Joint queue length distribution of multi-class, single-server queues with preemptive priorities

Joint queue length distribution of multi-class, single-server queues with preemptive priorities

A. Sleptchenko*
EURANDOM and Department of Technology Management
Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

I.J.B.F. Adan
Department of Mathematics and Computer Science
Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

G.J. van Houtum
Department of Technology Management
Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

October 4, 2004

Abstract

In this paper we analyze an $M_N/M_N/1$ queueing system with N customer classes and preemptive priorities between classes, by using matrix-analytic techniques. This leads to an exact method for the computation of the steady state joint queue length distribution. We also indicate how the method can be extended to models with multiple servers and other priority rules.

Keywords: priority queues; preemptive priority; $M_N/M_N/1$ queueing systems; matrix-analytic method; joint queue length distribution

*E-mail address: A.Sleptchenko@tue.nl

1 Introduction

Consider a single server queueing system shared by N customer classes, numbered $1, \ldots, N$. The arrival process of class i is Poisson with rate λ_i. The service times of class i customers are exponentially distributed with rate μ_i. The class index i indicates the priority rank; class 1 has lowest priority and class N has highest priority. The priority rule is preemptive, i.e., service may be interrupted by higher priority customers.

In this paper we present an exact method, based on matrix-analytic techniques, for finding the steady state joint queue length distribution. This distribution is required in many applications in the area of spare parts logistics and production.

Priority queueing systems have a long history (cf. Cobham [3], Davis [4], Jaiswal [9]) and single and multi server priority queues received much attention. Most of the earlier studies, however, concentrate on the transforms of *marginal* system characteristics (e.g., class i queue length and waiting time). Our interest in the problem arose from a spare parts logistics problem, see Subsection 4.2, where the *joint* queue length distribution is necessary for an exact performance analysis.

The first results for the joint queue length distribution of priority queueing systems are derived by Miller [10], who analyzes the $M_2/M_2/1$ priority queueing system by the matrix-geometric method. Later Alfa [1, 2], Isotupa [8] and Wagner [18, 19] observe that the matrix-geometric method is a natural choice for studying priority queueing systems with a Quasi-Birth-and-Death (QBD) structure. In these papers Miller's method is generalized to systems with a Markovian Arrival Process and phase type service time distributions, which require a more detailed registration of the arrival and service process. Also in [2, 8] the authors apply the matrix-geometric method to an N-class system. However they do not observe that lower priority customers see the queueing system as an $M/G/1$-type system, i.e., an $M/M/1$ system with an unreliable server (where the down times correspond to high priority service interruptions). This suggests to use the matrix-analytic method for $M/G/1$-type systems, and indeed, in the present paper it appears that this method leads to an elegant recursive analysis of priority queueing systems with $N \geq 2$ classes of customers.

There is also a number of papers studying the joint queue length distribution using different approaches. Gail et al. [5, 6] and Mitrani and King [11] use generating functions for the analysis of $M_2/M_2/k$ priority queueing systems. Later Sleptchenko et al. [17] and Sleptchenko [16] use a mixture of matrix-geometric and generating function approaches to analyze preemptive and non-preemptive priority $M_N/M_N/k$ queueing systems, where each class of customers has either

high or low priority.

In this paper we apply the matrix-analytic method (see Neuts [13], Ramaswami [14]), and in doing so, we exploit the $M/G/1$ structure of the Markov process describing the preemptive priority system with N customer classes. This yields an efficient algorithm for the exact computation of the steady state queue length distribution.

To this aim, in Section 2, we first present the formal description of the single server priority system, and formulate the steady state equations. In Section 3, we apply the matrix-analytic method. The proposed method is based on the matrix formulation of the steady state equations, using matrices with a complex nested structure. To ease the understanding of the nested structure and of the method in general, we first show in Section 3.1 how the 2-class model can be formulated and solved using the matrix-analytic method for $M/G/1$-type systems. Next, in Section 3.2, we show how the analysis can be extended to a 3-class system and highlight the recursive nature of the approach. Finally, in Section 3.3, we present the main steps of the algorithm for the general N-class case. After that, in Section 4, we discuss some implementation issues and present computational results for a spare parts problem. In the final section we describe how the proposed method can be applied to other queueing systems such as the multi-server $M_N/M_N/k$ preemptive and non-preemptive priority system.

2 Model and steady state equations

Throughout the paper we assume that the queueing system is stable, i.e. the traffic intensity ρ is less than 1 (see for example [7]):

$$\rho = \sum_{i=1}^{N} \lambda_i/\mu_i < 1.$$

The states of the $M_N/M_N/1$ preemptive priority system can be described by N-dimensional vectors $\mathbf{q} = (q_N, \ldots, q_1)$, where q_i is the number of class i customers in the system.

The state transitions are caused by an arrival or a service completion. From the preemptive priority rule it follows that, when a class i customer is in service, there are no higher priority customers in the system, i.e., $q_N = \cdots = q_{i+1} = 0$ whenever $q_i > 0$. Formally, the state transitions can be written as follows:

$$\begin{aligned}
(q_N, \ldots, q_1) &\xrightarrow{\lambda_i} (q_N, \ldots, q_i + 1, \ldots, q_1), \quad q_j \geq 0,\ j = 1, \ldots, N \\
(0, \ldots, 0, q_i, \ldots, q_1) &\xrightarrow{\mu_i} (0, \ldots, 0, q_i - 1, \ldots, q_1), \quad q_N = \cdots = q_{i+1} = 0,\ q_i > 0.
\end{aligned}$$

Given these transitions, the balance equations for the steady state probabilities $p_{\mathbf{q}}$ can be easily

obtained:

$$p_{\mathbf{q}}\left(\sum_{j=1}^{N}\lambda_j+\mu_N\right) = \left(\sum_{j=1}^{N}p_{\mathbf{q}-\mathbf{e}_j}\lambda_j\right)+p_{\mathbf{q}+\mathbf{e}_N}\mu_N, \quad \mathbf{q}\text{ s.t. } q_N>0 \tag{1}$$

$$\vdots$$

$$p_{\mathbf{q}}\left(\sum_{j=1}^{N}\lambda_j+\mu_i\right) = \left(\sum_{j=i}^{N}p_{\mathbf{q}-\mathbf{e}_j}\lambda_j\right) + \left(\sum_{j=i}^{N}p_{\mathbf{q}+\mathbf{e}_j}\mu_j\right), \quad \mathbf{q}\text{ s.t. } q_N=\cdots=q_{i+1}=0, q_i>0 \tag{2}$$

$$\vdots$$

$$p_{\mathbf{q}}\left(\sum_{j=1}^{N}\lambda_j\right) = \left(\sum_{j=1}^{N}p_{\mathbf{q}+\mathbf{e}_j}\mu_j\right), \quad \mathbf{q}=\mathbf{0} \tag{3}$$

where $\mathbf{e}_i$ is the vector with all components equal to 0 except for the i-th component being 1, and by convention, $p_{\mathbf{q}}=0$ if $q_j<0$ for some j.

3 Matrix-analytic approach

Miller [10] has shown that a two-class $M_2/M_2/1$ queue with preemptive priority behaves as a QBD process, where level $i\geq 0$ is the set of all states with i high priority customers in the system. Although the levels have infinitely many states, the standard matrix-geometric method (see e.g. [12]) can be used to analyze the two-class priority system.

However, the N-class priority system has a much more complicated structure, for which the matrix-analytic method for $M/G/1$-type queueing systems appears to be more appropriate, as will be shown later. We will show first how the 2-class and 3-class system can be solved by the matrix-analytic method.

3.1 Matrix-analytic approach for 2-class systems

The transition rate diagram of the $M_2/M_2/1$ preemptive priority queue is shown in Figure 1.

