Efficient Analysis of the Queue Length Moments of the MMAP/MAP/1 Preemptive Priority Queue

Gábor Horváth

Budapest University of Technology and Economics, Department of Telecommunications,
Magyar tudósok körútja 2, 1521 Budapest, Hungary

Abstract

The analysis of priority queues where both the arrival and the service processes are correlated does not have a long literature. Only a few results are known, that attack the problem with the matrix geometric machinery. Unfortunately, these results have some restrictions that limit their usability significantly, for example they require the calculation of infinite series of matrices and infinite summations, that can be implemented only by truncation.

The method presented in this paper calculates only the queue length moments, but without relying on infinite series of matrices and provides procedures to calculate the arising infinite sums accurately in an efficient way by means of linear equations, matrix-quadratic equations and a coupled matrix-quadratic equation.

The numerical examples demonstrate that the presented method is several orders of magnitudes faster than the existing ones. From the large number of queue length moments it is possible to obtain lower and upper bounds for the queue length distribution by using existing moment based distribution estimation results.

Keywords: preemptive priority queue, marked MAP, queue length moments, matrix geometric methods

1. Introduction

Queueing models are widely used for modeling the behavior of various systems in many application fields like logistics, manufacturing processes, computer and telecommunications systems. In particular, the success of queueing theory in the planning and dimensioning of telecommunication networks has a long history.

Email address: ghorvath@hit.bme.hu (Gábor Horváth)

This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

The problem of the presence of different kinds of jobs in the system requiring different treatment is often solved by job classification, i.e. the server can take the class of the jobs into consideration when serving them. Several multi-class queueing models have been developed in the past to model different kinds of service disciplines like strict priority service, weighted fair queueing, etc.

One way to model complex traffic behavior is to apply Markovian arrival process (MAPs), and its multi-class extension marked Markovian arrival process (MMAPs). MAPs are capable of describing correlated arrivals, and MMAPs are able to capture cross-correlations between the arrivals belonging to different classes as well.

It is not only the complex traffic characteristics that makes modeling telecommunication systems challenging. More and more complex scheduling algorithms are used in the devices to support the differentiation of traffic belonging to different applications and customers of different importance.

The development of analytical results turned out to be much slower than the development of the technology that requires these results. For example, the results available for MMAP driven multi-class queues are restricted to some simple service policy only, like FCFS, LCFS and priority service.

In this paper we will focus on the two-class single-server preemptive priority queue with infinite waiting room and MMAP arrivals. The discrete time version of this queue is solved by Alfa following a matrix geometric approach in [1] and [2]. The extension of these results with batch arrivals are presented in [3]. The non-preemptive priority case with general service time and no correlation between the classes is considered in [4] and [5], where generator functions are derived for the queue length and the waiting time distributions in Laplace transform domain.

In our former paper [6], an approximate analysis method is presented for queueing networks consisting of continuous time MMAP/MAP/1 preemptive priority queues. For the analysis of the continuous time queues, we basically followed the same steps given in [1] for the discrete time queues. The starting point of the present paper is where [6] stops: our goal here is to cure the weaknesses of the queue length analysis procedure in [6] (that are also the weaknesses of [1] in discrete time). Our aim is to eliminate all infinite summations, and to reduce the computational complexity of the solution of the matrix equations needed to analyze the system.

This paper is organized as follows. Section 2 introduces the model, gives an overview on how MMAP/MAP/1 preemptive priority queues have been analyzed in the former literature and identifies the related numerical difficulties. Section 3 presents the contribution of the paper, the efficient method for the queue length moments of the low priority class. A numerical example is discussed in Section 4 to demonstrate the behavior of the procedure. Finally, Section 5 concludes the paper.

2. The MMAP/MAP/1 preemptive priority queue as a QBD process

The arrival process of the MMAP/MAP/1 preemptive priority queue is a marked MAP [7, 8]. Similar to MAPs, MMAPs have a background continuous time Markov chain (CTMC) modulating the arrivals. The transition rates of the background CTMC accompanied by an arrival of a high (low) priority customer are given by matrix $\boldsymbol{D_H}$ ($\boldsymbol{D_L}$), rates of the transitions without arrivals are given by matrix $\boldsymbol{D_0}$. Thus, the generator of the background CTMC of the MMAP is $\boldsymbol{D} = \boldsymbol{D_0} + \boldsymbol{D_H} + \boldsymbol{D_L}$. The size of the MMAP is n. Let τ denote the unique solution of $\tau\boldsymbol{D} = 0,\ \tau\mathbb{1} = 1$. The arrival rate of high (low) priority customers is given by $\lambda^{(H)} = \tau\boldsymbol{D_H}\mathbb{1}$ ($\lambda^{(L)} = \tau\boldsymbol{D_L}\mathbb{1}$).

The matrices of the MAP describing the service process of the high (low) priority class are $\boldsymbol{S_0^{(H)}}, \boldsymbol{S_1^{(H)}}$ ($\boldsymbol{S_0^{(L)}}, \boldsymbol{S_1^{(L)}}$). The size of the MAPs corresponding to the high and the low priority class are denoted by m and ℓ. The stationary distribution of the background process of the MAP generating the service times is denoted by $\omega^{(H)}$, it satisfies $\omega^{(H)}(\boldsymbol{S_0^{(H)}} + \boldsymbol{S_1^{(H)}}) = 0,\ \omega^{(H)}\mathbb{1} = 1$. The mean service rate of the high priority class is $\mu^{(H)} = \omega^{(H)}\boldsymbol{S_1^{(H)}}\mathbb{1}$. The mean service rate of the low priority customers $\mu^{(L)}$ (ignoring the presence of the high priority class) is obtained similarly.

According to the preemptive priority service policy, the ongoing service of a low priority customer is interrupted when a high priority customer arrives. The interrupted service continues in the phase where it has been interrupted as soon as the high priority queue becomes idle. In our model both queues are infinite. In the rest of the paper we assume that the stability condition holds; thus

$$\lambda^{(H)}/\mu^{(H)} + \lambda^{(L)}/\mu^{(L)} < 1. \tag{1}$$

2.1. The QBD process modeling the high priority queue

The queue length of the MMAP/MAP/1 preemptive priority queue can be modeled by a four dimensional CTMC: the first dimension represents the length of the high priority queue, the second one the length of the low priority queue, while the third and the fourth dimension keeps track of the phases of the arrival and service processes, respectively. The arrival and service processes together are referred to as background process in the sequel.

With proper numbering of the states, the generator matrix of the four-dimensional CTMC ($\boldsymbol{Q}$) exhibits the following block-tri-diagonal (QBD) structure:

$$\boldsymbol{Q} = \begin{bmatrix} \bar{\boldsymbol{A}}_{\boldsymbol{0}} & \boldsymbol{A}_+ & & & \\ \boldsymbol{A}_- & \boldsymbol{A}_{\boldsymbol{0}} & \boldsymbol{A}_+ & & \\ & \boldsymbol{A}_- & \boldsymbol{A}_{\boldsymbol{0}} & \boldsymbol{A}_+ & \\ & & \ddots & \ddots & \ddots \end{bmatrix} \tag{2}$$

The levels of the QBD correspond to the length of the high priority queue, while the phase of the QBD represents the length of the low priority queue and the phase of the arrival and service processes. Thus, the blocks of the generator matrix are infinite matrices as well, defined as follows:

$$\boldsymbol{A}_{+} = \mathrm{diag}\langle \boldsymbol{F_H} \rangle, \tag{3}$$

$$\boldsymbol{A}_{-} = \mathrm{diag}\langle \boldsymbol{B_1^{(H)}} \rangle, \tag{4}$$

$$\boldsymbol{A_0} = \begin{bmatrix} \boldsymbol{F_0} + \boldsymbol{B_0^{(H)}} & \boldsymbol{F_L} & & & \\ & \boldsymbol{F_0} + \boldsymbol{B_0^{(H)}} & \boldsymbol{F_L} & & \\ & & \boldsymbol{F_0} + \boldsymbol{B_0^{(H)}} & \boldsymbol{F_L} & \\ & & & \ddots & \ddots \end{bmatrix}, \tag{5}$$

$$\bar{\boldsymbol{A}}_{\boldsymbol{0}} = \begin{bmatrix} \boldsymbol{F_0} & \boldsymbol{F_L} & & & \\ \boldsymbol{B_1^{(L)}} & \boldsymbol{F_0} + \boldsymbol{B_0^{(L)}} & \boldsymbol{F_L} & & \\ & \boldsymbol{B_1^{(L)}} & \boldsymbol{F_0} + \boldsymbol{B_0^{(L)}} & \boldsymbol{F_L} & \\ & & & \ddots & \ddots \end{bmatrix} \tag{6}$$

where the matrices of size $n \times m \times \ell$ describing the transitions of the phase process are

$$\begin{aligned} \boldsymbol{F_0} = \boldsymbol{D_0} \otimes \boldsymbol{I}_m \otimes \boldsymbol{I}_\ell, \quad \boldsymbol{F_H} = \boldsymbol{D_H} \otimes \boldsymbol{I}_m \otimes \boldsymbol{I}_\ell, \quad \boldsymbol{F_L} = \boldsymbol{D_L} \otimes \boldsymbol{I}_m \otimes \boldsymbol{I}_\ell, \\ \boldsymbol{B_0^{(H)}} = \boldsymbol{I}_n \otimes \boldsymbol{S_0^{(H)}} \otimes \boldsymbol{I}_\ell, \quad \boldsymbol{B_1^{(H)}} = \boldsymbol{I}_n \otimes \boldsymbol{S_1^{(H)}} \otimes \boldsymbol{I}_\ell, \\ \boldsymbol{B_0^{(L)}} = \boldsymbol{I}_n \otimes \boldsymbol{I}_m \otimes \boldsymbol{S_0^{(L)}}, \quad \boldsymbol{B_1^{(L)}} = \boldsymbol{I}_n \otimes \boldsymbol{I}_m \otimes \boldsymbol{S_1^{(L)}}, \end{aligned} \tag{7}$$

where $\boldsymbol{I}_n$ denotes the identity matrix of size $n \times n$.

The stationary distribution of (2) is matrix-geometric [9],

$$\pi_k = \pi_0 \boldsymbol{R}^k, k \geq 0, \tag{8}$$

where π_k can be partitioned according to the length of the low priority queue: vector $\pi_{k,j}$ of size $n \cdot m \cdot \ell$ denotes the probability of having k jobs in the high, j jobs in the low priority queue and different phases of the background process. Entry (a, b) of matrix $\boldsymbol{R}$ is the expected time spent in phase b of the QBD at level n before the first visit to level $n-1$ starting in phase a at level $n-1$.

