by

$$\tilde{Q} = \left[\begin{array}{c|cccccc} 0 & 0 & & & & & \\ \hline b_0 & \tilde{B}_{00} & 0 & & & & \\ b_1 & \tilde{B}_{10} & \tilde{B}_{11} & 0 & & & \\ b_2 & & \tilde{B}_{21} & \tilde{A}_1 & 0 & & \\ & & & A_2 & \tilde{A}_1 & \ddots & \\ & & & & \ddots & \ddots & \end{array}\right]. \tag{3}$$

The first row and column correspond to the absorbing state $\star$. The column vectors b_0, b_1 and b_2 contain elements that indicate transitions from $\ell(0)$, $\ell(1)$ and $\ell(2)$ into the absorbing state $\star$. We omit the details on the structure of the vectors b_0, b_1 and b_2 and of the matrices $\tilde{B}_{00}$, $\tilde{B}_{10}$, $\tilde{B}_{11}$ and $\tilde{B}_{21}$; they are presented in [Lee98]. The matrix $\tilde{A}_1$ in (3) is equal to A_1 in (1), with $\lambda_a = 0$.

We now apply the technique of randomization, i.e., we uniformize the service process with a Poisson process with rate $\theta = \max_i(-\tilde{Q})_{ii}$, so that the

generator is transformed into the discrete time transition probability matrix

$$K = \frac{1}{\theta}\tilde{Q} + I,$$

$$= \left[\begin{array}{c|cccccc} 1 & 0 & & & & & \\ \hline c_0 & K_{00} & & & & & \\ c_1 & K_{10} & K_{11} & & & & \\ c_2 & & K_{21} & K_1 & & & \\ & & & K_2 & K_1 & & \\ & & & & \ddots & \ddots & \end{array}\right].$$

In this uniformized process, points of a Poisson process are generated with rate θ and transitions occur at these epochs only. The probabilities that, at such an epoch, a transition is made towards a lower level are given by K_2 for all $\ell(k)$ $(k > 2)$, K_{21} for a transition from $\ell(2)$ to $\ell(1)$, or K_{10} for a transition from $\ell(1)$ to $\ell(0)$. The probabilities that, at such an epoch, a transition only involves a change in the major phase are given by K_1 for all $\ell(k)$ $(k \geq 2)$, K_{11} for $\ell(1)$, or K_{00} for $\ell(0)$. From this transition matrix, we calculate the stationary probabilities $P[W \geq x]$ that the process is *not* absorbed at time x. The probability that n Poisson points are generated in time x equals $e^{-\theta x}(\theta x)^n/n!$. Then, the process that starts in $\ell(k)$ is not absorbed at time x if at most k of the n transitions generated in time x involve a decrease of the level.

We now define the matrix $T_i^{(n)}$ to be such that it records the probability that the process goes down i homogeneous levels in n steps. It is easy to see that

$$T_i^{(n)} = T_{i-1}^{(n-1)}K_2 + T_i^{(n-1)}K_1, \tag{4}$$

with

$$T_i^{(n)} = \begin{cases} I, & i = n = 0, \\ 0, & n > i. \end{cases}$$

We also define the matrix $\tilde{T}_{k,i}^{(n)}$; this matrix records the conditional probability that the process goes down to $\ell(i)$, $i = 0$ or 1, in n steps, given that it started in $\ell(k)$:

$$\tilde{T}_{k,i}^{(n)} = \tilde{T}_{k,i+1}^{(n-1)}K_{i+1,i} + \tilde{T}_{k,i}^{(n-1)}K_{ii}, \qquad k \leq 2, \tag{5}$$

$$= T_{k-2} \otimes \tilde{T}_{2,i}, \qquad k > 2, \tag{6}$$

with

$$\tilde{T}_{2,2}^{(n)} = T_0^{(n)} \qquad \text{and} \qquad \tilde{T}_{k,i}^{(n)} = \begin{cases} I, & i = k \quad \text{AND} \quad n = 0, \\ 0, & i > k \quad \text{OR} \quad n > k - i. \end{cases}$$

Expression (6) requires a few words of explanation. If the process starts in a level $\ell(k)$, $k > 2$, and goes down to a level $\ell(i)$, $i = 0$ or 1, it makes a number of transitions through the homogeneous levels first and goes further down then to, or through, the boundary levels. The transition matrices for the boundary levels differ from the transition matrices in the homogeneous levels. Therefore, we write the probability of going down from $\ell(k)$, $k \geq 3$, to $\ell(i)$, $i = 0, 1$, in n steps as the probability of going down to $\ell(2)$ first and then moving from $\ell(2)$ to $\ell(i)$. This is denoted in (6) by the convolution ($\otimes$). We may now also write (6) as

$$\tilde{T}_{k,i}^{(n)} = \sum_{t=1}^{n} T_{k-3}^{(t-1)} K_2 \tilde{T}_{2,i}^{(n-t)}. \tag{7}$$