This queueing system behaves as a QBD process, where level $q_2\geq 0$ is the set of all states with q_2 high priority customers in the system. In this process the arrival and departure of a high priority customer corresponds to a birth and death event, respectively. The generator $\mathbb{Q}$

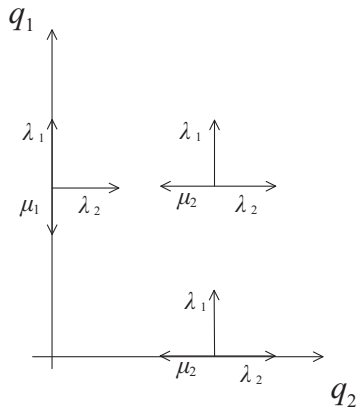

Figure 1: Transition rate diagram of the 2-class $M_2/M_2/1$ preemptive priority queue.

has the following form:

$$\mathbb{Q} = \begin{pmatrix} \tilde{\mathbf{A}}_0 & \lambda_2\mathbf{I} & & & \\ \mu_2\mathbf{I} & \mathbf{A}_0 & \lambda_2\mathbf{I} & & \\ & \mu_2\mathbf{I} & \mathbf{A}_0 & \lambda_2\mathbf{I} & \\ & & \mu_2\mathbf{I} & \mathbf{A}_0 & \ddots \\ & & & \ddots & \ddots \end{pmatrix}$$

The (infinite) block $\mathbf{A}_0$ corresponding to transitions within a level has an upper triangular, two-diagonal structure:

$$\mathbf{A}_0 = \begin{pmatrix} -(\lambda_1+\lambda_2+\mu_2) & \lambda_1 & & \\ & -(\lambda_1+\lambda_2+\mu_2) & \lambda_1 & \\ & & -(\lambda_1+\lambda_2+\mu_2) & \ddots \\ & & & \ddots \end{pmatrix}.$$

Transition rates corresponding to service completions of low priority customers only appear in the boundary block $\tilde{\mathbf{A}}_0$, which has a three-diagonal structure:

$$\tilde{\mathbf{A}}_0 = \begin{pmatrix} -(\lambda_1+\lambda_2) & \lambda_1 & & \\ \mu_1 & -(\lambda_1+\lambda_2+\mu_1) & \lambda_1 & \\ & \mu_1 & -(\lambda_1+\lambda_2+\mu_1) & \ddots \\ & & \ddots & \ddots \end{pmatrix}.$$

As mentioned before, we will apply the standard method for $M/G/1$-type queues. That is, we first derive the fundamental matrix $\mathbf{G}$ for the QBD with generator $\mathbb{Q}$. The matrix $\mathbf{G}$ contains the first passage probabilities from level q_2 to level $q_2 - 1$; i.e., $[\mathbf{G}]_{i,j}$ is the probability to arrive

at state (q_2-1, q_1+j), when level q_2-1 is reached for the first time, starting in state (q_2, q_1+i). These probabilities do not depend on q_2 since the transitions are the same at each level $q_2 > 0$. Note that

$$\sum_{j=0}^{\infty} [\mathbf{G}]_{i,j} = 1,$$

since in a stable system we will move sooner or later from a state at level q_2 to level $q_2 - 1$.

Since low priority customers are not served while there are high priority customers, we cannot move from a state $(q_2, q_1 + i)$, $q_2 > 0$, to a state $(q_2 - 1, q_1 + j)$ with $j < i$. Also, the number of low priority customers arriving during an excursion from state (q_2, q_1) to level $q_2 - 1$ does not depend on the value of q_1. Hence, the fundamental matrix $\mathbf{G}$ has an upper triangular structure with the same elements along diagonals:

$$\mathbf{G} = \begin{pmatrix} g_0 & g_1 & g_2 & \ddots \\ & g_0 & g_1 & \ddots \\ & & g_0 & \ddots \\ & & & \ddots \end{pmatrix}.$$

Element g_i represents the probability that the number of customers in the low priority queue has increased by i at the first passage to level $q_2 - 1$ when starting in state (q_2, q_1). These probabilities can be found recursively. First, g_0 is the probability that no low priority customer arrived when we return to level $q_2 - 1$ after starting at level q_2. Conditioning on the first event in state (q_2, q_1) we obtain:

$$g_0 = \frac{\mu_2}{(\lambda_1 + \lambda_2 + \mu_2)} \cdot 1 + \frac{\lambda_1}{(\lambda_1 + \lambda_2 + \mu_2)} \cdot 0 + \frac{\lambda_2}{(\lambda_1 + \lambda_2 + \mu_2)} \cdot (g_0)^2.$$

In the same way we find the other probabilities g_i:

$$g_i = \frac{\mu_2}{(\lambda_1 + \lambda_2 + \mu_2)} \cdot 0 + \frac{\lambda_1}{(\lambda_1 + \lambda_2 + \mu_2)} \cdot g_{i-1} + \frac{\lambda_2}{(\lambda_1 + \lambda_2 + \mu_2)} \cdot \sum_{j=0}^{i} g_j g_{i-j}, \quad i > 0.$$

Thus, the elements g_i can be obtained from the following recursive equations:

$$\mu_2 - (\lambda_1 + \lambda_2 + \mu_2)\, g_0 + \lambda_2 (g_0)^2 = 0, \qquad i = 0, \tag{4}$$

$$-(\lambda_1 + \lambda_2 + \mu_2)\, g_i + \lambda_1 g_{i-1} + \lambda_2 \sum_{j=0}^{i} g_j g_{i-j} = 0, \qquad i > 0, \tag{5}$$

where g_0 is the minimal non-negative solution of (4). These expressions also immediately follow from the standard matrix equation for the fundamental matrix (cf. [13]):

$$\mu_2 \mathbf{I} + \mathbf{A}_0 \mathbf{G} + \lambda_2 \mathbf{G}^2 = 0$$

The matrix $\mathbf{G}$ can be used to solve the balance equations (1)-(3), in matrix form written as:

$$\begin{aligned} \lambda_2 \mathbf{p}_{q_2-1} + \mathbf{p}_{q_2}\mathbf{A}_0 + \mu_2 \mathbf{p}_{q_2+1} &= 0, \quad q_2 > 0, \qquad (6) \\ \mathbf{p}_0 \tilde{\mathbf{A}}_0 + \mu_2 \mathbf{p}_1 &= 0, \quad q_2 = 0, \qquad (7) \end{aligned}$$

where $\mathbf{p}_{q_2} = (p_{q_2,0}, p_{q_2,1}, p_{q_2,2}, \ldots)$ is the vector of steady state probabilities at level q_2. Further, the balance equations at level q_2 of the Markov process *embedded on the levels* $0, 1, \ldots, q_2$ are given by (see [14]):

$$\begin{aligned} \lambda_2 \mathbf{p}_{q_2-1} + \mathbf{p}_{q_2}\left(\mathbf{A}_0 + \lambda_2 \mathbf{G}\right) = 0, \quad q_2 > 0, \qquad (8) \\ \mathbf{p}_0 \left(\tilde{\mathbf{A}}_0 + \lambda_2 \mathbf{G}\right) = 0, \quad q_2 = 0. \qquad (9) \end{aligned}$$

Combining the equations (6)-(9) immediately yields a simple relation between $\mathbf{p}_{q_2+1}$ and $\mathbf{p}_{q_2}$:

$$\mathbf{p}_{q_2+1} = \frac{\lambda_2}{\mu_2}\mathbf{p}_{q_2}\mathbf{G}, \quad q_2 \geq 0, \qquad (10)$$

which can be written in scalar form as:

$$p_{q_2+1,q_1} = \frac{\lambda_2}{\mu_2}\sum_{i_1=0}^{q_1} p_{q_2,q_1-i_1}\, g_{i_1}, \qquad q_2 \geq 0, q_1 \geq 0. \qquad (11)$$

Hence the vectors $\mathbf{p}_1, \mathbf{p}_2, \ldots$ can be recursively computed, once $\mathbf{p}_0$ is known. The vector $\mathbf{p}_0$ satisfies (9), i.e., the balance equations of the Markov process embedded on the q_1-axis with generator $\tilde{\mathbb{Q}} = \tilde{\mathbf{A}}_0 + \lambda_2 \mathbf{G}$:

$$\tilde{\mathbb{Q}} = \begin{pmatrix} -(\lambda_1+\lambda_2) + \lambda_2 g_0 & \lambda_1 + \lambda_2 g_1 & \lambda_2 g_2 & \\ \mu_1 & -(\lambda_1+\lambda_2+\mu_1) + \lambda_2 g_0 & \lambda_1 + \lambda_2 g_1 & \\ & \mu_1 & -(\lambda_1+\lambda_2+\mu_1) + \lambda_2 g_0 & \ddots \\ & & \ddots & \ddots \end{pmatrix}.$$

Note that $\tilde{\mathbb{Q}}$ is an $M/G/1$-type generator. The balance equation in $(0, q_1)$ can be rewritten as:

$$p_{0,q_1+1} = -\frac{1}{\mu_1}\left[-(\lambda_1+\lambda_2+\mu_1)\, p_{0,q_1} + \lambda_1 p_{0,q_1-1} + \lambda_2 \sum_{i_1=0}^{q_1} g_{i_1} p_{0,q_1-i_1}\right], \quad q_1 > 0. \qquad (12)$$