2.2. $\boldsymbol{R}$ and $\boldsymbol{G}$ matrices of the QBD

As the number of phases in this QBD is infinite, the corresponding fundamental matrices $\boldsymbol{R}$ and $\boldsymbol{G}$ are infinite as well. However, due to the special structure of the blocks of the generator, both matrix $\boldsymbol{R}$ and matrix $\boldsymbol{G}$ exhibit an upper-block-Toeplitz structure, since the number of low priority arrivals and the number of jobs in the system are independent given the phase of the arrival and service processes (shown in [1]). Thus we have:

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G_0^{(L)}} & \boldsymbol{G_1^{(L)}} & \boldsymbol{G_2^{(L)}} & \cdots \\ & \boldsymbol{G_0^{(L)}} & \boldsymbol{G_1^{(L)}} & \cdots \\ & & \boldsymbol{G_0^{(L)}} & \cdots \\ & & & \ddots \end{bmatrix}, \quad \boldsymbol{R} = \begin{bmatrix} \boldsymbol{R_0^{(L)}} & \boldsymbol{R_1^{(L)}} & \boldsymbol{R_2^{(L)}} & \cdots \\ & \boldsymbol{R_0^{(L)}} & \boldsymbol{R_1^{(L)}} & \cdots \\ & & \boldsymbol{R_0^{(L)}} & \cdots \\ & & & \ddots \end{bmatrix}. \tag{9}$$

These matrices are known to satisfy the following matrix quadratic equations:

$$\mathbf{0} = \boldsymbol{A}_{-} + \boldsymbol{A}_{\mathbf{0}}\boldsymbol{G} + \boldsymbol{A}_{+}\boldsymbol{G}^2. \tag{10}$$

$$\mathbf{0} = \boldsymbol{A}_{+} + \boldsymbol{R}\boldsymbol{A}_{\mathbf{0}} + \boldsymbol{R}^2\boldsymbol{A}_{-}, \tag{11}$$

By substituting (9) into (11) and (10), it is possible to derive recursive expressions for matrices $\boldsymbol{G}_{\boldsymbol{i}}^{\boldsymbol{(L)}}$ and $\boldsymbol{R}_{\boldsymbol{i}}^{\boldsymbol{(L)}}$ (see [1] and [6]).

The equations for matrices $\boldsymbol{G}_{\boldsymbol{i}}^{\boldsymbol{(L)}}$ are as follows:

$$\text{for } i = 0: \quad \mathbf{0} = \boldsymbol{B}_{\mathbf{1}}^{\boldsymbol{(H)}} + (\boldsymbol{F}_{\mathbf{0}} + \boldsymbol{B}_{\mathbf{0}}^{\boldsymbol{(H)}})\boldsymbol{G}_{\mathbf{0}}^{\boldsymbol{(L)}} + \boldsymbol{F}_{\boldsymbol{H}}{\boldsymbol{G}_{\mathbf{0}}^{\boldsymbol{(L)}}}^2, \tag{12}$$

$$\text{for } i > 0: \quad \mathbf{0} = \boldsymbol{F}_{\boldsymbol{L}}\boldsymbol{G}_{\boldsymbol{i-1}}^{\boldsymbol{(L)}} + (\boldsymbol{F}_{\mathbf{0}} + \boldsymbol{B}_{\mathbf{0}}^{\boldsymbol{(H)}})\boldsymbol{G}_{\boldsymbol{i}}^{\boldsymbol{(L)}} + \boldsymbol{F}_{\boldsymbol{H}}\sum_{k=0}^{i}\boldsymbol{G}_{\boldsymbol{k}}^{\boldsymbol{(L)}}\boldsymbol{G}_{\boldsymbol{i-k}}^{\boldsymbol{(L)}}. \tag{13}$$

Matrices $\boldsymbol{G}_{\boldsymbol{i}}^{\boldsymbol{(L)}}$ have important probabilistic interpretations. Entry (a, b) of matrix $\boldsymbol{G}_{\boldsymbol{i}}^{\boldsymbol{(L)}}$ is the conditional probability that starting from level n with the background process being in phase a, (1) the first visit to level $n-1$ occurs in phase b, (2) i low probability customers arrive during the first passage time.

The following expressions can be obtained for matrices $\boldsymbol{R}_{\boldsymbol{i}}^{\boldsymbol{(L)}}$:

$$\begin{aligned} &\text{for } i = 0: \quad \mathbf{0} = \boldsymbol{F}_{\boldsymbol{H}} + \boldsymbol{R}_{\mathbf{0}}^{\boldsymbol{(L)}}(\boldsymbol{F}_{\mathbf{0}} + \boldsymbol{B}_{\mathbf{0}}^{\boldsymbol{(H)}}) + {\boldsymbol{R}_{\mathbf{0}}^{\boldsymbol{(L)}}}^2\boldsymbol{B}_{\mathbf{1}}^{\boldsymbol{(H)}}, \\ &\text{for } i > 0: \quad \mathbf{0} = \boldsymbol{R}_{\boldsymbol{i-1}}^{\boldsymbol{(L)}}\boldsymbol{F}_{\boldsymbol{L}} + \boldsymbol{R}_{\boldsymbol{i}}^{\boldsymbol{(L)}}(\boldsymbol{F}_{\mathbf{0}} + \boldsymbol{B}_{\mathbf{0}}^{\boldsymbol{(H)}}) + \sum_{k=0}^{i}\boldsymbol{R}_{\boldsymbol{k}}^{\boldsymbol{(L)}}\boldsymbol{R}_{\boldsymbol{i-k}}^{\boldsymbol{(L)}}.\boldsymbol{B}_{\mathbf{1}}^{\boldsymbol{(H)}} \end{aligned} \tag{14}$$

The generating function of matrix series $\boldsymbol{G}_{\boldsymbol{i}}^{\boldsymbol{(L)}}$ defined by $\boldsymbol{G}^{\boldsymbol{(L)}}(z) = \sum_{k=0}^{\infty} z^k\boldsymbol{G}_{\boldsymbol{k}}^{\boldsymbol{(L)}}$ will be used several times in the sequel. From (13) and (12) we have that $\boldsymbol{G}^{\boldsymbol{(L)}}(z)$ is the solution of the following matrix-quadratic equation:

$$\mathbf{0} = \boldsymbol{B}_{\mathbf{1}}^{\boldsymbol{(H)}} + (\boldsymbol{F}_{\mathbf{0}} + z\boldsymbol{F}_{\boldsymbol{L}} + \boldsymbol{B}_{\mathbf{0}}^{\boldsymbol{(H)}})\boldsymbol{G}^{\boldsymbol{(L)}}(z) + \boldsymbol{F}_{\boldsymbol{H}}{\boldsymbol{G}^{\boldsymbol{(L)}}(z)}^2. \tag{15}$$

Similarly, the generating function of matrix series $\boldsymbol{R}_{\boldsymbol{i}}^{\boldsymbol{(L)}}$ defined by $\boldsymbol{R}^{\boldsymbol{(L)}}(z) = \sum_{k=0}^{\infty} z^k\boldsymbol{R}_{\boldsymbol{k}}^{\boldsymbol{(L)}}$ is the solution of the following matrix-quadratic equation:

$$\mathbf{0} = \boldsymbol{F}_{\boldsymbol{H}} + \boldsymbol{R}^{\boldsymbol{(L)}}(z)(\boldsymbol{F}_{\mathbf{0}} + z\boldsymbol{F}_{\boldsymbol{L}} + \boldsymbol{B}_{\mathbf{0}}^{\boldsymbol{(H)}}) + {\boldsymbol{R}^{\boldsymbol{(L)}}(z)}^2\boldsymbol{B}_{\mathbf{1}}^{\boldsymbol{(H)}}. \tag{16}$$

2.3. Analysis of the zero level

Having shown the structure of matrix $\boldsymbol{R}$, the second ingredient necessary to obtain the stationary distribution (8) is vector π_0. Relations for vector π_0 can be derived from the boundary equations of $\pi\boldsymbol{Q} = 0$ as:

$$\pi_0\bar{\boldsymbol{A}}_{\mathbf{0}} + \pi_0\boldsymbol{R}\boldsymbol{A}_{-} = \mathbf{0}, \qquad \pi_{0,0}\mathbb{1} = 1 - \lambda^{(H)}/\mu^{(H)} - \lambda^{(L)}/\mu^{(L)}, \tag{17}$$

where $\mathbb{1}$ denotes the column vector of ones.

Due to the structure of matrices $\bar{\boldsymbol{A}}_{\boldsymbol{0}}$, $\boldsymbol{A}_{-}$ and $\boldsymbol{R}$, (17) is equivalent to the solution of an M/G/1 type CTMC (see [3] and [6]). However, by using the relation $\boldsymbol{R}\boldsymbol{A}_{-} = \boldsymbol{A}_{+}\boldsymbol{G}$ (see e.g. [9], page 144) we can re-formulate (17) and the corresponding M/G/1 type CTMC to a more appropriate form.

The equations for π_0 by using $\boldsymbol{G}$ instead of $\boldsymbol{R}$ are then:

$$\pi_0 \bar{\boldsymbol{A}}_{\boldsymbol{0}} + \pi_0 \boldsymbol{A}_{+}\boldsymbol{G} = \boldsymbol{0}, \qquad \pi_{0,0}\mathbb{1} = 1 - \lambda^{(H)}/\mu^{(H)} - \lambda^{(L)}/\mu^{(L)}, \tag{18}$$

and the generator of the related M/G/1 type CTMC is as follows:

$$\boldsymbol{Q}_{\boldsymbol{0}} = \bar{\boldsymbol{A}}_{\boldsymbol{0}} + \boldsymbol{A}_{+}\boldsymbol{G} =$$
$$\begin{bmatrix} \boldsymbol{F}_{\boldsymbol{0}} + \boldsymbol{F}_{\boldsymbol{H}}\boldsymbol{G}_{\boldsymbol{0}}^{\boldsymbol{(L)}} & \boldsymbol{F}_{\boldsymbol{L}} + \boldsymbol{F}_{\boldsymbol{H}}\boldsymbol{G}_{\boldsymbol{1}}^{\boldsymbol{(L)}} & \boldsymbol{F}_{\boldsymbol{H}}\boldsymbol{G}_{\boldsymbol{2}}^{\boldsymbol{(L)}} & \boldsymbol{F}_{\boldsymbol{H}}\boldsymbol{G}_{\boldsymbol{3}}^{\boldsymbol{(L)}} & \ldots \\ \boldsymbol{B}_{\boldsymbol{1}}^{\boldsymbol{(L)}} & \boldsymbol{F}_{\boldsymbol{0}} + \boldsymbol{B}_{\boldsymbol{0}}^{\boldsymbol{(L)}} + \boldsymbol{F}_{\boldsymbol{H}}\boldsymbol{G}_{\boldsymbol{0}}^{\boldsymbol{(L)}} & \boldsymbol{F}_{\boldsymbol{L}} + \boldsymbol{F}_{\boldsymbol{H}}\boldsymbol{G}_{\boldsymbol{1}}^{\boldsymbol{(L)}} & \boldsymbol{F}_{\boldsymbol{H}}\boldsymbol{G}_{\boldsymbol{2}}^{\boldsymbol{(L)}} & \ldots \\ & \boldsymbol{B}_{\boldsymbol{1}}^{\boldsymbol{(L)}} & \boldsymbol{F}_{\boldsymbol{0}} + \boldsymbol{B}_{\boldsymbol{0}}^{\boldsymbol{(L)}} + \boldsymbol{F}_{\boldsymbol{H}}\boldsymbol{G}_{\boldsymbol{0}}^{\boldsymbol{(L)}} & \boldsymbol{F}_{\boldsymbol{L}} + \boldsymbol{F}_{\boldsymbol{H}}\boldsymbol{G}_{\boldsymbol{1}}^{\boldsymbol{(L)}} & \ldots \\ & \ddots & \ddots & \ddots & \ldots \end{bmatrix} \tag{19}$$