This equation gives the probability that – for any value of t ($1 \leq t \leq n$) – the process is in $\ell(3)$ at time $t-1$ (given by $T_{k-3}^{(t-1)}$) AND goes down to $\ell(2)$ for the first time at time t (given by K_2) AND goes down to $\ell(i)$ in $n-t$ steps (given by $\tilde{T}_{2,i}^{(n-t)}$).

Let $\overline{W}_k(x)$ denote the conditional probability that the process is not absorbed at time x, given that it starts in $\ell(k)$. The probability that the process starts in $\ell(k)$ is – by the PASTA property – given by $\boldsymbol{\pi}_k$, as derived in the previous section. The stationary probability $\overline{W}(x)$ that a unit of class A has to wait more than x units of time before entering service is then clearly given by

$$\overline{W}(x) = \sum_{k=0}^{\infty} \boldsymbol{\pi}_k \overline{W}_k(x) \mathbf{1}.$$

As it was shown in [RL85],

$$\overline{W}(x) = \sum_{n=0}^{\infty} e^{-\theta x} \frac{(\theta x)^n}{n!} d_n. \tag{8}$$

The series d_n is calculated only once for all values of x. The infinite sum in (8) is truncated at $n = N$ so that $d_N < \varepsilon$, with ε arbitrarily small. The smaller ε, the higher the accuracy of the waiting time distribution. The stationary waiting time distribution is then simply

$$W(x) = \begin{cases} 1 - \overline{W}(x), & \forall x > 0, \\ \Pr(\star), & x = 0. \end{cases} \tag{9}$$

We now derive an expression for d_n. From the definition of the service process, we know that it is absorbed when it enters the state $\star$. If the process

starts in $\ell(0)$, it is not absorbed until it goes one level down. If the process starts in $\ell(k)$, it can go down at most k levels without being absorbed in $\star$. $\overline{W}(x)$ is then written as the sum of

$\tilde{\boldsymbol{\pi}}_0\overline{W}_0(x)\mathbf{1}$:	the probability of starting in one of the non-absorbing states of level $\ell(0)$ and not being absorbed at time x,
$\tilde{\boldsymbol{\pi}}_1\overline{W}_1(x)\mathbf{1}$:	the probability of starting in one of the non-absorbing states of level $\ell(1)$ and not being absorbed at time x, and
$\sum_{k=2}^{\infty}\boldsymbol{\pi}_k\overline{W}_k(x)\mathbf{1}$:	the probability of starting in a level $\ell(k)$, $k \geq 2$ and not being absorbed at time x.

$\boxed{\tilde{\boldsymbol{\pi}}_0\overline{W}_0(x)\mathbf{1}}$

Starting from $\ell(0)$, the process is not absorbed at time x if it has not gone down in any of the n steps generated by the Poisson process with parameter θ during the interval $(0, x]$. The conditional probability of not going down in n steps, given that the process started in $\ell(0)$, is given by the matrix $\tilde{T}_{0,0}^{(n)}$. Conditioning on the number of Poisson points generated in the interval $(0, x]$, it follows that

$$\tilde{\boldsymbol{\pi}}_0\overline{W}_0(x)\mathbf{1} = \sum_{n=0}^{\infty} e^{-\theta x}\frac{(\theta x)^n}{n!}\tilde{\boldsymbol{\pi}}_0\tilde{T}_{0,0}^{(n)}\mathbf{1}. \tag{10}$$

From the recursion formula (5), it is clear that

$$\tilde{T}_{0,0}^{(n)} = \tilde{T}_{0,0}^{(n-1)}K_{00}, \tag{11}$$

with $\tilde{T}_{0,0}^{(0)} = I$, since no transition can be made in zero steps.