From (12) the probabilities $p_{0,1}, p_{0,2}, \ldots$ can be recursively computed starting with $p_{0,0} = 1-\rho$. However, equation (12) includes negative terms and therefore might be numerically unstable. This can be avoided by further embedding the Markov process on the states $(0,0), \ldots, (0, q_1+1)$; then the balance equation in $(0, q_1+1)$ is given by:

$$p_{0,q_1+1} = \frac{1}{\mu_1}\left(\lambda_1 p_{0,q_1} + \lambda_2 \sum_{i_1=0}^{q_1} p_{0,q_1-i_1} \sum_{l=i_1+1}^{\infty} g_l\right), \quad q_1 \geq 0.$$

Using that $\sum_{l=0}^{\infty} g_l = 1$, the above equation can be simplified to:

$$p_{0,q_1+1} = \frac{1}{\mu_1}\left[\lambda_1 p_{0,q_1} + \lambda_2 \sum_{i_1=0}^{q_1} p_{0,q_1-i_1}\left(1 - \sum_{l=0}^{i_1} g_l\right)\right], \quad q_1 \geq 0. \tag{13}$$

Equation (13) provides a numerically stable recursion to compute $\mathbf{p}_0$.

3.2 Matrix-analytic approach for 3-class systems

The transition rate diagram of the $M_3/M_3/1$ preemptive priority queue is shown in Figure 2.

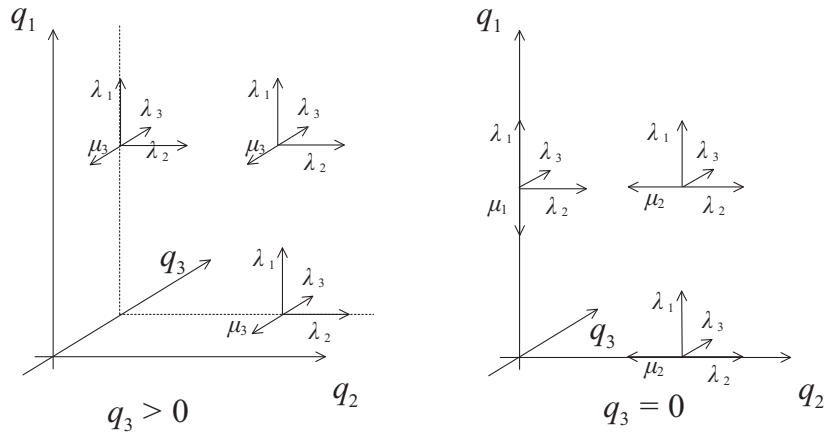

Figure 2: Transition rate diagram of the 3-class $M_3/M_3/1$ preemptive priority queue.

This queueing system behaves again as a QBD process, where level $q_3 \geq 0$ is the set of all states with q_3 high priority class-3 customers in the system. According to this partitioning the generator $\mathbb{Q}^{(3)}$ (the index indicates the number of customers classes) has the following form:

$$\mathbb{Q}^{(3)} = \begin{pmatrix} \tilde{\mathbf{A}}_0 & \lambda_3\mathbf{I} & & & \\ \mu_3\mathbf{I} & \mathbf{A}_0 & \lambda_3\mathbf{I} & & \\ & \mu_3\mathbf{I} & \mathbf{A}_0 & \lambda_3\mathbf{I} & \\ & & \mu_3\mathbf{I} & \mathbf{A}_0 & \ddots \\ & & & \ddots & \ddots \end{pmatrix}$$

However, block $\mathbf{A}_0$ now has a more complicated structure than in the 2-class case; we further partition level q_3 into sublevels, where sublevel q_2 is the set of all states with q_2 class-2 customers (and q_3 class-3 customers) in the system. Then $\mathbf{A}_0$ has an upper triangular block structure with

blocks of infinite size:

$$\mathbf{A}_0 = \begin{pmatrix} \mathbf{A}_{0,0} & \lambda_2\mathbf{I} & & \\ & \mathbf{A}_{0,0} & \lambda_2\mathbf{I} & \\ & & \mathbf{A}_{0,0} & \ddots \\ & & & \ddots \end{pmatrix}$$

where block $\mathbf{A}_{0,0}$ contains the transition rates within sublevel q_2, i.e.,

$$\mathbf{A}_{0,0} = \begin{pmatrix} -(\lambda_1+\lambda_2+\lambda_3+\mu_3) & \lambda_1 & & \\ & -(\lambda_1+\lambda_2+\lambda_3+\mu_3) & \lambda_1 & \\ & & -(\lambda_1+\lambda_2+\lambda_3+\mu_3) & \ddots \\ & & & \ddots \end{pmatrix}.$$

As before, the boundary block $\tilde{\mathbf{A}}_0$ contains the transition rates within level 0, i.e., the set of states with no high priority class-3 customers in the system:

$$\tilde{\mathbf{A}}_0 = \begin{pmatrix} \tilde{\tilde{\mathbf{A}}}_{0,0} & \lambda_2\mathbf{I} & & & \\ \mu_2\mathbf{I} & \tilde{\mathbf{A}}_{0,0} & \lambda_2\mathbf{I} & & \\ & \mu_2\mathbf{I} & \tilde{\mathbf{A}}_{0,0} & \lambda_2\mathbf{I} & \\ & & \mu_2\mathbf{I} & \tilde{\mathbf{A}}_{0,0} & \ddots \\ & & & \ddots & \ddots \end{pmatrix},$$

where

$$\tilde{\mathbf{A}}_{0,0} = \begin{pmatrix} -(\lambda_1+\lambda_2+\lambda_3+\mu_2) & \lambda_1 & & \\ & -(\lambda_1+\lambda_2+\lambda_3+\mu_2) & \lambda_1 & \\ & & -(\lambda_1+\lambda_2+\lambda_3+\mu_2) & \ddots \\ & & & \ddots \end{pmatrix},$$

and

$$\tilde{\tilde{\mathbf{A}}}_{0,0} = \begin{pmatrix} -(\lambda_1+\lambda_2+\lambda_3) & \lambda_1 & & \\ \mu_1 & -(\lambda_1+\lambda_2+\lambda_3+\mu_1) & \lambda_1 & \\ & \mu_1 & -(\lambda_1+\lambda_2+\lambda_3+\mu_1) & \ddots \\ & & \ddots & \end{pmatrix}.$$

The fundamental matrix $\mathbf{G}^{(3)}$ for the QBD with generator $\mathbb{Q}^{(3)}$ has the following upper

triangular block structure:

$$\mathbf{G}^{(3)} = \begin{pmatrix} \mathbf{G}_0^{(3)} & \mathbf{G}_1^{(3)} & \mathbf{G}_2^{(3)} & \ddots \\ & \mathbf{G}_0^{(3)} & \mathbf{G}_1^{(3)} & \ddots \\ & & \mathbf{G}_0^{(3)} & \ddots \\ & & & \ddots \end{pmatrix},$$

where the blocks $\mathbf{G}_{i_2}^{(3)}$ also have an upper triangular structure:

$$\mathbf{G}_{i_2}^{(3)} = \begin{pmatrix} g_{i_2,0}^{(3)} & g_{i_2,1}^{(3)} & g_{i_2,2}^{(3)} & \ddots \\ & g_{i_2,0}^{(3)} & g_{i_2,1}^{(3)} & \ddots \\ & & g_{i_2,0}^{(3)} & \ddots \\ & & & \ddots \end{pmatrix}, \qquad i_2 = 0, 1, 2, \ldots$$