The stationary probability vectors $\pi_{0,i}$ can be calculated recursively using the Ramaswami formula [10]. Tailoring it to this particular system gives:

$$\pi_{0,i} = \left(\sum_{k=0}^{i-1} \pi_{0,k} \boldsymbol{T}_{\boldsymbol{i-k}} \right) (-\boldsymbol{T}_{\boldsymbol{0}})^{-1}, \qquad i \geq 1, \tag{20}$$

where matrices $\boldsymbol{T}_{\boldsymbol{i}}$ are defined by

$$\begin{aligned} \boldsymbol{T}_{\boldsymbol{i}} &= \sum_{k=i}^{\infty} \boldsymbol{F}_{\boldsymbol{H}}\boldsymbol{G}_{\boldsymbol{k}}^{\boldsymbol{(L)}} {\boldsymbol{G}_{\boldsymbol{H}_{\boldsymbol{0}}}}^{k-i}, \qquad i \geq 2, \\ \boldsymbol{T}_{\boldsymbol{1}} &= \boldsymbol{F}_{\boldsymbol{L}} + \sum_{k=1}^{\infty} \boldsymbol{F}_{\boldsymbol{H}}\boldsymbol{G}_{\boldsymbol{k}}^{\boldsymbol{(L)}} {\boldsymbol{G}_{\boldsymbol{H}_{\boldsymbol{0}}}}^{k-1}, \\ \boldsymbol{T}_{\boldsymbol{0}} &= \boldsymbol{F}_{\boldsymbol{0}} + \boldsymbol{B}_{\boldsymbol{0}}^{\boldsymbol{(L)}} + \boldsymbol{F}_{\boldsymbol{H}}\boldsymbol{G}_{\boldsymbol{0}}^{\boldsymbol{(L)}} + \boldsymbol{T}_{\boldsymbol{1}}\boldsymbol{G}_{\boldsymbol{H}_{\boldsymbol{0}}}, \end{aligned} \tag{21}$$

and, vector $\pi_{0,0}$ is the solution of the following set of linear equations:

$$\begin{aligned} &\pi_{0,0} \left(\boldsymbol{F}_{\boldsymbol{0}} + \boldsymbol{F}_{\boldsymbol{H}}\boldsymbol{G}_{\boldsymbol{0}}^{\boldsymbol{(L)}} + \boldsymbol{T}_{\boldsymbol{1}}(-\boldsymbol{T}_{\boldsymbol{0}})^{-1}\boldsymbol{B}_{\boldsymbol{1}}^{\boldsymbol{(L)}} \right) = 0, \\ &\pi_{0,0}\mathbb{1} = 1 - \lambda^{(H)}/\mu^{(H)} - \lambda^{(L)}/\mu^{(L)}. \end{aligned} \tag{22}$$

In order to compute π_0, we need to obtain matrix $\boldsymbol{G}_{\boldsymbol{H}_{\boldsymbol{0}}}$. Matrix $\boldsymbol{G}_{\boldsymbol{H}_{\boldsymbol{0}}}$ is the minimal non-negative solution of the following matrix equation ([11]):

$$\boldsymbol{0} = \boldsymbol{B}_{\boldsymbol{1}}^{\boldsymbol{(L)}} + (\boldsymbol{F}_{\boldsymbol{0}} + \boldsymbol{B}_{\boldsymbol{0}}^{\boldsymbol{(L)}})\boldsymbol{G}_{\boldsymbol{H}_{\boldsymbol{0}}} + \boldsymbol{F}_{\boldsymbol{L}}{\boldsymbol{G}_{\boldsymbol{H}_{\boldsymbol{0}}}}^2 + \boldsymbol{F}_{\boldsymbol{H}} \sum_{i=0}^{\infty} \boldsymbol{G}_{\boldsymbol{i}}^{\boldsymbol{(L)}} {\boldsymbol{G}_{\boldsymbol{H}_{\boldsymbol{0}}}}^{i} \cdot \boldsymbol{G}_{\boldsymbol{H}_{\boldsymbol{0}}}. \tag{23}$$

2.4. Performance measures

Having matrix $\boldsymbol{R}$ and vector π_0 determined, the steady state probabilities of the QBD are given by (8). Utilizing the block structure of $\boldsymbol{R}$ (9), the steady state joint probability vectors of the lengths of the high and the low priority queues can be expressed by the following recursive formula:

$$\pi_{i,j} = \sum_{k=0}^{j} \pi_{i-1,k} \boldsymbol{R}^{(\boldsymbol{L})}_{\boldsymbol{j-k}}. \tag{24}$$

The marginal steady-state probability vectors of the classes are calculated by the appropriate summation of (24). For the high priority class we have

$$\pi_i^{(H)} = \sum_{j=0}^{\infty} \pi_{i,j} = \sum_{j=0}^{\infty}\sum_{k=0}^{j} \pi_{i-1,k} \boldsymbol{R}^{(\boldsymbol{L})}_{\boldsymbol{j-k}} = \underbrace{\sum_{k=0}^{\infty} \pi_{i-1,k}}_{\pi_{i-1}^{(H)}} \sum_{j=0}^{\infty} \boldsymbol{R}^{(\boldsymbol{L})}_{\boldsymbol{j}}. \tag{25}$$

Note that the queue length distribution of the high priority class is matrix-geometric. Furthermore, $\sum_{j=0}^{\infty} \boldsymbol{R}^{(\boldsymbol{L})}_{\boldsymbol{j}}$ can be obtained efficiently as the minimal non-negative solution of a matrix quadratic equation (see [6] or Section 3.2), so the corresponding performance measures can be obtained efficiently. Actually, the high priority queue can even be analyzed in isolation, since the service policy is preemptive.

For the queue length distribution of the low priority class we have

$$\begin{aligned}\pi_j^{(L)} &= \sum_{i=0}^{\infty} \pi_{i,j} = \pi_{0,j} + \sum_{i=1}^{\infty}\sum_{k=0}^{j} \pi_{i-1,k} \boldsymbol{R}^{(\boldsymbol{L})}_{\boldsymbol{j-k}} \\ &= \pi_{0,j} + \sum_{k=0}^{j} \left(\sum_{i=1}^{\infty} \pi_{i-1,k} \right) \boldsymbol{R}^{(\boldsymbol{L})}_{\boldsymbol{j-k}} = \pi_{0,j} + \sum_{k=0}^{j} \pi_k^{(L)} \boldsymbol{R}^{(\boldsymbol{L})}_{\boldsymbol{j-k}}.\end{aligned} \tag{26}$$

For the low priority class the calculation of the performance measures is more difficult. For the first term of (26) an M/G/1 type system has to be analyzed, which dominates the complexity of the solution. Since the analysis of the high priority queue is straightforward, we will exclusively focus on the low priority class in the rest of the paper.

2.5. Methods and challenges to compute the queue length moments

In [1] the stationary distribution of the number of customers, the mean queue lengths and the waiting time distributions are considered, but even for the mean queue length of the low priority class the inverse of a matrix of infinite dimension $\boldsymbol{I} - \boldsymbol{R}$ is calculated. It is noted both in [1] (page 45) and [2] (page 675) that they used a truncation at an appropriate point to calculate performance measures. [6] uses a more advanced procedure by expressing the generating functions of vectors $\pi_j^{(L)}$ and matrices $\boldsymbol{T_i}$, and deriving the queue length moments by taking the derivatives of these generating functions.

To obtain even the mean queue length of the low priority queue, both procedures suffer from the following drawbacks:

- the calculation of the infinite series of $\boldsymbol{R}_{\boldsymbol{i}}^{\boldsymbol{(L)}}$ and/or $\boldsymbol{G}_{\boldsymbol{i}}^{\boldsymbol{(L)}}$ that requires a considerable computational effort and storage,
- the computation of the $\boldsymbol{G}_{\boldsymbol{H_0}}$ matrix of an M/G/1 type Markov chain,
- to obtain $\pi_{0,0}$ an infinite sum is computed,
- in [1] the inversion of an infinite matrix is required to calculate the mean queue length,
- in [6] an infinite summation over a quarter-plane is required to calculate the mean queue length.

In the rest of the paper we propose a solution to these problems.

3. Efficient queue length analysis of the MMAP/MAP/1 preemptive priority queue

The proposed analysis method is based on the following observations, making it more efficient than former methods.

- The M/G/1 type Markov chain (19) describing the system at level zero is a special one, since its matrix blocks are defined recursively according to (12) and (13). This special structure makes it possible to develop solution methods that are more efficient than general M/G/1 type Markov chain solvers. Namely,
  - the commutativity of two involved matrices allows calculating the invariant matrix of the M/G/1 type Markov chain without generating and storing matrix series $\boldsymbol{G}_{\boldsymbol{i}}^{\boldsymbol{(L)}}$;
  - the ETAQA approach [12] allows us to obtain the queue length moments via systems of linear equations. The way the blocks of our M/G/1 type system are defined makes it possible to derive closed-form coefficient matrices for these systems of linear equations.
- Furthermore, we show that the moment-like summations of $\boldsymbol{G}_{\boldsymbol{i}}^{\boldsymbol{(L)}}$ and $\boldsymbol{R}_{\boldsymbol{i}}^{\boldsymbol{(L)}}$ matrices, required to calculate the queue length moments, can be obtained by solving matrix-quadratic equations and systems of linear equations.

These key elements are detailed in the subsequent subsections.

3.1. Two fundamental matrices and their relations

There are two matrices that play key roles in the efficient analysis of the system. One of them is $\boldsymbol{G_{H_0}}$, that is the solution of (23), and the other one is matrix $\boldsymbol{S}$, defined by

$$\boldsymbol{S} = \sum_{i=0}^{\infty} \boldsymbol{G_i^{(L)}} \boldsymbol{G_{H_0}}^i. \tag{27}$$

Theorem 1. *If the algebraic multiplicities of the eigenvalues of matrices $\boldsymbol{G_{H_0}}$ and $\boldsymbol{S}$ are one, $\boldsymbol{G_{H_0}}\boldsymbol{S} = \boldsymbol{S}\boldsymbol{G_{H_0}}$ holds.*

To prove this theorem we first need the following Lemma:

Lemma 1. *If the algebraic multiplicities of the eigenvalues of matrix $\boldsymbol{G^{(L)}}(z)$ and matrix $\boldsymbol{F_0} + z\boldsymbol{F_L} + \boldsymbol{F_H}\boldsymbol{G^{(L)}}(z)$ are one, the eigenvectors of $\boldsymbol{G^{(L)}}(z)$ and $\boldsymbol{F_0} + z\boldsymbol{F_L} + \boldsymbol{F_H}\boldsymbol{G^{(L)}}(z)$ are the same.*

Proof. The proof uses the same techniques as in [13].