$\boxed{\tilde{\boldsymbol{\pi}}_1\overline{W}_1(x)\mathbf{1}}$

Starting from $\ell(1)$, the process is not absorbed if, at time x, it is in one of the non-absorbing states of $\ell(1)$ or $\ell(0)$. The conditional probability of remaining in $\ell(1)$ in each of the n steps is given by the matrix $\tilde{T}_{1,1}^{(n)}$; the conditional probability of going down from $\ell(1)$ to $\ell(0)$ in n steps is given by the matrix $\tilde{T}_{1,0}^{(n)}$. Proceeding as for $\tilde{\boldsymbol{\pi}}_0\overline{W}_0(x)\mathbf{1}$, we can write

$$\tilde{\boldsymbol{\pi}}_1\overline{W}_1(x)\mathbf{1} = \sum_{n=0}^{\infty} e^{-\theta x}\frac{(\theta x)^n}{n!}\tilde{\boldsymbol{\pi}}_1\left(\tilde{T}_{1,1}^{(n)}\mathbf{1} + \tilde{T}_{1,0}^{(n)}\mathbf{1}\right). \tag{12}$$

Again, from (5), it follows that

$$\tilde{T}_{1,1}^{(n)} = \tilde{T}_{1,1}^{(n-1)} K_{11}, \tag{13}$$

$$\tilde{T}_{1,0}^{(n)} = \tilde{T}_{1,1}^{(n-1)} K_{10} + \tilde{T}_{1,0}^{(n-1)} K_{00}, \tag{14}$$

with $\tilde{T}_{1,1}^{(0)} = I$ and $\tilde{T}_{1,0}^{(0)} = 0$, since no transition can be made in zero steps.

$\boxed{\sum_{k=2}^{\infty} \boldsymbol{\pi}_k \overline{W}_k(x)\mathbf{1}}$

Given that the process starts in $\ell(k)$ ($k \geq 2$), it is not absorbed if, at time x, it is in some homogeneous level between $\ell(k)$ and $\ell(2)$, or in one of the non-absorbing states of the boundary levels $\ell(1)$ or $\ell(0)$. The probability of starting in $\ell(k)$ and going down at most $k-2$ homogeneous levels in n steps is given by

$$\boldsymbol{\pi}_2 R^{k-2} \sum_{i=0}^{k-2} T_i^{(n)} \mathbf{1}. \tag{15}$$

The probability of starting in $\ell(k)$ and going down to $\ell(1)$ in n steps is

$$\boldsymbol{\pi}_2 R^{k-2} \tilde{T}_{k,1}^{(n)} \mathbf{1}. \tag{16}$$

Finally, the probability of starting in $\ell(k)$ and going down to $\ell(0)$ in n steps equals

$$\boldsymbol{\pi}_2 R^{k-2} \tilde{T}_{k,0}^{(n)} \mathbf{1}. \tag{17}$$

Combining (15), (16) and (17), summing over all levels $k \geq 2$ and conditioning on the number of Poisson points generated in the interval $(0, x]$, $\sum_{k=2}^{\infty} \boldsymbol{\pi}_k \overline{W}_k(x)\mathbf{1}$ becomes

$$\sum_{k=2}^{\infty} \boldsymbol{\pi}_k \overline{W}_k(x)\mathbf{1} = \sum_{n=0}^{\infty} e^{-\theta x} \frac{(\theta x)^n}{n!} \boldsymbol{\pi}_2 \left\{ \sum_{k=2}^{\infty} R^{k-2} \sum_{i=0}^{k-2} T_i^{(n)} \mathbf{1} + \Delta^{(n)}\mathbf{1} + \Omega^{(n)}\mathbf{1} \right\}, \tag{18}$$

with

$$\Delta^{(n)} = \sum_{k=2}^{\infty} R^{k-2} \tilde{T}_{k,1}^{(n)} \qquad \text{and} \qquad \Omega^{(n)} = \sum_{k=2}^{\infty} R^{k-2} \tilde{T}_{k,0}^{(n)}.$$

The first term in the curly braces of (18) is simplified proceeding as in [RL85], so that

$$\begin{aligned}
\sum_{k=2}^{\infty} R^{k-2} \sum_{i=0}^{k-2} T_i^{(n)} \mathbf{1} &= \sum_{k=0}^{n} R^k \sum_{i=0}^{k} T_i^{(n)} \mathbf{1} + \sum_{k=n+1}^{\infty} R^k \sum_{i=0}^{n} T_i^{(n)} \mathbf{1}, \\
&\qquad \text{since } T_i^{(n)} = 0 \text{ for } i > n, \\
&= \sum_{i=0}^{n} \sum_{k=i}^{n} R^k T_i^{(n)} \mathbf{1} + (I-R)^{-1} R^{n+1} \mathbf{1}, \\
&\qquad \text{since } \textstyle\sum_{i=0}^{n} T_i^{(n)} \mathbf{1} = \mathbf{1}, \\
&= (I-R)^{-1} \sum_{i=0}^{n} R^i T_i^{(n)} \mathbf{1}, \\
&= (I-R)^{-1} H^{(n)} \mathbf{1}, \qquad (19)
\end{aligned}$$