Element $g_{i_2,i_1}^{(3)}$ is the probability that the number of class-2 and class-1 customers in the queue has increased by i_2 and i_1, respectively, at the first passage to level $q_3 - 1$ when starting in state (q_3, q_2, q_1). From this interpretation we can derive recursive equations for the elements $g_{i_2,i_1}^{(3)}$:

$$\mu_3 - \left(\sum_{l=1}^{3} \lambda_l + \mu_3\right) g_{0,0}^{(3)} + \lambda_3 (g_{0,0}^{(3)})^2 = 0, \quad i_2, i_1 = 0, \tag{14}$$

$$- \left(\sum_{l=1}^{3} \lambda_l + \mu_3\right) g_{i_2,i_1}^{(3)} + \lambda_2 g_{i_2-1,i_1}^{(3)} + \lambda_1 g_{i_2,i_1-1}^{(3)} \tag{15}$$

$$+\lambda_3 \sum_{j_2=0}^{i_2} \sum_{j_1=0}^{i_1} g_{j_2,j_1}^{(3)} g_{i_2-j_2,i_1-j_1}^{(3)} = 0, \quad i_2 + i_1 > 0,$$

where, by convention, $g_{i_2,i_1}^{(3)} = 0$ if $i_2 < 0$ or $i_1 < 0$. Just as for the 2-class system, we can use the fundamental matrix $\mathbf{G}^{(3)}$ to derive a relation between $\mathbf{p}_{q_3+1}$ and $\mathbf{p}_{q_3}$:

$$\mathbf{p}_{q_3+1} = \frac{\lambda_3}{\mu_3} \mathbf{p}_{q_3} \mathbf{G}^{(3)}, \quad q_3 \geq 0, \tag{16}$$

where $\mathbf{p}_{q_3}$ is the vector of steady-state probabilities at level q_3. The above equation can be written in scalar form as:

$$p_{q_3+1,q_2,q_1} = \frac{\lambda_3}{\mu_3} \sum_{i_2=0}^{q_2} \sum_{i_1=0}^{q_1} p_{q_3,q_2-i_2,q_1-i_1} \, g_{i_2,i_1}^{(3)}, \qquad q_3, q_2, q_1 \geq 0. \tag{17}$$

Hence the vectors $\mathbf{p}_1, \mathbf{p}_2, \ldots$ can be recursively computed, once $\mathbf{p}_0$ is known. The vector $\mathbf{p}_0$ satisfies the balance equations of the Markov process embedded on the (q_2, q_1)-plain with generator

$\mathbb{Q}^{(2)} = \tilde{\mathbf{A}}_0 + \lambda_3 \mathbf{G}^{(3)}$:

$$\mathbb{Q}^{(2)} = \begin{pmatrix} \tilde{\tilde{\mathbf{A}}}_{0,0} + \lambda_3 \mathbf{G}_0^{(3)} & \lambda_2 \mathbf{I} + \lambda_3 \mathbf{G}_1^{(3)} & \lambda_3 \mathbf{G}_2^{(3)} & \ddots \\ \mu_2 \mathbf{I} & \tilde{\mathbf{A}}_{0,0} + \lambda_3 \mathbf{G}_0^{(3)} & \lambda_2 \mathbf{I} + \lambda_3 \mathbf{G}_1^{(3)} & \ddots \\ & \mu_2 \mathbf{I} & \tilde{\mathbf{A}}_{0,0} + \lambda_3 \mathbf{G}_0^{(3)} & \ddots \\ & & \ddots & \ddots \end{pmatrix}.$$

Clearly the embedded Markov process is an $M/G/1$-type process with two priority classes: in states $(0, q_2, q_1)$ with $q_2 > 0$ only class-2 customers receive service. Hence, to determine $\mathbf{p}_0$, we can follow the same approach as for the 2-class system. First we have to find the fundamental matrix $\mathbf{G}^{(2)}$, which satisfies the following matrix equation (cf. [13]):

$$\mu_2 \mathbf{I} + \tilde{\mathbf{A}}_{0,0} \mathbf{G}^{(2)} + \lambda_2 \left(\mathbf{G}^{(2)} \right)^2 + \lambda_3 \sum_{m=1}^{\infty} \mathbf{G}_{m-1}^{(3)} \left(\mathbf{G}^{(2)} \right)^m = 0,$$

which can be written in scalar form as:

$$\mu_2 - \left(\sum_{l=1}^{3} \lambda_l + \mu_2 \right) g_0^{(2)} + \lambda_2 \left(g_0^{(2)} \right)^2 + \lambda_3 \sum_{m=1}^{\infty} g_{m-1,0}^{(3)} \left(g_0^{(2)} \right)^m = 0, \quad i_1 = 0, \tag{18}$$

$$- \left(\sum_{l=1}^{3} \lambda_l + \mu_2 \right) g_{i_1}^{(2)} + \lambda_1 g_{i_1-1}^{(2)} + \lambda_2 \sum_{j_1=0}^{i_1} g_{i_1-j_1}^{(2)} g_{j_1}^{(2)} \tag{19}$$

$$+ \lambda_3 \sum_{m=1}^{\infty} \sum_{\substack{j_0,\ldots,j_m \geq 0 \\ j_0+\cdots+j_m = i_1}} g_{m-1,j_0}^{(3)} \, g_{j_1}^{(2)} \cdots g_{j_m}^{(2)} = 0, \quad i_1 > 0.$$

The matrix $\mathbf{G}^{(2)}$ can now be used to express the probability vector $\mathbf{p}_{0,q_2+1}^{(2)} = (p_{0,q_2+1,0}, p_{0,q_2+1,1}, \ldots)$ in terms of $\mathbf{p}_{0,q_2}^{(2)}, \ldots, \mathbf{p}_{0,0}^{(2)}$ as follows (see [14]):

$$\mathbf{p}_{0,q_2+1}^{(2)} = \frac{1}{\mu_2} \left(\lambda_2 \mathbf{p}_{0,q_2}^{(2)} \mathbf{G}^{(2)} + \lambda_3 \sum_{i_2=0}^{q_2} \mathbf{p}_{0,q_2-i_2}^{(2)} \sum_{m=0}^{\infty} \mathbf{G}_{i_2+m+1}^{(3)} (\mathbf{G}^{(2)})^m \right), \tag{20}$$

which reads in scalar form as (cf. (17)):

$$p_{0,q_2+1,q_1} = \frac{1}{\mu_2} \left(\lambda_2 \sum_{i_1=0}^{q_1} p_{0,q_2,q_1-i_1} \, g_{i_1}^{(2)} \right. \tag{21}$$

$$\left. + \lambda_3 \sum_{i_2=0}^{q_2} \sum_{i_1=0}^{q_1} p_{0,q_2-i_2,q_1-i_1} \sum_{m=0}^{\infty} \sum_{\substack{l_0,\ldots,l_m \geq 0 \\ l_0+\cdots+l_m = i_1}} g_{i_2+m+1,l_0}^{(3)} \, g_{l_1}^{(2)} \cdots \, g_{l_m}^{(2)} \right), \qquad q_2, q_1 \geq 0.$$

Thus we can recursively compute $\mathbf{p}_{0,1}^{(2)}, \mathbf{p}_{0,2}^{(2)}, \ldots$ starting from $\mathbf{p}_{0,0}^{(2)}$. Finally the vector $\mathbf{p}_{0,0}^{(2)}$ follows from the balance equations of the Markov process embedded on q_1-axis with generator $\mathbb{Q}^{(1)}$:

$$\mathbb{Q}^{(1)} = \tilde{\tilde{\mathbf{A}}}_{0,0} + \lambda_2 \mathbf{G}^{(2)} + \lambda_3 \sum_{m=0}^{\infty} \mathbf{G}_m^{(3)} \left(\mathbf{G}^{(2)} \right)^m.$$

Note that $\mathbb{Q}^{(1)}$ is an $M/G/1$-type generator, with the following components:

$$\left[\mathbb{Q}^{(1)}\right]_{i,j\geq 0} = \begin{cases} \mu_1, & i-1=j>0 \\ -(\lambda_1+\lambda_2+\lambda_3)+\lambda_2 g_0^{(2)}+\lambda_3\sum\limits_{l=0}^{\infty} g_{l,0}^{(3)}(g_0^{(2)})^l, & i=j=0 \\ -(\lambda_1+\lambda_2+\lambda_3+\mu_1)+\lambda_2 g_0^{(2)}+\lambda_3\sum\limits_{m=0}^{\infty} g_{m,0}^{(3)}(g_0^{(2)})^m, & i=j>0 \\ \lambda_1+\lambda_2 g_1^{(2)}+\lambda_3\sum\limits_{m=0}^{\infty}\sum\limits_{\substack{l_0,\ldots,l_m\geq 0\\ l_0+\cdots+l_m=1}} g_{m,l_0}^{(3)} g_{l_1}^{(2)}\cdots g_{l_m}^{(2)}, & i+1=j\geq 0 \\ \vdots & \\ \lambda_2 g_h^{(2)}+\lambda_3\sum\limits_{m=0}^{\infty}\sum\limits_{\substack{l_0,\ldots,l_m\geq 0\\ l_0+\cdots+l_m=h}} g_{m,l_0}^{(3)} g_{l_1}^{(2)}\cdots g_{l_m}^{(2)}, & i+h=j\geq 0,\ h>1 \end{cases}$$