Let ν_i and u_i be the eigenvalue and the corresponding right eigenvector of $\boldsymbol{G^{(L)}}(z)$ (for simplicity we assume that $\boldsymbol{G^{(L)}}(z)$ has distinct eigenvalues). As $\boldsymbol{G^{(L)}}(z)$ satisfies the matrix-quadratic equation of (15), ν_i satisfies

$$\det\left[\boldsymbol{B_1^{(H)}} + (\boldsymbol{F_0} + z\boldsymbol{F_L} + \boldsymbol{B_0^{(H)}})\nu_i + \boldsymbol{F_H}\nu_i^2\right] = 0, \tag{28}$$

and the associated right eigenvector u_i is the solution of

$$\left[\boldsymbol{B_1^{(H)}} + (\boldsymbol{F_0} + z\boldsymbol{F_L} + \boldsymbol{B_0^{(H)}})\nu_i + \boldsymbol{F_H}\nu_i^2\right] \cdot u_i = 0. \tag{29}$$

Note that both ν_i and vectors u_i are functions of z.

By substituting (7) into (28) and by some basic manipulations we get

$$\det\left[\left(\underbrace{(\nu_i\boldsymbol{D_0} + \nu_i z\boldsymbol{D_L} + \nu_i^2\boldsymbol{D_H})}_{\boldsymbol{M_1}} \oplus \underbrace{(\boldsymbol{S_1^{(H)}} + \nu_i\boldsymbol{S_0^{(H)}})}_{\boldsymbol{M_2}}\right) \otimes \boldsymbol{I}_\ell\right] = 0, \tag{30}$$

from which it follows that $\boldsymbol{M_1} \oplus \boldsymbol{M_2}$ has a zero eigenvalue. Let δ_j,$j = 1,\dots,n$ and σ_k,$k = 1,\dots,m$ denote the eigenvalues of $\boldsymbol{M_1}$ and $\boldsymbol{M_2}$, respectively. Since the eigenvalues of $\boldsymbol{M_1} \oplus \boldsymbol{M_2}$ are $\delta_j + \sigma_k$, to have a zero eigenvalue there must exist j' and k' such that $\delta_{j'} = -\sigma_{k'}$. The eigenvector of $\boldsymbol{M_1}$ belonging to $\delta_{j'}$ is denoted by $\theta_{j'}$, the one of $\boldsymbol{M_2}$ belonging to $\sigma_{k'}$ is denoted by $\psi_{k'}$. Let us introduce $\phi_i = \theta_{j'} \otimes \psi_{k'} \otimes \mathbb{1}_\ell$.

Now we show that ϕ_i is an eigenvector of $\boldsymbol{G^{(L)}}(z)$ associated with ν_i, thus it satisfies (29):

$$\begin{aligned}
&\left[\boldsymbol{B_1^{(H)}} + (\boldsymbol{F_0} + z\boldsymbol{F_L} + \boldsymbol{B_0^{(H)}})\nu_i + \boldsymbol{F_H}\nu_i^2\right] \cdot \phi_i \\
&\quad = [\boldsymbol{I}_n \otimes \boldsymbol{M_2} \otimes \boldsymbol{I}_\ell + \boldsymbol{M_1} \otimes \boldsymbol{I}_m \otimes \boldsymbol{I}_\ell] \cdot (\theta_{j'} \otimes \psi_{k'} \otimes \mathbb{1}_\ell) \\
&\quad = \theta_{j'} \otimes (\sigma_{k'}\psi_{k'}) \otimes \mathbb{1}_\ell + (\delta_{j'}\theta_{j'}) \otimes \psi_{k'} \otimes \mathbb{1}_\ell = 0.
\end{aligned} \tag{31}$$

Next, we show that ϕ_i is an eigenvector of $\boldsymbol{F_0} + z\boldsymbol{F_L} + \boldsymbol{F_H}\boldsymbol{G^{(L)}}(z)$:

$$\begin{aligned}&\left[\boldsymbol{F_0} + z\boldsymbol{F_L} + \boldsymbol{F_H}\boldsymbol{G^{(L)}}(z)\right] \cdot \phi_i = \left[\boldsymbol{F_0} + z\boldsymbol{F_L} + \boldsymbol{F_H}\nu_i\right] \cdot \phi_i\\ &= \left[\boldsymbol{D_0} \otimes \boldsymbol{I}_m \otimes \boldsymbol{I}_\ell + z\boldsymbol{D_L} \otimes \boldsymbol{I}_m \otimes \boldsymbol{I}_\ell + \boldsymbol{D_H}\nu_i \otimes \boldsymbol{I}_m \otimes \boldsymbol{I}_\ell\right] \cdot (\theta_{j'} \otimes \psi_{k'} \otimes \mathbb{1}_\ell)\\ &= (\boldsymbol{M_1}/\nu_i \otimes \boldsymbol{I}_m \otimes \boldsymbol{I}_\ell) \cdot (\theta_{j'} \otimes \psi_{k'} \otimes \mathbb{1}_\ell) = \delta_{j'}/\nu_i \cdot \phi_i.\end{aligned} \tag{32}$$

□

Proof of Theorem 1. First observe that matrices $\boldsymbol{G_i^{(L)}}$ can be written as $\boldsymbol{G_i^{(L)}} = \boldsymbol{\tilde{G}_i^{(L)}} \otimes \boldsymbol{I}_\ell$ since the service process of the low priority class is stopped during the busy period of the high priority class.

The proof will be similar to the one of Lemma 1, using the same techniques as in [13] again.

Let λ_k and v_k be the eigenvalue and the corresponding right eigenvector of $\boldsymbol{G_{H_0}}$ (for simplicity we assume that $\boldsymbol{G_{H_0}}$ has distinct eigenvalues). As $\boldsymbol{G_{H_0}}$ satisfies the matrix equation of (23), λ_k satisfies

$$\det\left[\boldsymbol{B_1^{(L)}} + (\boldsymbol{F_0} + \boldsymbol{B_0^{(L)}})\lambda_k + \boldsymbol{F_L}\lambda_k^2 + \boldsymbol{F_H}\sum_{i=0}^{\infty}(\boldsymbol{\tilde{G}_i^{(L)}} \otimes \boldsymbol{I}_\ell)\lambda_k^i \cdot \lambda_k\right] = 0, \tag{33}$$

and the associated right eigenvector v_k is the solution of

$$\left[\boldsymbol{B_1^{(L)}} + (\boldsymbol{F_0} + \boldsymbol{B_0^{(L)}})\lambda_k + \boldsymbol{F_L}\lambda_k^2 + \boldsymbol{F_H}\sum_{i=0}^{\infty}(\boldsymbol{\tilde{G}_i^{(L)}} \otimes \boldsymbol{I}_\ell)\lambda_k^i \cdot \lambda_k\right] \cdot v_k = 0. \tag{34}$$

By substituting (7) into (33) and by some basic manipulation we get

$$\det\Big[\underbrace{\Big(\lambda_k\boldsymbol{D_0}\otimes\boldsymbol{I}_m + \lambda_k^2\boldsymbol{D_L}\otimes\boldsymbol{I}_m + \sum_{i=0}^{\infty}\lambda_k^{i+1}(\boldsymbol{D_H}\otimes\boldsymbol{I}_m)\boldsymbol{\tilde{G}_i^{(L)}}\Big)}_{\boldsymbol{N_1}} \oplus \underbrace{\Big(\boldsymbol{S_1^{(L)}} + \lambda_k\boldsymbol{S_0^{(L)}}\Big)}_{\boldsymbol{N_2}}\Big]$$
$$= 0,$$

from which it follows that $\boldsymbol{N_1} \oplus \boldsymbol{N_2}$ has a zero eigenvalue. Let α_j, $j = 1, \ldots, n{\cdot}m$ and β_h, $h = 1, \ldots, \ell$ denote the eigenvalues of $\boldsymbol{N_1}$ and $\boldsymbol{N_2}$, respectively. Since the eigenvalues of $\boldsymbol{N_1} \oplus \boldsymbol{N_2}$ are $\alpha_j + \beta_h$, to have a zero eigenvalue there must exist j' and h' such that $\alpha_{j'} = -\beta_{h'}$. The eigenvector of $\boldsymbol{N_1}$ belonging to $\alpha_{j'}$ is denoted by $\zeta_{j'}$, the one of $\boldsymbol{N_2}$ belonging to $\beta_{h'}$ is denoted by $\xi_{h'}$. Let us introduce $\mu_k = \zeta_{j'} \otimes \xi_{h'}$.

Now we show that μ_k is an eigenvector of $\boldsymbol{G_{H_0}}$, thus it satisfies (34):

$$\begin{aligned}(\boldsymbol{N_1} \oplus \boldsymbol{N_2}) \cdot \mu_k &= (\boldsymbol{N_1} \otimes \boldsymbol{I}_\ell + \boldsymbol{I}_{n+m} \otimes \boldsymbol{N_2}) \cdot (\zeta_{j'} \otimes \xi_{h'}) =\\ &= \alpha_{j'}\zeta_{j'} \otimes \xi_{h'} + \zeta_{j'} \otimes (\beta_{h'}\xi_{h'}) = 0.\end{aligned} \tag{35}$$

Next, we show that μ_k is an eigenvector of $\boldsymbol{S}$. Observe that μ_k is an eigenvector of $\boldsymbol{S}$ if and only if it is an eigenvector of $\boldsymbol{G^{(L)}}(z)|_{z=\lambda_k}$ since

$$\boldsymbol{S}\cdot\mu_k=\sum_{i=0}^{\infty}\boldsymbol{G_i^{(L)}}\boldsymbol{G_{H_0}}^i\cdot\mu_k=\sum_{i=0}^{\infty}\boldsymbol{G_i^{(L)}}\lambda_k^i\mu_k=\boldsymbol{G^{(L)}}(z)|_{z=\lambda_k}\cdot\mu_k. \tag{36}$$

As Lemma 1 states that the eigenvectors of matrix $\boldsymbol{G^{(L)}}(z)$ and matrix $\boldsymbol{F_0}+z\boldsymbol{F_L}+\boldsymbol{F_H}\boldsymbol{G^{(L)}}(z)$ are the same, it is enough to prove that μ_k is an eigenvector of $(\boldsymbol{F_0}+z\boldsymbol{F_L}+\boldsymbol{F_H}\boldsymbol{G^{(L)}}(z))|_{z=\lambda_k}$:

$$\begin{aligned}&(\boldsymbol{F_0}+\lambda_k\boldsymbol{F_L}+\boldsymbol{F_H}\boldsymbol{G^{(L)}}(\lambda_k))\cdot\mu_k\\&=\left[(\boldsymbol{D_0}\otimes\boldsymbol{I}_m\otimes\boldsymbol{I}_\ell+\lambda_k\boldsymbol{D_L}\otimes\boldsymbol{I}_m\otimes\boldsymbol{I}_\ell+\sum_{i=0}^{\infty}\lambda_k^i(\boldsymbol{D_H}\otimes\boldsymbol{I}_m\otimes\boldsymbol{I}_\ell)(\boldsymbol{\tilde{G}_i^{(L)}}\otimes\boldsymbol{I}_\ell)\right]\\&\quad\cdot(\zeta_{j'}\otimes\xi_{h'})\\&=[\boldsymbol{N_1}/\lambda_k\otimes\boldsymbol{I}_\ell]\cdot(\zeta_{j'}\otimes\xi_{h'})=\alpha_{j'}/\lambda_k(\zeta_{j'}\otimes\xi_{h'}).\end{aligned} \tag{37}$$

As $\boldsymbol{G_{H_0}}$ and $\boldsymbol{S}$ have the same eigenvectors, the same matrix diagonalizes them, consequently they commute. □

Note that the theorem can be generalized to the case when the eigenvalues are not distinct, but it requires the detailed discussion of the combination of the multiplicities of the eigenvalues of $\boldsymbol{M_1}$ and $\boldsymbol{M_2}$ (or, in case of the lemma, $\boldsymbol{N_1}$ and $\boldsymbol{N_2}$) that we neglect here.