with $H^{(n)} = \sum_{i=0}^{n} R^i T_i^{(n)}$. A recursion formula for $H^{(n)}$ is developed using the formula (4):

$$\begin{aligned}
H^{(n)} &= \sum_{i=0}^{n} R^i T_i^{(n)}, \\
&= \sum_{i=0}^{n} R^i \left(T_{i-1}^{(n-1)} K_2 + T_i^{(n-1)} K_1 \right), \\
&= \sum_{i=1}^{n} R^i T_{i-1}^{(n-1)} K_2 + \sum_{i=0}^{n-1} R^i T_i^{(n-1)} K_1, \\
H^{(n)} &= R H^{(n-1)} K_2 + H^{(n-1)} K_1, \qquad (20)
\end{aligned}$$

with $H^{(0)} = I$.

Next, $\Delta^{(n)}$ is simplified as follows:

$$\begin{aligned}
\Delta^{(n)} &= \sum_{k=2}^{\infty} R^{k-2} \tilde{T}_{k,1}^{(n)}, \\
&= \sum_{k=3}^{\infty} R^{k-2} T_{k-2} \otimes \tilde{T}_{2,1} + \tilde{T}_{2,1}^{(n)}, && \text{by (6)}, \\
&= \sum_{k=3}^{\infty} R^{k-2} \sum_{t=1}^{n} T_{k-3}^{(t-1)} K_2 \tilde{T}_{2,1}^{(n-t)} + \tilde{T}_{2,1}^{(n)}, && \text{by (7)}, \\
&= \sum_{t=1}^{n} R \sum_{k=0}^{\infty} R^k T_k^{(t-1)} K_2 \tilde{T}_{2,1}^{(n-t)} + \tilde{T}_{2,1}^{(n)}, \\
&= R \sum_{t=1}^{n} H^{(t-1)} K_2 \tilde{T}_{2,1}^{(n-t)} + \tilde{T}_{2,1}^{(n)}.
\end{aligned}$$

A recursion formula for $\Delta^{(n)}$ is now established by

$$\begin{aligned}
\Delta^{(n)} &= R\sum_{t=1}^{n-1} H^{(t-1)}K_2\tilde{T}_{2,1}^{(n-t)} + RH^{(n-1)}K_2\tilde{T}_{2,1}^{(0)} + \tilde{T}_{2,1}^{(n)}, \\
&= R\sum_{t=1}^{n-1} H^{(t-1)}K_2\left(T_0^{(n-t-1)}K_{21} + \tilde{T}_{2,1}^{(n-t-1)}K_{11}\right) \\
&\qquad + \left(T_0^{(n-1)}K_{21} + \tilde{T}_{2,1}^{(n-1)}K_{11}\right), \\
&\qquad\qquad \text{by (5) and since } \tilde{T}_{2,1}^{(0)} = 0, \\
&= \left(R\sum_{t=1}^{n-1} H^{(t-1)}K_2T_0^{(n-t-1)} + T_0^{(n-1)}\right)K_{21} \\
&\qquad + \left(R\sum_{t=1}^{n-1} H^{(t-1)}K_2\tilde{T}_{2,1}^{(n-t-1)} + \tilde{T}_{2,1}^{(n-1)}\right)K_{11}, \\
\Delta^{(n)} &= \Phi^{(n-1)}K_{21} + \Delta^{(n-1)}K_{11}, \qquad (21)
\end{aligned}$$

with $\Phi^{(n)} = R\sum_{t=1}^{n} H^{(t-1)}K_2T_0^{(n-t)} + T_0^{(n)}$ and $\Delta^{(0)} = 0$.

The recursion (21) for $\Delta^{(n)}$ in turn requires a recursion formula for $\Phi^{(n)}$, which is obtained proceeding as for (21) and which is given by

$$\Phi^{(n)} = \Phi^{(n-1)}K_1 + RH^{(n-1)}K_2, \qquad (22)$$

with $\Phi^{(0)} = I$.