Starting with $p_{0,0,0}=1-\rho$ the probabilities $p_{0,0,1}, p_{0,0,2},\ldots$ can be determined recursively from the balance equations (cf. (13)):

$$\begin{aligned} p_{0,0,q_1+1} &= \frac{1}{\mu_1}\left(\lambda_1 p_{0,0,q_1}+\lambda_2\sum_{i_1=0}^{q_1} p_{0,0,q_1-i_1}\left(\sum_{l=i_1+1}^{\infty} g_l^{(2)}\right)\right. \\ &\quad \left.+\lambda_3\sum_{i_1=0}^{q_1} p_{0,0,q_1-i_1}\sum_{m=0}^{\infty}\sum_{\substack{l_0,\ldots,l_m\\ l_0+\cdots+l_m=i_1+1}} g_{m,l_0}^{(3)} g_{l_1}^{(2)}\cdots g_{l_m}^{(2)}\right), \quad q_1\geq 0. \end{aligned} \tag{22}$$

Summarizing, to find the joint queue length distribution of the 3-class $M_3/M_3/1$ priority system we first have to determine the fundamental matrices $\mathbf{G}^{(3)}$ and $\mathbf{G}^{(2)}$ from (14,15) and (18,19) and then we can compute the state probabilities using (22), (21) and (17). Let us now show how this approach can be generalized to N-classes of customers.

3.3 Matrix-analytic approach for N-class systems

As demonstrated in the previous sections, the method to compute the steady state probabilities of the N-class preemptive priority system includes the following steps. First, for $n=N,\ldots,2$, we determine iteratively (from a matrix equation) the fundamental matrix $\mathbf{G}^{(n)}$ of the Markov process embedded on the $(q_n,\ldots,q_1)$-plane with generator $\mathbb{Q}^{(n)}$. Note that $\mathbb{Q}^{(N)}$ is the generator of the original Markov process. After the fundamental matrices $\mathbf{G}^{(n)}$ $(n=N,\ldots,2)$ have been determined, we can compute the steady state probabilities using a forward recursion on the number of dimensions. That is, starting from $p_{0,\ldots,0}=1-\rho$, we compute the probabilities $p_{0,\ldots,0,q_1+1}$, $q_1\geq 0$, then $p_{0,\ldots,0,q_2+1,q_1}$, $q_2,q_1\geq 0$, and so forth until all the probabilities have been computed. Clearly the main operations are:

- Computation of the generator $\mathbb{Q}^{(n)}$ and the fundamental matrix $\mathbf{G}^{(n)}$ of the Markov process process embedded on the $(q_n,\ldots,q_1)$-plane, for $n=N,\ldots,1$;

- Computation of the probabilities $p_{0,\ldots,0,q_n+1,q_{n-1},\ldots,q_1}$, for $n = 1, \ldots, N$.

In the remainder of this section we will describe these operations in more detail. Let us first explain the structure of the generator $\mathbb{Q}^{(n)}$.

Structure of the generator $\mathbb{Q}^{(n)}$

The Markov process embedded on the $(q_n, \ldots, q_1)$-plane is of the $M/G/1$-type, where level $q_n \geq 0$ is defined as the set of states with q_n class-n customers (and no higher priority customers) in the system. According to this partitioning the generator $\mathbb{Q}^{(n)}$ has an upper-triangular block structure with one subdiagonal under the main diagonal:

$$\mathbb{Q}^{(n)} = \begin{pmatrix} \tilde{\mathbf{B}}_0^{(n)} & \mathbf{B}_1^{(n)} & \mathbf{B}_2^{(n)} & \mathbf{B}_3^{(n)} & \ddots \\ \mathbf{B}_{-1}^{(n)} & \mathbf{B}_0^{(n)} & \mathbf{B}_1^{(n)} & \mathbf{B}_2^{(n)} & \ddots \\ & \mathbf{B}_{-1}^{(n)} & \mathbf{B}_0^{(n)} & \mathbf{B}_1^{(n)} & \ddots \\ & & \mathbf{B}_{-1}^{(n)} & \mathbf{B}_0^{(n)} & \ddots \\ & & & \ddots & \ddots \end{pmatrix}$$

Here the infinite blocks $\mathbf{B}_{i_n}^{(n)}$ contain the transition rates from level q_n to level $q_n + i_n$, where i_n runs from -1 (service completion) to ∞ (arrivals). So $\mathbf{B}_{-1}^{(n)} = \mu_n \mathbf{I}$. As long as $q_n > 0$ the number of low priority customers in the system can only increase. This implies that, if we further partition level q_n into sublevels, where sublevel q_{n-1} is the set of all states with q_{n-1} class-$(n-1)$ customers in the system, then block $\mathbf{B}_{i_n}^{(n)}$ has an upper-triangular structure:

$$\mathbf{B}_{i_n}^{(n)} = \begin{pmatrix} \mathbf{B}_{i_n,0}^{(n)} & \mathbf{B}_{i_n,1}^{(n)} & \mathbf{B}_{i_n,2}^{(n)} & \ddots \\ & \mathbf{B}_{i_n,0}^{(n)} & \mathbf{B}_{i_n,1}^{(n)} & \ddots \\ & & \mathbf{B}_{i_n,0}^{(n)} & \ddots \\ & & & \ddots \end{pmatrix},$$

where block $\mathbf{B}_{i_n,i_{n-1}}^{(n)}$ contains the transition rates from sublevel q_{n-1} at level q_n to sublevel $q_{n-1} + i_{n-1}$ at level $q_n + i_n$. In the same way we can further partition the sublevels, so that block $\mathbf{B}_{i_n,i_{n-1}}^{(n)}$ is of the form:

$$\mathbf{B}_{i_n,i_{n-1}}^{(n)} = \begin{pmatrix} \mathbf{B}_{i_n,i_{n-1},0}^{(n)} & \mathbf{B}_{i_n,i_{n-1},1}^{(n)} & \mathbf{B}_{i_n,i_{n-1},2}^{(n)} & \ddots \\ & \mathbf{B}_{i_n,i_{n-1},0}^{(n)} & \mathbf{B}_{i_n,i_{n-1},1}^{(n)} & \ddots \\ & & \mathbf{B}_{i_n,i_{n-1},0}^{(n)} & \ddots \\ & & & \ddots \end{pmatrix}.$$

When we continue to partition the sublevels, we finally arrive at the blocks $\mathbf{B}^{(n)}_{i_n,\ldots,i_2}$:

$$\mathbf{B}^{(n)}_{i_n,\ldots,i_2} = \begin{pmatrix} b^{(n)}_{i_n,\ldots,i_2,0} & b^{(n)}_{i_n,\ldots,i_2,1} & b^{(n)}_{i_n,\ldots,i_2,2} & \ddots \\ & b^{(n)}_{i_n,\ldots,i_2,0} & b^{(n)}_{i_n,\ldots,i_2,1} & \ddots \\ & & b^{(n)}_{i_n,\ldots,i_2,0} & \ddots \\ & & & \ddots \end{pmatrix}.$$

where the scalar $b^{(n)}_{i_n,\ldots,i_1}$ denotes the transition rate from state $(q_n,\ldots,q_1)$ to $(q_n+i_n,\ldots,q_1+i_1)$. All indices i_k should be nonnegative, except for i_n, which can be -1 (service completion). Also note that for the initial generator $\mathbb{Q}^{(N)}$ we have:

$$b^{(N)}_{i_N,\ldots,i_1} = \begin{cases} \mu_N, & i_N = -1,\ i_{N-1} = \cdots = i_1 = 0 \\ -\left(\sum_{j=1}^N \lambda_j + \mu_N\right) & i_N = \cdots = i_1 = 0 \\ \lambda_j & i_j = 1,\ i_k = 0, \text{ for all } k \neq j,\ j = 1,\ldots,N \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

The structure of $\tilde{\mathbf{B}}^{(n)}_0$ is slightly different from the structure of $\mathbf{B}^{(n)}_0$, since it includes transition rates corresponding to service completions of lower priority customers.