Based on this commutative property, the next theorem makes the efficient computation of $\boldsymbol{G_{H_0}}$ and $\boldsymbol{S}$ possible.

Theorem 2. *Matrices $\boldsymbol{G_{H_0}}$ and $\boldsymbol{S}$ satisfy the following coupled matrix-quadratic equations:*

$$\begin{aligned}0&=\boldsymbol{B_1^{(H)}}+(\boldsymbol{F_0}+\boldsymbol{B_0^{(H)}})\boldsymbol{S}+\boldsymbol{F_L}\boldsymbol{G_{H_0}}\boldsymbol{S}+\boldsymbol{F_H}\boldsymbol{S}^2,\\0&=\boldsymbol{B_1^{(L)}}+(\boldsymbol{F_0}+\boldsymbol{B_0^{(L)}})\boldsymbol{G_{H_0}}+\boldsymbol{F_H}\boldsymbol{S}\boldsymbol{G_{H_0}}+\boldsymbol{F_L}\boldsymbol{G_{H_0}}^2.\end{aligned} \tag{38}$$

Proof. To obtain equations for $\boldsymbol{S}$, we multiply (13) by $\boldsymbol{G_{H_0}}^i$ from the right, sum it from $i=1$ to ∞, and add (12) to it. By using (27) we get

$$0=\boldsymbol{B_1^{(H)}}+\boldsymbol{F_L}\boldsymbol{S}\boldsymbol{G_{H_0}}+(\boldsymbol{F_0}+\boldsymbol{B_0^{(H)}})\boldsymbol{S}+\boldsymbol{F_H}\sum_{i=0}^{\infty}\sum_{k=0}^{i}\boldsymbol{G_k^{(L)}}\boldsymbol{G_{i-k}^{(L)}}\boldsymbol{G_{H_0}}^i. \tag{39}$$

The last term becomes $\boldsymbol{F_H}\boldsymbol{S}^2$ by swapping the sums and exploiting that $\boldsymbol{G_{H_0}}$ and $\boldsymbol{S}$ commute, providing the first matrix quadratic equation. The second matrix quadratic equation can be obtained from (23), when the definition of $\boldsymbol{S}$ is applied. □

Interestingly, the two matrix equations show perfect symmetry. While the solution of coupled Sylvester equations has an extensive literature (e.g. [14, 15, 16] are recent methods), there are no methods available for coupled matrix quadratic equations (according to our best knowledge). A very simple method is given by Algorithm 1. (We are sure that more efficient solution methods can be developed as well, but even this simple method performs very well in our numerical examples.)

Algorithm 1 Solving the coupled matrix quadratic equations of (38)

```
$\boldsymbol{G_{H_0}}, \boldsymbol{S} = $ SolveCoupled$\Big(\boldsymbol{F_0}, \boldsymbol{F_H}, \boldsymbol{F_L}, \boldsymbol{B_0^{(H)}}, \boldsymbol{B_1^{(H)}}, \boldsymbol{B_0^{(L)}}, \boldsymbol{B_1^{(L)}}\Big)$
begin
$k := 0$
$\boldsymbol{G_{H_0}}^{(0)} := \boldsymbol{I}$
repeat
   Solve $0 = \boldsymbol{B_1^{(H)}} + (\boldsymbol{F_0} + \boldsymbol{B_0^{(H)}} + \boldsymbol{F_L}\boldsymbol{G_{H_0}}^{(k)})\boldsymbol{S}^{(k+1)} + \boldsymbol{F_H}\boldsymbol{S}^{(k+1)^2}$ for $\boldsymbol{S}^{(k+1)}$
   Solve $0 = \boldsymbol{B_1^{(L)}} + (\boldsymbol{F_0} + \boldsymbol{B_0^{(L)}} + \boldsymbol{F_H}\boldsymbol{S}^{(k+1)})\boldsymbol{G_{H_0}}^{(k+1)} + \boldsymbol{F_L}\boldsymbol{G_{H_0}}^{(k+1)^2}$ for $\boldsymbol{G_{H_0}}^{(k+1)}$
   $k := k + 1$
until $\|\boldsymbol{G_{H_0}}^{(k)} - \boldsymbol{G_{H_0}}^{(k-1)}\| < \epsilon$ and $\|\boldsymbol{S}^{(k)} - \boldsymbol{S}^{(k-1)}\| < \epsilon$
return $\boldsymbol{G_{H_0}}^{(k)}$, $\boldsymbol{S}^{(k)}$
end
```

Note that Algorithm 1 needs only successive solution of matrix quadratic equations, that is much more efficient than the direct application of the results of Section 2 both in time and in space requirement, since the infinite series of $\boldsymbol{G_i^{(L)}}$ matrices and the solution of (23) are not needed.

Although the results above have been derived in an algebraic way (and we will keep using algebraic arguments in the rest of the paper as well), matrices $\boldsymbol{G_{H_0}}$ and $\boldsymbol{S}$ have important probabilistic interpretations. Let us denote by (n_H, n_L) the set of states in which there are n_H high and n_L low priority customers in the queue. Then, entry (a, b) of matrix $\boldsymbol{G_{H_0}}$ is the conditional probability that starting from state a in $(0, 1)$ the first visit to $(0, 0)$ occurs in state b. Similarly, the entry (a, b) of matrix $\boldsymbol{S}$ is the conditional probability that starting from state a in $(1, 0)$ the first visit to $(0, 0)$ occurs in state b. Using these probabilistic interpretations it is easy to see that $\boldsymbol{S}$ and $\boldsymbol{G_{H_0}}$ commute: let us investigate the busy period generated by a high and a low priority customer, thus the system is in $(1, 1)$ initially. Since the probability that the first passage to $(0, 0)$ occurs in state b is not affected by the order of service, we immediately have that $\boldsymbol{G_{H_0}}\boldsymbol{S} = \boldsymbol{S}\boldsymbol{G_{H_0}}$.

3.2. Some properties of matrix series $\boldsymbol{G_i^{(L)}}$ and $\boldsymbol{R_i^{(L)}}$

Let us define the following sums on matrix series $\boldsymbol{G_i^{(L)}}$ and $\boldsymbol{R_i^{(L)}}$:

$$\boldsymbol{E(G^k)} = \sum_{i=0}^{\infty} i^k \boldsymbol{G_i^{(L)}}, \qquad \boldsymbol{E(R^k)} = \sum_{i=0}^{\infty} i^k \boldsymbol{R_i^{(L)}}, \qquad k \geq 0. \tag{40}$$

According to the following theorem, $\boldsymbol{E(G^k)}$ and $\boldsymbol{E(R^k)}$ can be obtained efficiently as the solution of linear and matrix quadratic equations:

Theorem 3. *Matrix $\boldsymbol{E(G^0)}$ satisfies the following matrix-quadratic equation:*

$$\boldsymbol{0} = \boldsymbol{B_1^{(H)}} + (\boldsymbol{F_0} + \boldsymbol{F_L} + \boldsymbol{B_0^{(H)}})\boldsymbol{E(G^0)} + \boldsymbol{F_H}\boldsymbol{E(G^0)}^2. \tag{41}$$

Furthermore, matrix $\boldsymbol{E(G^k)},\ k > 0$ is the solution of the following set of linear equations:

$$\boldsymbol{0} = (\boldsymbol{F_0} + \boldsymbol{B_0^{(H)}})\boldsymbol{E(G^k)} + \sum_{a=0}^{k}\binom{k}{a}\left[\boldsymbol{F_L} + \boldsymbol{F_H}\boldsymbol{E(G^{k-a})}\right]\boldsymbol{E(G^a)}. \tag{42}$$

Similarly, matrix $\boldsymbol{E(R^0)}$ satisfies the following matrix-quadratic equation:

$$\boldsymbol{0} = \boldsymbol{E(R^0)}^2\boldsymbol{B_1^{(H)}} + \boldsymbol{E(R^0)}(\boldsymbol{F_0} + \boldsymbol{F_L} + \boldsymbol{B_0^{(H)}}) + \boldsymbol{F_H}. \tag{43}$$

Furthermore, matrix $\boldsymbol{E(R^k)},\ k > 0$ is the solution of the following set of linear equations:

$$\boldsymbol{0} = \boldsymbol{E(R^k)}(\boldsymbol{F_0} + \boldsymbol{B_0^{(H)}}) + \sum_{a=0}^{k}\binom{k}{a}\boldsymbol{E(R^a)}\left[\boldsymbol{F_L} + \boldsymbol{E(R^{k-a})}\boldsymbol{F_H}\right]. \tag{44}$$

Proof. (41) can be easily obtained by summing up equations (12) and (13) from $i = 0$ to ∞.

To prove (42), let us multiply the ith equation of (13) by i^k and sum from $i = 1$ to ∞. We get

$$\begin{aligned}\boldsymbol{0} = {} & \underbrace{\boldsymbol{F_L}\sum_{i=1}^{\infty}(i-1+1)^k\boldsymbol{G_{i-1}^{(L)}}}_{\boldsymbol{F_L}\sum_{a=0}^{k}\binom{k}{a}\boldsymbol{E(G^a)}} + \underbrace{(\boldsymbol{F_0} + \boldsymbol{B_0^{(H)}})\sum_{i=1}^{\infty} i^k\boldsymbol{G_i^{(L)}}}_{(\boldsymbol{F_0}+\boldsymbol{B_0^{(H)}})\boldsymbol{E(G^k)}} \\ & + \underbrace{\boldsymbol{F_H}\sum_{i=1}^{\infty}\sum_{j=0}^{i}(i-j+j)^k\boldsymbol{G_j^{(L)}}\boldsymbol{G_{i-j}^{(L)}}}_{\boldsymbol{F_H}\sum_{a=0}^{k}\binom{k}{a}\boldsymbol{E(G^{k-a})}\boldsymbol{E(G^a)}}.\end{aligned} \tag{45}$$

(43) and (44) can be proven similarly. □

Notice that $\boldsymbol{E(G^k)}$ and $\boldsymbol{E(R^k)},\ k > 0$, can be obtained by recursive substitution by rearranging the terms of (42) and (44) appropriately.

3.3. Analysis of the zero level using the ETAQA method

A crucial step in the efficient analysis of the low priority queue is the efficient analysis of the M/G/1 type system providing $\pi_{0,i}$. As we address only the solution of the queue length moments, we do not need the whole distribution itself. It will be shown later that to obtain the queue length moments of the low priority queue it is enough to obtain the queue length moments of the M/G/1 type system corresponding to the zero level. The ETAQA method has been developed exactly for such problems, thus it is able to calculate reward like quantities in M/G/1 type systems efficiently [12].

Taking a closer look at our M/G/1 type system (19) we can see that it is not a generic M/G/1 type system, as its matrix blocks are defined recursively according to (12) and (13). Using the ETAQA framework it is possible to obtain the queue length moments as the solution of linear systems of equations. In case of our special M/G/1 type system, however, these linear systems turn out to have closed form coefficient matrices.