We simplify $\Omega^{(n)}$ in (18) in a similar way as we did for $\Delta^{(n)}$ and we obtain that

$$\Omega^{(n)} = R\sum_{t=1}^{n} H^{(t-1)}K_2\tilde{T}_{2,0}^{(n-t)} + \tilde{T}_{2,0}^{(n)},$$

with the recursion formula

$$\Omega^{(n)} = \Delta^{(n-1)}K_{10} + \Omega^{(n-1)}K_{00}, \qquad (23)$$

where $\Omega^{(0)} = 0$.

We now insert (19) into (18) and sum all components (10), (12) and (18) of $\overline{W}(x)$ to find (8), with

$$d_n = \tilde{\boldsymbol{\pi}}_0 \tilde{T}_{0,0}^{(n)} \mathbf{1} + \tilde{\boldsymbol{\pi}}_1 \left(\tilde{T}_{1,1}^{(n)} + \tilde{T}_{1,0}^{(n)} \mathbf{1} \right) \mathbf{1}$$

$$+ \boldsymbol{\pi}_2 \left[(I - R)^{-1} H^{(n)} \mathbf{1} + \Delta^{(n)} \mathbf{1} + \Omega^{(n)} \mathbf{1} \right]. \qquad (24)$$

The values of d_n are calculated iteratively, starting with $n = 0$ and using the recursion formulas (11), (13), (14), (20), (21), (22) and (23).

Note that the algorithm results in the distribution of waiting times for the class on the level of the process. It is straightforward to derive the distribution for the other class by simply switching the classes.

4 Numerical examples

The algorithms for the queue length and waiting time distributions are easily implemented. For the queue lengths, most of the computation effort is spent on the calculation of the rate matrix R. It was shown in Latouche and Ramaswami [LR93] that the execution time for the algorithm LR is $O(m^3)$, where m is the number of phases. In our calculations, the number of iterations within LR depends on the load of the class on the level, while the computation time per iteration increases with the number of major phases, indicated by the truncation parameter M. The execution time for the waiting time algorithm mainly increases with the desired level of accuracy as well as with the load of the class on the level and the number of major phases.

As an illustration, we have calculated and plotted in Figure 2 the stationary distribution of waiting times for class A and class B. The load of class A is fixed and equal to 0.9, class B load is successively changed from 0.3 to 0.6 and 0.9. Class B jobs are twice as large as class A jobs, i.e., $\mu_a = 2\mu_b$, with μ_a equal to unity.

The graphs clearly illustrate the effect of varying class B load on the waiting times for class A. Unlike in the FCFS M/M/2 queue – where waiting times are equal for both classes (see Figure 3) – the impact here is stronger for class B than for class A. Notice also how the waiting times differ from those of the M/M/1 queue, as illustrated in Figure 4. The differences arise from a 'limited sharing' of the servers in our model, a property that makes of this model an appealing scheduling alternative in other situations as well.

The algorithms are readily applicable for similar models with multiple classes and/or servers. The execution time, however, will strongly increase as the number of minor phases in each major phase increases with each additional class and server (see [Lee98]).

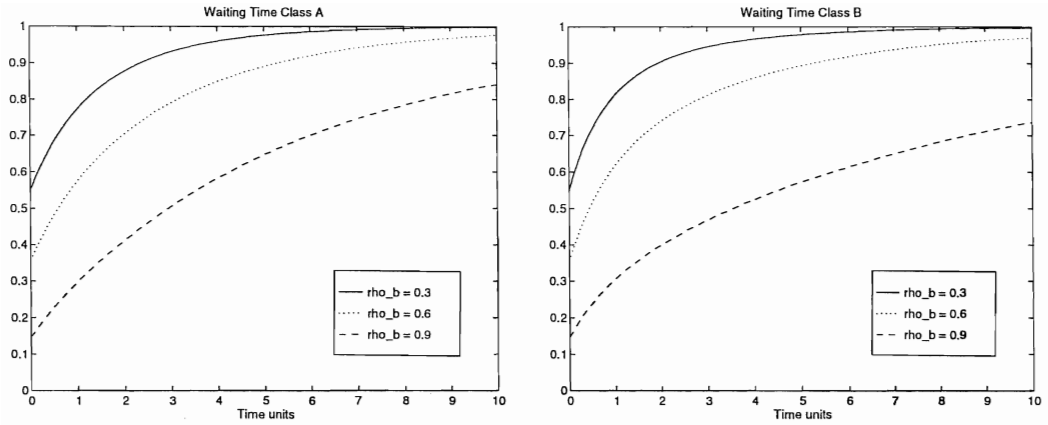

Figure 2: Stationary distribution of waiting times for class A and class B with $\mu_a = 1$, $\mu_b = 0.5$ and $\rho_a = 0.9$.