Computation of the fundamental matrix $\mathbf{G}^{(n)}$

The matrix $\mathbf{G}^{(n)}$ is the minimal non-negative solution of the matrix equation (cf. [13, 14]):

$$\mathbf{B}^{(n)}_{-1} + \mathbf{B}^{(n)}_0\mathbf{G}^{(n)} + \mathbf{B}^{(n)}_1(\mathbf{G}^{(n)})^2 + \cdots = 0. \tag{24}$$

Hence $\mathbf{G}^{(n)}$ inherits the nested structure of the matrices $\mathbf{B}^{(n)}_{i_n}$:

$$\mathbf{G}^{(n)} = \begin{pmatrix} \mathbf{G}^{(n)}_0 & \mathbf{G}^{(n)}_1 & \mathbf{G}^{(n)}_2 & \ddots \\ & \mathbf{G}^{(n)}_0 & \mathbf{G}^{(n)}_1 & \ddots \\ & & \mathbf{G}^{(n)}_0 & \ddots \\ & & & \ddots \end{pmatrix}, \text{ where } \mathbf{G}^{(n)}_{i_{n-1}} = \begin{pmatrix} \mathbf{G}^{(n)}_{i_{n-1},0} & \mathbf{G}^{(n)}_{i_{n-1},1} & \mathbf{G}^{(n)}_{i_{n-1},2} & \ddots \\ & \mathbf{G}^{(n)}_{i_{n-1},0} & \mathbf{G}^{(n)}_{i_{n-1},1} & \ddots \\ & & \mathbf{G}^{(n)}_{i_{n-1},0} & \ddots \\ & & & \ddots \end{pmatrix},$$

$$\ldots, \mathbf{G}^{(n)}_{i_{n-1},\ldots,i_2} = \begin{pmatrix} g^{(n)}_{i_{n-1},\ldots,i_2,0} & g^{(n)}_{i_{n-1},\ldots,i_2,1} & g^{(n)}_{i_{n-1},\ldots,i_2,2} & \ddots \\ & g^{(n)}_{i_{n-1},\ldots,i_2,0} & g^{(n)}_{i_{n-1},\ldots,i_2,1} & \ddots \\ & & g^{(n)}_{i_{n-1},\ldots,i_2,0} & \ddots \\ & & & \ddots \end{pmatrix}.$$

Let us introduce the vector-index $\mathbf{i}^{(n)} = (i_n,\ldots,i_1)$ to simplify notations; so $g^{(n)}_{\mathbf{i}^{(n-1)}} = g^{(n)}_{i_{n-1},\ldots,i_1}$. Element $g^{(n)}_{\mathbf{i}^{(n-1)}}$ is the probability that the lengths of the lower priority queues have increased by $i_{n-1},\ldots,i_1$ at the first passage to level q_n when starting in state $(q_n,\ldots,q_1)$. From (24) it

follows that the scalars $g^{(n)}_{\mathbf{i}^{(n-1)}}$ are the minimal non-negative solution to the following recursive equations (cf.(4,5)):

$$
\begin{aligned}
b^{(n)}_{-1,\mathbf{0}^{(n-1)}} + b^{(n)}_{0,\mathbf{0}^{(n-1)}}\, g^{(n)}_{\mathbf{0}^{(n-1)}} + b^{(n)}_{1,\mathbf{0}^{(n-1)}}\, (g^{(n)}_{\mathbf{0}^{(n-1)}})^2 + b^{(n)}_{2,\mathbf{0}^{(n-1)}}\, (g^{(n)}_{\mathbf{0}^{(n-1)}})^3 + \cdots &= 0 \qquad (25)\\
b^{(n)}_{-1,\mathbf{e}^{(n-1)}_j} + \left(b^{(n)}_{0,\mathbf{0}^{(n-1)}}\, g^{(n)}_{\mathbf{e}^{(n-1)}_j} + b^{(n)}_{0,\mathbf{e}^{(n-1)}_j}\, g^{(n)}_{\mathbf{0}^{(n-1)}} \right) + & \qquad (26)\\
+ \left(b^{(n)}_{1,\mathbf{e}^{(n-1)}_j}\, (g^{(n)}_{\mathbf{0}^{(n-1)}})^2 + 2 b^{(n)}_{1,\mathbf{0}^{(n-1)}}\, g^{(n)}_{\mathbf{0}^{(n-1)}}\, g^{(n)}_{\mathbf{e}^{(n-1)}_j} \right) + \cdots &= 0\\
&\vdots\\
b^{(n)}_{-1,\mathbf{i}^{(n-1)}} + \sum_{l=1}^{\infty} \sum_{\substack{\mathbf{j}_0^{(n-1)},\ldots,\mathbf{j}_l^{(n-1)} \geq 0,\ \text{s.t.} \\ \mathbf{j}_0^{(n-1)}+\cdots+\mathbf{j}_l^{(n-1)} = \mathbf{i}^{(n-1)}}} b^{(n)}_{l-1,\mathbf{j}_0^{(n-1)}}\, g^{(n)}_{\mathbf{j}_1^{(n-1)}} \cdots g^{(n)}_{\mathbf{j}_l^{(n-1)}} &= 0. \qquad (27)
\end{aligned}
$$

Note that the elements of $\mathbf{G}^{(N)}$ can be written in a slightly simpler form due to (23):

$$
\begin{aligned}
\mu_N - \left(\sum_{j=1}^{N} \lambda_j + \mu_N \right) g^{(N)}_{\mathbf{0}^{(N-1)}} + \lambda_N (g^{(N)}_{\mathbf{0}^{(N-1)}})^2 &= 0,\\
- \left(\sum_{j=1}^{N} \lambda_j + \mu_N \right) g^{(N)}_{\mathbf{e}^{(N-1)}_j} + \lambda_{N-1} g^{(N)}_{\mathbf{0}^{(N-1)}} + 2\lambda_N g^{(N)}_{\mathbf{0}^{(N-1)}} g^{(N)}_{\mathbf{e}^{(N-1)}_j} &= 0,\\
&\vdots\\
- \left(\sum_{j=1}^{N} \lambda_j + \mu_N \right) g^{(N)}_{\mathbf{i}^{(N-1)}} + \sum_{j=1}^{N-1} \lambda_j g^{(N)}_{\mathbf{i}^{(N-1)}-\mathbf{e}^{(N-1)}_j} + \lambda_N \sum_{\mathbf{j}^{(N-1)}=\mathbf{0}^{(N-1)}}^{\mathbf{i}^{(N-1)}} g^{(N)}_{\mathbf{j}^{(N-1)}} g^{(N)}_{\mathbf{i}^{(N-1)}-\mathbf{j}^{(N-1)}} &= 0.
\end{aligned}
$$

Computation of the generator $\mathbb{Q}^{(n-1)}$

Once the fundamental matrix $\mathbf{G}^{(n)}$ is known, the generator $\mathbb{Q}^{(n-1)}$ of the Markov process embedded on the $(q_{n-1},\ldots,q_1)$-plane can be computed as:

$$
\mathbb{Q}^{(n-1)} = \tilde{\mathbf{B}}^{(n)}_0 + \mathbf{B}^{(n)}_1 \mathbf{G}^{(n)} + \mathbf{B}^{(n)}_2 \left(\mathbf{G}^{(n)}\right)^2 + \mathbf{B}^{(n)}_3 \left(\mathbf{G}^{(n)}\right)^3 + \cdots .
$$

Hence $\mathbb{Q}^{(n-1)}$ has the same structure as $\mathbb{Q}^{(n)}$, and its components $b^{(n-1)}_{\mathbf{i}^{(n-1)}}$ follow from:

$$
b^{(n-1)}_{\mathbf{i}^{(n-1)}} = \tilde{b}^{(n)}_{0,\mathbf{i}^{(n-1)}} + \sum_{l=1}^{\infty} \sum_{\substack{\mathbf{j}_0^{(n-1)},\ldots,\mathbf{j}_l^{(n-1)} \geq 0,\ \text{s.t.} \\ \mathbf{j}_0^{(n-1)}+\cdots+\mathbf{j}_l^{(n-1)} = \mathbf{i}^{(n-1)}}} b^{(n)}_{l,\mathbf{j}_0^{(n-1)}} g^{(n)}_{\mathbf{j}_1^{(n-1)}} \cdots g^{(n)}_{\mathbf{j}_l^{(n-1)}} \qquad (28)
$$

Computation of the steady state probabilities

Given $\mathbb{Q}^{(n)}$ and $\mathbf{G}^{(n)}$ for $n = N,\ldots,1$, we can determine the steady state probabilities recursively. Let $\mathbf{p}^{(n)}_{q_n}$ denote the vector of steady state probabilities $p_{\mathbf{q}}$ with $\mathbf{q} = (0,\ldots,0,q_n,\ldots,q_1)$ and

fixed q_n. Then for $n = 0, \ldots, N$ we can compute the vectors $\mathbf{p}_{q_n}^{(n)}$ from the balance equations (cf. (16, 20, 22)):

$$\mathbf{p}_{q_n+1}^{(n)} = \frac{1}{\mu_n} \sum_{i_n=0}^{q_n} \left(\mathbf{p}_{q_n-i_n}^{(n)} \sum_{m=1}^{\infty} \mathbf{B}_{i_n+m}^{(n)} \left(\mathbf{G}^{(n)} \right)^m \right), \quad q_n \geq 0, \tag{29}$$

with initially $\mathbf{p}_0^{(0)} = 1 - \rho$.