The discussion on the construction of these systems and their transformations to closed form is divided into two parts:

- the present section describes how to obtain vectors $\pi_{0,0}$ and $\pi_{0,\star} = \sum_{i=1}^{\infty} \pi_{0,i}$, that play an important role in the ETAQA methodology;
- the next section describes all other details related to the queue length moments.

Now we are going to follow the same steps as in [12], but since our M/G/1 type system has only one irregular block (the top left one, see (19)), the solution will be simpler (the generator considered in [12] has three irregular blocks).

Theorem 4. *Probability vectors $\pi_{0,0}$ and $\pi_{0,\star} = \sum_{i=1}^{\infty} \pi_{0,i}$ are the solutions of the following set of linear equations:*

$$\begin{aligned}
&\begin{bmatrix}\pi_{0,0} & \pi_{0,\star}\end{bmatrix} \\
&\quad\cdot \begin{bmatrix} \boldsymbol{F_0}+\boldsymbol{F_H}\boldsymbol{G_0^{(L)}}-\sum\limits_{i=2}^{\infty}\boldsymbol{T_i}\boldsymbol{G_{H_0}} & \boldsymbol{F_L}+\boldsymbol{F_H}\sum\limits_{i=1}^{\infty}\boldsymbol{G_i^{(L)}}+\sum\limits_{i=2}^{\infty}\boldsymbol{T_i}\boldsymbol{G_{H_0}} \\ \boldsymbol{B_1^{(L)}}-\sum\limits_{i=1}^{\infty}\boldsymbol{T_i}\boldsymbol{G_{H_0}} & \boldsymbol{F_0}+\boldsymbol{B_0^{(L)}}+\boldsymbol{F_L}+\boldsymbol{F_H}\sum\limits_{i=0}^{\infty}\boldsymbol{G_i^{(L)}}+\sum\limits_{i=1}^{\infty}\boldsymbol{T_i}\boldsymbol{G_{H_0}} \end{bmatrix} \\
&\quad= \begin{bmatrix}0 & 0\end{bmatrix},
\end{aligned} \tag{46}$$

$$\pi_{0,0}\mathbb{1} + \pi_{0,\star}\mathbb{1} = 1. \tag{47}$$

Proof. The first block of equations in $\pi_0\boldsymbol{Q_0} = 0$ (see (19)) are

$$0 = \pi_{0,0}(\boldsymbol{F_0} + \boldsymbol{F_H}\boldsymbol{G_0^{(L)}}) + \pi_{0,1}\boldsymbol{B_1^{(L)}}. \tag{48}$$

To express $\pi_{0,1}$ in terms of $\pi_{0,0}$ and $\pi_{0,\star}$ we apply the same steps as in [12], thus we have

$$\pi_{0,1}\boldsymbol{B_1^{(L)}} = \pi_{0,\star}\boldsymbol{B_1^{(L)}} - \sum_{i=2}^{\infty}\pi_{0,i}\boldsymbol{B_1^{(L)}}. \tag{49}$$

By exploiting that (23) gives $\boldsymbol{B_1^{(L)}} = -\boldsymbol{T_0 G_{H_0}}$ and by applying (20), the second term can be expressed as

$$\begin{aligned}\sum_{i=2}^{\infty} \pi_{0,i} \boldsymbol{B_1^{(L)}} &= \sum_{i=2}^{\infty} \sum_{k=0}^{i-1} \pi_{0,k} \boldsymbol{T_{i-k} G_{H_0}} \\ &= \pi_{0,0} \sum_{i=2}^{\infty} \boldsymbol{T_i G_{H_0}} + \sum_{k=1}^{\infty} \pi_{0,k} \sum_{i=k+1}^{\infty} \boldsymbol{T_{i-k} G_{H_0}} \\ &= \pi_{0,0} \sum_{i=2}^{\infty} \boldsymbol{T_i G_{H_0}} + \pi_{0,\star} \sum_{i=1}^{\infty} \boldsymbol{T_i G_{H_0}},\end{aligned} \tag{50}$$

that, combining with (49) and (48) gives the first set of equations in (46).

The second set of equations in (46) is obtained by summing the block of equations of $\pi_0 \boldsymbol{Q_0} = 0$ from the second one to infinity. This gives

$$\begin{aligned}0 = \pi_{0,0} &\left(\boldsymbol{F_L} + \boldsymbol{F_H} \sum_{i=1}^{\infty} \boldsymbol{G_i^{(L)}} \right) + \sum_{k=2}^{\infty} \pi_{0,k} \boldsymbol{B_1^{(L)}} \\ &+ \sum_{k=1}^{\infty} \pi_{0,k} \left(\boldsymbol{F_0} + \boldsymbol{F_L} + \boldsymbol{B_0^{(L)}} + \boldsymbol{F_H} \sum_{i=0}^{\infty} \boldsymbol{G_i^{(L)}} \right),\end{aligned} \tag{51}$$

that, by expressing the second term according to (50) provides the second set of equations in (46).

Finally, (47) is the normalization condition corresponding to (17). □

Note that $\sum_{i=0}^{\infty} \boldsymbol{G_i^{(L)}} = \boldsymbol{E(G^0)}$ and $\sum_{i=1}^{\infty} \boldsymbol{G_i^{(L)}} = \boldsymbol{E(G^0)} - \boldsymbol{G_0^{(L)}}$. The next theorem provides closed form expressions for the remaining infinite sums in (46).

Theorem 5. *Matrices* $\sum_{i=1}^{\infty} \boldsymbol{T_i G_{H_0}}$ *and* $\sum_{i=2}^{\infty} \boldsymbol{T_i G_{H_0}}$ *satisfy*

$$\sum_{i=1}^{\infty} \boldsymbol{T_i G_{H_0}} = \\ \boldsymbol{F_L G_{H_0}} + \boldsymbol{F_H} \left[(\boldsymbol{E(G^0)} - \boldsymbol{S G_{H_0}} + \mathbb{1}\alpha)(\boldsymbol{I} - \boldsymbol{G_{H_0}} + \mathbb{1}\alpha)^{-1} + \boldsymbol{E(G^1)} \mathbb{1}\alpha - \boldsymbol{S} \right] \boldsymbol{G_{H_0}}, \tag{52}$$

and

$$\sum_{i=2}^{\infty} \boldsymbol{T_i G_{H_0}} = \sum_{i=1}^{\infty} \boldsymbol{T_i G_{H_0}} - \boldsymbol{F_L G_{H_0}} - \boldsymbol{F_H} (\boldsymbol{S} - \boldsymbol{G_0^{(L)}}), \tag{53}$$

where vector α *is the solution of* $\alpha\, \boldsymbol{G_{H_0}} = \alpha, \alpha \mathbb{1} = 1$.

Proof. Applying the definition of $\boldsymbol{T_i}$ (21) and $\boldsymbol{S}$ (27) we have

$$\begin{aligned}\sum_{i=1}^{\infty}\boldsymbol{T_i G_{H_0}} &= \boldsymbol{F_L G_{H_0}} + \boldsymbol{F_H}\sum_{i=1}^{\infty}\sum_{k=i}^{\infty}\boldsymbol{G_k^{(L)} G_{H_0}}^{k-i}\boldsymbol{G_{H_0}} \\ &= \boldsymbol{F_L G_{H_0}} + \boldsymbol{F_H}\left(\sum_{i=0}^{\infty}\sum_{k=i}^{\infty}\boldsymbol{G_k^{(L)} G_{H_0}}^{k-i} - \boldsymbol{S}\right)\boldsymbol{G_{H_0}} \\ &= \boldsymbol{F_L G_{H_0}} + \boldsymbol{F_H}\left(\sum_{k=0}^{\infty}\boldsymbol{G_k^{(L)}}\sum_{i=0}^{k}\boldsymbol{G_{H_0}}^{i} - \boldsymbol{S}\right)\boldsymbol{G_{H_0}}.\end{aligned} \tag{54}$$

It is easy to check by induction that $\boldsymbol{G_{H_0}}^i = (\boldsymbol{G_{H_0}} - \mathbb{1}\alpha)^i + \mathbb{1}\alpha$ holds for $i > 0$:

$$\begin{aligned}\boldsymbol{G_{H_0}}^i &= \left((\boldsymbol{G_{H_0}} - \mathbb{1}\alpha)^{i-1} + \mathbb{1}\alpha\right)(\boldsymbol{G_{H_0}} - \mathbb{1}\alpha + \mathbb{1}\alpha) \\ &= (\boldsymbol{G_{H_0}} - \mathbb{1}\alpha)^i + \underbrace{(\boldsymbol{G_{H_0}} - \mathbb{1}\alpha)^{i-1}\mathbb{1}\alpha}_{(\boldsymbol{G_{H_0}}^{i-1} - \mathbb{1}\alpha)\mathbb{1}\alpha = 0} + \underbrace{\mathbb{1}\alpha(\boldsymbol{G_{H_0}} - \mathbb{1}\alpha)}_{\mathbb{1}\alpha - \mathbb{1}\alpha = 0} + \underbrace{\mathbb{1}\alpha\mathbb{1}\alpha}_{\mathbb{1}\alpha}.\end{aligned} \tag{55}$$

Furthermore, by following the same arguments as in [9], page 64, we have that the inverse of matrix $\boldsymbol{I} - \boldsymbol{G_{H_0}} + \mathbb{1}\alpha$ exists as it does not have zero eigenvalue. Thus, we can express $\sum_{i=0}^{k}\boldsymbol{G_{H_0}}^i$ in a closed form as

$$\begin{aligned}\sum_{i=0}^{k}\boldsymbol{G_{H_0}}^i &= \sum_{i=0}^{k}(\boldsymbol{G_{H_0}} - \mathbb{1}\alpha)^i + k\mathbb{1}\alpha \\ &= (\boldsymbol{I} - \boldsymbol{G_{H_0}}^{k+1} + \mathbb{1}\alpha)(\boldsymbol{I} - \boldsymbol{G_{H_0}} + \mathbb{1}\alpha)^{-1} + k\mathbb{1}\alpha.\end{aligned} \tag{56}$$

Combining (54) and (56) gives

$$\begin{aligned}&\sum_{i=1}^{\infty}\boldsymbol{T_i G_{H_0}} = \boldsymbol{F_L G_{H_0}} \\ &+ \boldsymbol{F_H}\Bigg(\underbrace{\sum_{k=0}^{\infty}\boldsymbol{G_k^{(L)}}(\boldsymbol{I} - \boldsymbol{G_{H_0}}^{k+1} + \mathbb{1}\alpha)}_{\boldsymbol{E(G^0)} - \boldsymbol{S G_{H_0}} + \boldsymbol{E(G^0)}\mathbb{1}\alpha}(\boldsymbol{I} - \boldsymbol{G_{H_0}} + \mathbb{1}\alpha)^{-1} + \boldsymbol{E(G^1)}\mathbb{1}\alpha - \boldsymbol{S}\Bigg)\boldsymbol{G_{H_0}},\end{aligned}$$

which equals to (52). We expoited the fact that $\boldsymbol{E(G^0)}$ is a stochastic matrix thus $\boldsymbol{E(G^0)}\mathbb{1} = \mathbb{1}$.