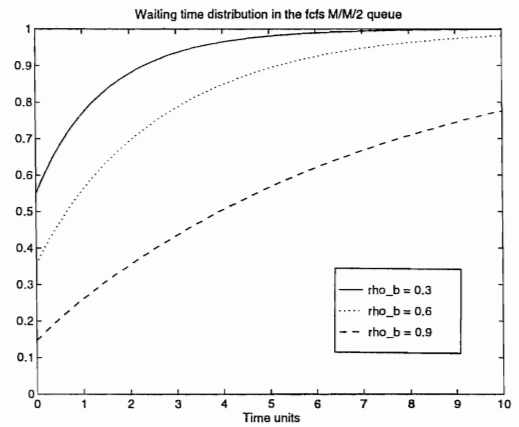

Figure 3: Waiting times in the FCFS M/M/2 queue.

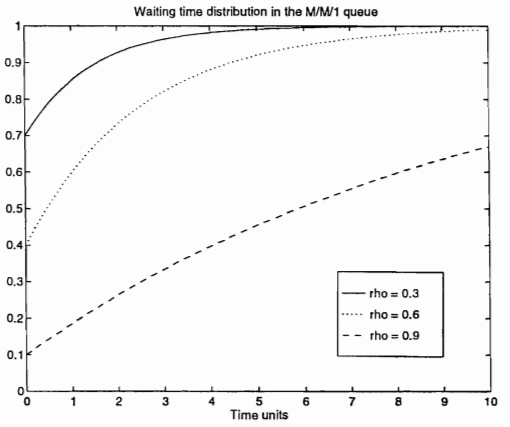

Figure 4: Waiting times in the M/M/1 queue.

Acknowledgement

The authors would like to thank Guy Latouche (U.L.B., Belgium) for his invaluable help in this research.

References

[FI79] G. Fayolle and R. Iasnogorodski. Two coupled processors: the reduction to a Riemann-Hilbert problem. *Z. Wahrscheinlichkeitsth.*, 47:325–351, 1979.

[FKM82] G. Fayolle, P. J. B. King, and I. Mitrani. The solution of certain two-dimensional Markov models. *Adv. Appl. Prob.*, 14:295–308, 1982.

[GHT88] H. R. Gail, S. L. Hantler, and B. A. Taylor. Analysis of a non-preemptive priority multiserver queue. *Adv. Appl. Prob.*, 20:852–879, 1988.

[GHT92] H. R. Gail, S. L. Hantler, and B. A. Taylor. On a preemptive Markovian queue with multiple servers and two priority classes. *Mathematics of Operations Research*, 17:365–391, 1992.

[KN91] E. P. C. Kao and K. S. Narayanan. Modeling a multiprocessor system with preemptive priorities. *Mgmt. Sci.*, 37(2):185–197, 1991.

[LD98] H. Leemans and G. Dedene. Bounds for the mean queue lengths in a two-class two-server system with heterogeneous priority. To appear in *Proceedings of the 2nd International Conference on Matrix-Analytic Methods in Stochastic Models*, A. Alfa and S. Chakravarthy (editors), Notable Publications Inc., New Jersey, 1998.

[Lee98] H. Leemans. *The Two-Class Two-Server Queueing Model with Nonpreemptive Heterogeneous Priority Structures.* PhD thesis, K.U.Leuven, Department of Applied Economic Sciences, 1998.

[LR93] G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for Quasi-Birth-Death processes. *J. Appl. Prob.*, 30:650–674, 1993.

[Mil92] D. R. Miller. Steady-state algorithmic analysis of M/M/c two-priority queues with heterogeneous servers. In R. L. Disney and T. J. Ott, editors, *Applied Probability - Computer Science, The Interface*, volume II, pages 207–222. Birkhaüser, Boston, 1992.

[MK81] I. Mitrani and P. J. B. King. Multiprocessor systems with preemptive priorities. *Performance Evaluation*, 1:118–125, 1981.

[Neu81] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models. An Algorithmic Approach.* The John Hopkins University Press, Baltimore, Md., 1981.

[RL85] V. Ramaswami and D. Lucantoni. Stationary waiting time distribution in queues with phase type service and in Quasi-Birth-and-Death processes. *Commun. Statist. — Stochastic Models*, 1(2):125–136, 1985.

[RP86] B. M. Rao and M. J. M. Posner. Parallel exponential queues with dependent service rates. *Computers and Operations Research*, 13(6):681–692, 1986.