4 Computational issues and numerical results

In Section 4.1 we present the algorithm and discuss some computational issues. After that, in Section 4.2 we present results of an application in spare parts logistics.

4.1 Algorithm and computational issues

The method works with the *infinite* matrices $\mathbb{Q}^{(n)}$ and $\mathbf{G}^{(n)}$. Hence, for numerical computations, we need to truncate both matrices, i.e., the elements $b_{i_n,\ldots,i_1}^{(n)}$ and $g_{i_n,\ldots,i_1}^{(n)}$ are set to zero if one of the indices i_j is greater than some threshold. Note that, if we set $b_{i_n,\ldots,i_1}^{(n)} = 0$ for $i_n > K$, then equation (24) for $\mathbf{G}^{(n)}$ immediately reduces to a polynomial matrix equation of finite degree K. The algorithm to compute the joint queue length distribution is now as follows:

Algorithm

1. Construct the matrix $\mathbb{Q}^{(N)}$ using (23);
2. For $n = N - 1$ downto 1 do
   (a) Compute the probabilities $g_{i_n,\ldots,i_1}^{(n+1)}$ from (25-27) for all $0 \leq i_n, \ldots, i_1 \leq K_n$, where the threshold K_n is determined (during runtime) such that:
   $$\sum_{0 \leq i_n,\ldots,i_1 \leq K_n} g_{i_n,\ldots,i_1}^{(n+1)} \geq 1 - \epsilon,$$
   where ϵ is a small positive number;
   (b) Compute the transition rates $b_{i_n,\ldots,i_1}^{(n)}$ from (28) for all $0 \leq i_n, \ldots, i_1 \leq K_n$;
3. Set $p_{0,\ldots,0} = 1 - \rho$ and $g^{(1)}$=1;
4. For $n = 1$ to N do
   (a) Compute the probabilities $p_{0,\ldots,0,q_n+1,q_{n-1},\ldots,q_1}$ from (29) by using (the previously computed) $b_{i_n,\ldots,i_1}^{(n)}$, $g_{i_n-1,\ldots,i_1}^{(n)}$ and the probabilities $p_{0,\ldots,0,q_{n-1},\ldots,q_1}$.

4.2 Application in spare parts logistics

Our interest in the joint queue length distribution arose from a spare parts supply problem for repairable parts sharing the same repair shop. Below we apply the matrix-analytic approach to a simple spare parts supply problem (see Figure 3) to demonstrate the influence of assigning repair priorities on the performance of the system.

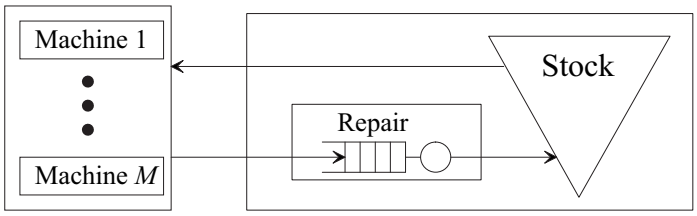

Figure 3: Example of a simple spare parts supply system

There are M machines and each machine contains three types of parts, subject to failure and replacement. A machine is working only when all three parts are working. To improve the reliability each machine is equipped with Z_i type i parts, so that, when one part fails, another one can immediately take over the necessary functions. Thus we have a system with redundancy. There are many examples of systems with this structure; typical ones can be found in [15].

When one of the parts fails, a service engineer takes a new part from a stock of parts and replaces the failed one. The failed part is then sent to a single-server repair facility. The repair time of a type i part is exponentially distributed with mean R_i; the delivery and replacement times are neglected. After repair the broken parts are assumed to be as good as new and they are put back to stock. The stock level of type i parts is denoted by S_i.

The system availability is defined as the average number of working machines:

$$Avail(S_1, S_2, S_3) = \frac{1}{M} \sum_{m=1}^{M} P(\text{Machine } m \text{ is working}).$$

The number of backorders of type i parts is given by $\max(0, q_i - S_i)$, where q_i is the number of type i parts in repair. Hence the probability that, for an arbitrary machine, all type i parts are broken is equal to

$$\frac{\max(0, q_i - S_i)}{Z_i M} \cdot \frac{\max(0, q_i - 1 - S_i)}{Z_i M - 1} \cdots \frac{\max(0, q_i - Z_i + 1 - S_i)}{Z_i M - Z_i + 1};$$

the first term in the above product is the probability that the first part is broken, the second one that the second part is broken *given* that the first one is broken, and so on. In terms of the joint queue length probabilities the system availability can now be written as:

$$Avail(S_1, S_2, S_3) = \sum_{q_3, q_2, q_1} \left[\prod_{i=1}^{3} \left(1 - \prod_{n=0}^{Z_i - 1} \frac{\max(0, q_i - n - S_i)}{Z_i M - n} \right) \right] p_{q_3, q_2, q_1}. \tag{30}$$

To compute the queue length probabilities p_{q_3,q_2,q_1} we assume that failures of type i parts occur according to a Poisson process with rate λ_i. This approximation, which is the only one needed, is valid when M, the total number of machines in the system, is large. Expression (30) much better estimates the system availability than other approximations proposed in the literature; e.g., the system availability defined in [15] only uses information on the mean number of backorders. But the matrix-analytic approach makes it possible to use the detailed distribution of the number of parts in repair. To demonstrate the approach we executed a set of experiments with the following parameters:

$$\begin{aligned} M &= 50, \\ Z_i &= 2,\ i = 1, 2, 3, \\ \lambda_i &= i/100, \\ R_i &= c/(3 - i + 1),\ \text{where } c \text{ is chosen such that } \sum_{i=1}^{3} \lambda_i R_i = \rho = 0.9 \text{ or } 0.95. \end{aligned}$$

In Table 1 we list the system availability according to (30) for different utilization rates of the repair shop and different priority assignments. The stock levels in these experiments depend on the mean queue lengths, i.e., we set $S_i = \lfloor E[q_i]/3 \rfloor$ for $i = 1, 2, 3$. The algorithm for the 3-class system was executed on a PC with a Pentium IV-2000 processor and 512MB operative memory. The computation times mentioned in Table 1 depend on the number of states with significant probability mass, i.e., on the load of the system and on the priority assignment.

Util.	Priorities			Queue length			Avail.	Comp.
ρ	r_1	r_2	r_3	$E[q_1]$	$E[q_2]$	$E[q_3]$		time
0.9	M	M	M	1.627	3.323	5.296	1.000	2.5 sec.
0.9	H	M	L	7.325	0.301	0.074	1.000	0.14 sec.
0.9	L	H	M	4.164	0.262	9.614	0.998	6.4 sec.
0.9	M	L	H	1.653	10.800	1.341	0.999	89.3 sec.
0.95	M	M	M	3.542	7.157	11.064	0.999	2.12 sec.
0.95	H	M	L	15.644	0.326	0.079	0.998	0.22 sec.
0.95	L	H	M	6.094	0.281	28.797	0.956	12.73 sec.
0.95	M	L	H	1.921	26.600	1.836	0.971	103.46 sec.

Table 1: System availability for different combinations of repair shop utilizations and priority assignments; the assignment M M M refers to the FCFS discipline.

From Table 1 it is clear repair priorities have a strong effect on the performance of the

system. The system availability is similar for all priority assignments, but the stock levels S_i are different. This implies that, in the presence of cheap and expensive parts, repair priorities may lead to substantial cost savings.

5 Extensions

The matrix-analytic approach can be easily extended to systems with other priority rules and service disciplines. Below we discuss the extension for multi-class, multi-server systems with preemptive and non-preemptive priorities.