Now we prove (53). Applying the definition of $\boldsymbol{T_1}$ (21) we get

$$\begin{aligned}\sum_{i=2}^{\infty}\boldsymbol{T_i G_{H_0}} &= \sum_{i=1}^{\infty}\boldsymbol{T_i G_{H_0}} - \boldsymbol{T_1 G_{H_0}} \\ &= \sum_{i=1}^{\infty}\boldsymbol{T_i G_{H_0}} - \boldsymbol{F_L G_{H_0}} - \boldsymbol{F_H}\underbrace{\sum_{k=1}^{\infty}\boldsymbol{G_k^{(L)} G_{H_0}}^{k}}_{\boldsymbol{S} - \boldsymbol{G_0^{(L)}}}.\end{aligned} \tag{57}$$

□

At this point we are able to obtain $\pi_{0,0}$ and $\pi_{0,\star}$ as the solution of a linear system, with the coefficient matrix expressed in a closed form.

3.4. Efficient computation of the queue length moments

Instead of expressing the queue length moments directly, it is more useful to introduce the following vectors:

$$E(N_L^k) = \sum_{j=0}^{\infty} j^k \pi_j^{(L)}. \tag{58}$$

The kth moment of the queue length of the low priority class is given by the sum of the elements of $E(N_L^k)$

Theorem 6. *Vectors $E(N_L^k)$, $k \geq 0$ are the solutions of the following set of linear equations:*

$$E(N_L^k) = \sum_{d=0}^{k} \binom{k}{d} E(N_L^d) \boldsymbol{E(R^{k-d})} + r_0^{(k)}, \tag{59}$$

where matrices $\boldsymbol{E(R^{k-d})}$ are given by (44) *and vectors $r_0^{(k)}$, $k \geq 0$ are defined by*

$$r_0^{(k)} = \sum_{j=0}^{\infty} j^k \pi_{0,j}. \tag{60}$$

Proof. By plugging (26) into (58), we have

$$\begin{aligned} E(N_L^k) &= \sum_{j=0}^{\infty} \sum_{a=0}^{j} (j-a+a)^k \pi_a^{(L)} \boldsymbol{R_{j-a}^{(L)}} + r_0^{(k)} \\ &= \sum_{d=0}^{k} \binom{k}{d} \sum_{a=0}^{\infty} \sum_{j=a}^{\infty} a^d \pi_a^{(L)} (j-a)^{k-d} \boldsymbol{R_{j-a}^{(L)}} + r_0^{(k)}, \end{aligned} \tag{61}$$

that gives (59). □

The real difficulty is obtaining vectors $r_0^{(k)}$. Note that $r_0^{(k)}$ is similar to the kth moment of the queue length of an M/G/1 type queue (see (19)) where the probabilities are not normalized to one but to a different constant. This little difference in the normalization does not withhold us to apply the ETAQA method, that has been developed exactly for such problems, thus to calculate reward like quantities like queue length moments efficiently. Just like in Section 3.3, our special M/G/1 type system leads to a solution that is simpler than the one presented in [12] for the general case.

Vector $r_0^{(0)}$ is obtained easily since

$$r_0^{(0)} = \sum_{j=0}^{\infty} \pi_{0,j} = \pi_{0,0} + \pi_{0,\star}. \tag{62}$$

Vectors $r_0^{(k)}, \;\; k > 0$ are given by the next theorem, which is the customization of ETAQA to our specific problem.

Theorem 7. *Vector $r_0^{(k)}, \;\; k > 0$ is the solution of the following system of linear equations:*

$$\begin{aligned} r_0^{(k)} \cdot & \left(\boldsymbol{B_0^{(L)}} + \boldsymbol{B_1^{(L)}} + \boldsymbol{F_H E(G^0)} + \boldsymbol{F_0} + \boldsymbol{F_L} \right) \\ & = -\pi_{0,0} \left(\boldsymbol{F_0} + 2^k \boldsymbol{F_L} + \boldsymbol{F_H} \sum_{d=0}^{k} \binom{k}{d} \boldsymbol{E(G^d)} \right) \\ & \quad - \sum_{a=1}^{k} \binom{k}{a} r_0^{(k-a)} \left(\boldsymbol{B_0^{(L)}} + \boldsymbol{F_0} + 2^a \boldsymbol{F_L} + \boldsymbol{F_H} \sum_{d=0}^{a} \binom{a}{d} \boldsymbol{E(G^d)} \right), \end{aligned} \tag{63}$$

$$\begin{aligned} r_0^{(k)} \cdot & \left(\boldsymbol{B_1^{(L)}} \mathbb{1} - \boldsymbol{F_L} \mathbb{1} - \boldsymbol{F_H} g^{(0)} \right) \\ & = \pi_{0,0} \left(\boldsymbol{F_L} \mathbb{1} + \boldsymbol{F_H} g^{(k)} \right) + \sum_{j=1}^{k} \binom{k}{j} r_0^{(k-j)} \left(\boldsymbol{F_L} \mathbb{1} + \boldsymbol{F_H} g^{(j)} \right), \end{aligned} \tag{64}$$

where vector $g^{(k)}, \;\; k \geq 0$ is defined by

$$g^{(k)} = \sum_{i=1}^{\infty} \sum_{a=i}^{\infty} i^k \boldsymbol{G_a^{(L)}} \mathbb{1}. \tag{65}$$

Proof. Multiplying the ith equilibrium equation in (19) by i^k, we get

$$\begin{aligned} & 1^k \pi_{0,0} (\boldsymbol{F_0} + \boldsymbol{F_H G_0^{(L)}}) + 1^k \pi_{0,1} \boldsymbol{B_1^{(L)}} = 0, \\ & 2^k \pi_{0,0} (\boldsymbol{F_L} + \boldsymbol{F_H G_1^{(L)}}) + 2^k \pi_{0,1} (\boldsymbol{F_0} + \boldsymbol{F_H G_0^{(L)}} + \boldsymbol{B_0^{(L)}}) + 2^k \pi_{0,2} \boldsymbol{B_1^{(L)}} = 0, \\ & 3^k \pi_{0,0} (\boldsymbol{F_H G_2^{(L)}}) + 3^k \pi_{0,1} (\boldsymbol{F_L} + \boldsymbol{F_H G_1^{(L)}}) + 3^k \pi_{0,2} (\boldsymbol{F_0} + \boldsymbol{F_H G_0^{(L)}} + \boldsymbol{B_0^{(L)}}) \\ & \qquad + 3^k \pi_{0,3} \boldsymbol{B_1^{(L)}} = 0, \\ & \cdots. \end{aligned} \tag{66}$$

Summing up these equations from $i = 1$ to ∞ gives:

$$\begin{aligned}0 = \pi_{0,0} &\left(\boldsymbol{F_0} + 2^k \boldsymbol{F_L} + \boldsymbol{F_H} \sum_{i=0}^{\infty} (i+1)^k \boldsymbol{G_i^{(L)}}\right) + \underbrace{\sum_{i=1}^{\infty} i^k \pi_{0,i} \boldsymbol{B_1^{(L)}}}_{r_0^{(k)} \boldsymbol{B_1^{(L)}}} \\ &+ \sum_{h=1}^{\infty} \pi_{0,h} \left(\sum_{i=0}^{\infty} (i+h+1)^k \boldsymbol{F_H} \boldsymbol{G_i^{(L)}} + (h+1)^k (\boldsymbol{F_0} + \boldsymbol{B_0^{(L)}}) + (h+2)^k \boldsymbol{F_L} \right).\end{aligned} \tag{67}$$

The last term can be simplified further as

$$\begin{aligned}&\sum_{h=1}^{\infty} \pi_{0,h} \left(\sum_{i=0}^{\infty} (i+h+1)^k \boldsymbol{F_H} \boldsymbol{G_i^{(L)}} + (h+1)^k (\boldsymbol{F_0} + \boldsymbol{B_0^{(L)}}) + (h+2)^k \boldsymbol{F_L} \right) \\ &= \sum_{a=0}^{k} \binom{k}{a} \sum_{h=1}^{\infty} \pi_{0,h} \left(\sum_{i=0}^{\infty} (i+1)^a h^{k-a} \boldsymbol{F_H} \boldsymbol{G_i^{(L)}} + h^{k-a} (\boldsymbol{F_0} + \boldsymbol{B_0^{(L)}}) + h^{k-a} 2^a \boldsymbol{F_L} \right) \\ &= \sum_{a=0}^{k} \binom{k}{a} r_0^{(k-a)} \left(\boldsymbol{F_H} \sum_{i=0}^{\infty} (i+1)^a \boldsymbol{G_i^{(L)}} + \boldsymbol{F_0} + \boldsymbol{B_0^{(L)}} + 2^a \boldsymbol{F_L} \right),\end{aligned} \tag{68}$$

which, plugging into (67) and applying $\sum_{i=0}^{\infty} (i+1)^a \boldsymbol{G_i^{(L)}} = \sum_{d=0}^{a} \binom{a}{d} \boldsymbol{E(G^d)}$ provides (63).

Note that the coefficient matrix of this set of linear equations is a generator, thus it is under-determined. An additional linear equation is obtained from the flow balance equations between the neighboring levels multiplied by i^k:

$$\begin{aligned}1^k \pi_{0,1} \boldsymbol{B_1^{(L)}} \mathbb{1} &= 1^k \pi_{0,0} \left(\boldsymbol{F_L} \mathbb{1} + \boldsymbol{F_H} \sum_{a=1}^{\infty} \boldsymbol{G_a^{(L)}} \mathbb{1} \right) \\ 2^k \pi_{0,2} \boldsymbol{B_1^{(L)}} \mathbb{1} &= 2^k \pi_{0,0} \boldsymbol{F_H} \sum_{a=2}^{\infty} \boldsymbol{G_a^{(L)}} \mathbb{1} + 2^k \pi_{0,1} \left(\boldsymbol{F_L} \mathbb{1} + \boldsymbol{F_H} \sum_{a=1}^{\infty} \boldsymbol{G_a^{(L)}} \mathbb{1} \right) \\ 3^k \pi_{0,3} \boldsymbol{B_1^{(L)}} \mathbb{1} &= 3^k \pi_{0,0} \boldsymbol{F_H} \sum_{a=3}^{\infty} \boldsymbol{G_a^{(L)}} \mathbb{1} + 3^k \pi_{0,1} \boldsymbol{F_H} \sum_{a=2}^{\infty} \boldsymbol{G_a^{(L)}} \mathbb{1} \\ &\quad + 3^k \pi_{0,2} \left(\boldsymbol{F_L} \mathbb{1} + \boldsymbol{F_H} \sum_{a=1}^{\infty} \boldsymbol{G_a^{(L)}} \mathbb{1} \right) \\ &\ldots\end{aligned} \tag{69}$$

Summing up these equations gives

$$\begin{aligned}\underbrace{\sum_{i=1}^{\infty} i^k \pi_{0,i} \boldsymbol{B_1^{(L)}} \mathbb{1}}_{r_0^{(k)} \boldsymbol{B_1^{(L)}}\mathbb{1}} &= \pi_{0,0}\left(\boldsymbol{F_L}\mathbb{1} + \boldsymbol{F_H}\underbrace{\sum_{i=1}^{\infty}\sum_{a=i}^{\infty} i^k \boldsymbol{G_a^{(L)}}\mathbb{1}}_{g^{(k)}}\right) \\ &+ \sum_{i=1}^{\infty} \pi_{0,i}\left((i+1)^k\boldsymbol{F_L}\mathbb{1} + \sum_{a=1}^{\infty} (a+i)^k \sum_{j=a}^{\infty} \boldsymbol{F_H}\boldsymbol{G_j^{(L)}}\mathbb{1}\right).\end{aligned} \tag{70}$$

With some algebra the last term simplifies to

$$\begin{aligned}&\sum_{i=1}^{\infty} \pi_{0,i}\left((i+1)^k\boldsymbol{F_L}\mathbb{1} + \sum_{a=1}^{\infty} (a+i)^k \sum_{j=a}^{\infty} \boldsymbol{F_H}\boldsymbol{G_j^{(L)}}\mathbb{1}\right) \\ &= \sum_{d=0}^{k}\binom{k}{d}\left(\underbrace{\sum_{i=1}^{\infty} i^{k-d}\pi_{0,i}}_{r^{(k-d)}}\,\boldsymbol{F_L}\mathbb{1} + \underbrace{\sum_{i=1}^{\infty} i^{k-d}\pi_{0,i}}_{r^{(k-d)}}\,\boldsymbol{F_H}\underbrace{\sum_{a=1}^{\infty}\sum_{j=a}^{\infty} a^d \boldsymbol{G_j^{(L)}}\mathbb{1}}_{g^{(d)}}\right).\end{aligned} \tag{71}$$

Finally, (70) and (71) provide (64). □

The next theorem gives an efficient way to obtain vectors $g^{(k)}$.