$M_N/M_N/k$ system with preemptive priorities

Consider a priority system with k parallel and identical servers. As before, the states of this preemptive priority system can be described by N-dimensional vectors $\mathbf{q} = (q_N, \ldots, q_1)$. The balance equations for the steady state probabilities can be written as:

$$\begin{aligned} p_{\mathbf{q}} \left(\sum_{j=1}^{N} \lambda_j + \sum_{j=1}^{n} s_j \mu_j \right) &= \sum_{j=1}^{n} \lambda_j p_{\mathbf{q}-\mathbf{e}_j} + \sum_{j=1}^{N} (s_j + 1)\mu_j p_{\mathbf{q}+\mathbf{e}_j}, \quad q_N, \ldots, q_{n+1} = 0, \\ &\vdots \\ p_{\mathbf{0}} \left(\sum_{j=1}^{N} \lambda_j \right) &= \sum_{j=1}^{N} p_{\mathbf{e}_j} \mu_j, \end{aligned}$$

where s_j is defined as the number of class-j customers in service in state $\mathbf{q}$, so

$$s_j = \min \left(\max \left(k - \sum_{i=j+1}^{n} q_i \right), q_j \right).$$

Clearly, when $q_N > k$, all server are busy with class-N customers and the Markov process behaves the same as in the single server case (except that the service process is k times faster). In the single-server case we determined $\mathbf{p}_0^{(N)}$ by considering the Markov process embedded on level 0; now we have to determine $\mathbf{p}_0^{(N)}, \ldots, \mathbf{p}_{k-1}^{(N)}$ by embedding the Markov process on the levels $0, \ldots, k-1$. Thus the embedded Markov process gets an extra (but finite) dimension, and this repeats in the following steps of the algorithm. Essentially, the result is that, in the algorithm, the scalars $b_{\mathbf{i}^{(n)}}^{(n)}$ and $g_{\mathbf{i}^{(n-1)}}^{(n)}$ are replaced by finite dimensional matrices; i.e., the dimension is equal to $\binom{N-n+k}{k}$ (so in each step of the algorithm the dimension grows).

$M_N/M_N/k$ system with non-preemptive priorities

Now consider a single-server priority system with non-preemptive priorities, i.e., service of a low priority customer may not be interrupted by a higher priority customer. Then the system

can be described by the N-dimensional vector $\mathbf{q} = (q_N, \ldots, q_1)$ plus another N-dimensional vector $\mathbf{s} = (s_N, \ldots, s_1)$ that contains information about the customers in service, i.e., we have to work directly with matrices of finite dimension instead of the scalars $\mathbf{b}^{(n)}_{\mathbf{i}^{(n)}}$ and $\mathbf{g}^{(n)}_{\mathbf{i}^{(n-1)}}$. The dimension of these matrices is the same on all steps of the algorithm, namely $\binom{N-1+k}{k}$. The balance equations for the steady state probabilities can now be written as:

$$\begin{aligned} p_{\mathbf{q},\mathbf{s}} \left(\sum_{j=1}^{N} \lambda_j + \sum_{j=1}^{N} s_j \mu_j \right) &= \sum_{j=1}^{n} \lambda_j p_{\mathbf{q}-\mathbf{e}_j,\mathbf{s}} + \sum_{j=n}^{N} \sum_{i=1}^{N} (s_i+1)\mu_j p_{\mathbf{q}+\mathbf{e}_j,\mathbf{s}-\mathbf{e}_j+\mathbf{e}_i}, \quad q_N, \ldots, q_{n+1} = 0, \\ &\vdots \\ p_{\mathbf{0},\mathbf{0}} \left(\sum_{j=1}^{N} \lambda_j \right) &= \sum_{j=1}^{N} p_{\mathbf{0},\mathbf{e}_j} \mu_j. \end{aligned}$$

If we join the steady state probabilities with the same queue composition into vectors $\mathbf{p_q} = (p_{\mathbf{q},N}, \ldots, p_{\mathbf{q},1})$, then the above equations can be rewritten as:

$$\mathbf{p_q} \left(\sum_{j=1}^{N} \lambda_j + \mathbf{D} \right) = \sum_{j=1}^{N} \lambda_j \mathbf{p}_{\mathbf{q}-\mathbf{e}_j} + \sum_{j=n}^{N} p_{\mathbf{q}+\mathbf{e}_j} \mathbf{B}_j, \quad q_N, \ldots, q_{n+1} = 0,$$

where $\mathbf{D}$ and $\mathbf{B}_j$ are appropriately defined matrices. So, the algorithm in Section 3 can again be applied if we replace the scalars $b^{(n)}_{\mathbf{i}^{(n)}}$ and $g^{(n)}_{\mathbf{i}^{(n-1)}}$ by finite matrices $\mathbf{b}^{(n)}_{\mathbf{i}^{(n)}}$ and $\mathbf{g}^{(n)}_{\mathbf{i}^{(n-1)}}$, where the initial elements $\mathbf{b}^{(N)}_{\mathbf{i}^{(N)}}$ are computed as:

$$\mathbf{b}^{(N)}_{\mathbf{i}^{(N)}} = \begin{cases} \mathbf{B}_N, & i_N = -1,\ i_{N-1} = \cdots = i_2 = 0 \\ -\left(\sum_{j=1}^{N} \lambda_j + \mathbf{D} \right) & i_N = \cdots = i_1 = 0 \\ \lambda_j \mathbf{I} & i_j = 1,\ i_k = 0,\ \forall k \neq j,\ j = 1, \ldots, N \\ 0, & \text{otherwise} \end{cases}$$

Similarly, other priority systems can be analyzed, e.g., multi-server systems with multiple customer classes within each priority group (cf. [16, 17] for such systems with two priority groups).

References

[1] A. S. Alfa. Matrix-geometric solution of discrete time MAP/PH/1 priority queue. *Naval Res. Logist.*, 45(1):23–50, 1998.

[2] A. S. Alfa, B. Liu, and Q. M. He. Discrete-time analysis of $MAP/PH/1$ multiclass general preemptive priority queue. *Naval Res. Logist.*, 50(6):662–682, 2003.

[3] A. Cobham. Priority assignment in waiting line problems. *Operations Research*, 2:70–76, 1954.

[4] R. H. Davis. Waiting-time distribution of a multi-server priority queueing system. *Operations Research*, 14:133–136, 1966.

[5] H. R. Gail, S. L. Hantler, and B. A. Taylor. On preemptive markovian queue with multiple servers and two priority classes. *Mathematics of Operations Research*, 17(2):365–391, 1992.

[6] H. R. Gail, S. L. Hantler, and B. A. Taylor. Analysis of a non-preemptive priority multiserver queue. *Advances in Applied Probability*, 20(4):852–879, 1998.

[7] D. Gross and C. M. Harris. *Fundamentals of Queueing Theory.* John Wiley & Sons: New York, 1974.

[8] K. P. S. Isotupa and D. A. Stanford. An infinite-phase quasi-birth-and-death model for the non-preemptive priority $M/PH/1$ queue. *Stochastic Models*, 18(3):387–424, 2002.

[9] N. Jaiswal. *Priority Queues.* Academic Press: New York, 1968.

[10] D. Miller. Computation of steady-state probabilities for $M/M/1$ priority queues. *Operations Research*, 29(5):945–958, 1981.

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

[12] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: an algorithmic approach.* John Hopkins University Press, 1981.

[13] M. F. Neuts. *Structured Stochastic Matrices of $M/G/1$ type and their applications.* Dekker: New York, 1989.

[14] V. Ramaswami. A stable recursion for the steady state vector in markov chains of $M/G/1$ type. *Commun. Statist. - stochastic models*, 4(1):183–188, 1988.

[15] C. C. Sherbrooke. *Optimal inventory modelling of systems: Multi-Echelon techniques.* Wiley Press: New York, 1992.

[16] A. Sleptchenko. Multi-class, multi-server queues with non-preemptive priorities. Eurandom report 2003-016, EURANDOM, Technical University of Eindhoven, The Netherlands, 2003.

[17] A. Sleptchenko, A. van Harten, and M. C. van der Heijden. Analyzing multi-class, multi-server queueing systems with preemptive priorities. BETA technical report WP-77, University of Twente, The Netherlands, 2002. Submitted for publication.

[18] D. Wagner. Analysis of a finite capacity multiserver model with nonpreemptive priorities and nonrenewal input. In *Matrix-analytic methods in stochastic models (Flint, MI)*, volume 183 of *Lecture Notes in Pure and Appl. Math.*, pages 67–86. Dekker, New York, 1997.

[19] D. Wagner. A finite capacity multi-server multi-queueing priority model with non-renewal input. *Ann. Oper. Res.*, 79:63–82, 1998.