Theorem 8. *Vector $g^{(0)}$ is given by*

$$g^{(0)} = \boldsymbol{E(G^1)}\mathbb{1}, \tag{72}$$

and vectors $g^{(k)},\;\; k > 0$ are the solutions of the following set of linear equations:

$$\begin{aligned}0 &= \left(\boldsymbol{F_L} + \boldsymbol{F_0} + \boldsymbol{F_H} + \boldsymbol{B_0^{(H)}}\right) g^{(k)} + \boldsymbol{F_H}\sum_{d=0}^{\infty}\binom{k}{d}\boldsymbol{E(G^{k-d})}g^{(d)} \\ &+ \boldsymbol{F_L}\sum_{d=0}^{k}\binom{k}{d}\boldsymbol{E(G^d)}\mathbb{1}.\end{aligned} \tag{73}$$

Proof. (72) can be proven by simple algebraic manipulations:

$$g^{(0)} = \sum_{i=1}^{\infty}\sum_{a=i}^{\infty}\boldsymbol{G_a^{(L)}}\mathbb{1} = \sum_{a=1}^{\infty}\sum_{i=1}^{a}\boldsymbol{G_a^{(L)}}\mathbb{1} = \sum_{a=1}^{\infty} a\,\boldsymbol{G_a^{(L)}}\mathbb{1} = \boldsymbol{E(G^1)}\mathbb{1}. \tag{74}$$

To prove (73) let us multiply equations (13) by $\sum_{a=0}^{i} a^k\mathbb{1}$ from the right and sum them from $i = 1$ to ∞. We get

$$\begin{aligned}0 &= \boldsymbol{F_L}\sum_{i=1}^{\infty}\boldsymbol{G_{i-1}^{(L)}}\sum_{a=0}^{i} a^k\mathbb{1} + (\boldsymbol{F_0} + \boldsymbol{B_0^{(H)}})\sum_{i=1}^{\infty}\boldsymbol{G_i^{(L)}}\sum_{a=0}^{i} a^k\mathbb{1} \\ &+ \boldsymbol{F_H}\sum_{i=1}^{\infty}\sum_{j=0}^{i}\boldsymbol{G_j^{(L)}}\boldsymbol{G_{i-j}^{(L)}}\sum_{a=0}^{i} a^k\mathbb{1}.\end{aligned} \tag{75}$$

The first term at the right hand side of (75) simplifies to

$$\begin{aligned}\boldsymbol{F_L}\sum_{i=1}^{\infty}\boldsymbol{G^{(L)}_{i-1}}\sum_{a=0}^{i}a^k\mathbb{1} &= \boldsymbol{F_L}\sum_{i=1}^{\infty}\boldsymbol{G^{(L)}_{i-1}}\sum_{a=0}^{i-1}a^k\mathbb{1} + \boldsymbol{F_L}\sum_{i=1}^{\infty}\boldsymbol{G^{(L)}_{i-1}}(i-1+1)^k\mathbb{1}\\ &= \boldsymbol{F_L}g^{(k)} + \boldsymbol{F_L}\sum_{d=0}^{k}\binom{k}{d}\boldsymbol{E(G^{d})}\mathbb{1}.\end{aligned} \tag{76}$$

The second term of (75) equals to $(\boldsymbol{F_0}+\boldsymbol{B_0^{(H)}})g^{(k)}$. The third term needs a bit more complex treatment. Using (40) and (65) we have

$$\begin{aligned}&\boldsymbol{F_H}\sum_{i=1}^{\infty}\sum_{j=0}^{i}\boldsymbol{G^{(L)}_{j}}\boldsymbol{G^{(L)}_{i-j}}\sum_{a=0}^{i}a^k\mathbb{1} = \boldsymbol{F_H}\sum_{j=0}^{\infty}\boldsymbol{G^{(L)}_{j}}\sum_{i=j}^{\infty}\boldsymbol{G^{(L)}_{i-j}}\left(\sum_{a=0}^{j}a^k\mathbb{1}+\sum_{a=j+1}^{i}a^k\mathbb{1}\right)\\ &= \boldsymbol{F_H}\sum_{j=0}^{\infty}\boldsymbol{G^{(L)}_{j}}\sum_{a=0}^{j}a^k\underbrace{\sum_{i=j}^{\infty}\boldsymbol{G^{(L)}_{i-j}}\mathbb{1}}_{=\boldsymbol{E(G^{0})}\mathbb{1}=\mathbb{1}} + \boldsymbol{F_H}\sum_{j=0}^{\infty}\boldsymbol{G^{(L)}_{j}}\sum_{i=j}^{\infty}\boldsymbol{G^{(L)}_{i-j}}\sum_{a=1}^{i-j}(a+j)^k\mathbb{1}\\ &= \boldsymbol{F_H}g^{(k)} + \boldsymbol{F_H}\sum_{d=0}^{k}\binom{k}{d}\underbrace{\sum_{j=0}^{\infty}\boldsymbol{G^{(L)}_{j}}j^{k-d}}_{\boldsymbol{E(G^{k-d})}}\underbrace{\sum_{i=0}^{\infty}\sum_{a=1}^{i}\boldsymbol{G^{(L)}_{i}}a^d\mathbb{1}}_{g^{(d)}}.\end{aligned} \tag{77}$$

Substituting (76) and (77) into (75) provides (73). □

As an overview of the presented results, Algorithm 2 enumerates the steps needed to calculate the first K moments of the low priority queue with the proper references to the corresponding formulas.

Algorithm 2 Calculating the first K moments of the low priority queue

```
$E(N_L^0),\dots,E(N_L^K)$ = LowQLMoms$\Big(\boldsymbol{D_0},\boldsymbol{D_H},\boldsymbol{D_L},\boldsymbol{S_0^{(H)}},\boldsymbol{S_1^{(H)}},\boldsymbol{S_0^{(L)}},\boldsymbol{S_1^{(L)}},K\Big)$
begin
    Calculate matrices $\boldsymbol{F_0},\boldsymbol{F_H},\boldsymbol{F_L},\boldsymbol{B_0^{(H)}},\boldsymbol{B_1^{(H)}},\boldsymbol{B_0^{(L)}},\boldsymbol{B_1^{(L)}}$ based on (7)
    Obtain matrices $\boldsymbol{G_{H_0}}$ and $\boldsymbol{S}$ using Algorithm 1
    Obtain matrix $\boldsymbol{G_0^{(L)}}$ by solving (12)
    Obtain matrices $\boldsymbol{E(G^k)}$ and $\boldsymbol{E(R^k)}$,  $k=0\dots K$ applying Theorem 3
    Calculate matrices $\sum_{i=1}^{\infty}\boldsymbol{T_i G_{H_0}}$ and $\sum_{i=2}^{\infty}\boldsymbol{T_i G_{H_0}}$ applying Theorem 5
    Obtain matrices $\pi_{0,0}$ and $\pi_{0,\star}$ applying Theorem 4
    Calculate vectors $g^{(k)}$,  $k=0\dots K$ applying Theorem 8
    Calculate vectors $r_0^{(k)}$,  $k=0\dots K$ applying Theorem 7
    Calculate vectors $E(N_L^k)$,  $k=0\dots K$ applying Theorem 6
return $E(N_L^0),\dots,E(N_L^K)$
end
```

4. Numerical example

In this section we demonstrate the features of the presented method. All algorithms have been implemented in MATLAB. The execution environment is an average PC with an Intel Core i7 processor running at 3.4 GHz and 4 GB of RAM. The matrix-quadratic equations arising at several steps are solved by logarithmic reduction, while we used the SMCSolver package ([17]) for obtaining the fundamental matrix of the M/G/1 queue corresponding to the zero level (note that it is used only by the other methods involved in the comparison as our method does not need it).

4.1. Comparison of three solution methods

In the first numerical experiment three methods are compared to solve a particular MMAP/MAP/1 preemptive priority queue. The methods involved in the comparison are:

- The method of [1]. We transformed this method for the continuous time system we have.
- The method of [6].
- The method presented in this paper.

The MAPs describing the arrival and service processes have been generated by the inverse characterization method of [18] based on the first three marginal moments and the lag-1 autocorrelations given by Table 1 (there is no dependence between the high and low priority arrivals now).

The matrices of the MAPs generating the arrivals are as follows:

$$\boldsymbol{D_0^{(H)}} = \begin{bmatrix} -0.40571 & 0.039699 \\ 0 & -2.433 \end{bmatrix}, \quad \boldsymbol{D_1^{(H)}} = \begin{bmatrix} 0.36601 & 0 \\ 0.037527 & 2.3955 \end{bmatrix},$$
$$\boldsymbol{D_0^{(L)}} = \begin{bmatrix} -0.52822 & 0.071871 \\ 0 & -2.2718 \end{bmatrix}, \quad \boldsymbol{D_1^{(L)}} = \begin{bmatrix} 0.45635 & 0 \\ 0.16813 & 2.1037 \end{bmatrix},$$

from which the MMAP input of the queue is obtained as $\boldsymbol{D_0} = \boldsymbol{D_0^{(H)}} \oplus \boldsymbol{D_0^{(L)}}$, $\boldsymbol{D_H} = \boldsymbol{D_1^{(H)}} \otimes \boldsymbol{I}_2$, $\boldsymbol{D_L} = \boldsymbol{I}_2 \otimes \boldsymbol{D_1^{(L)}}$, thus it has 4 phases.

The MAPs corresponding to the service processes are

$$\boldsymbol{S_0^{(H)}} = \begin{bmatrix} -0.12905 & 0.10141 \\ 0 & -6.6834 \end{bmatrix}, \quad \boldsymbol{S_1^{(H)}} = \begin{bmatrix} 0.027642 & 0 \\ 0.17159 & 6.5119 \end{bmatrix},$$
$$\boldsymbol{S_0^{(L)}} = \begin{bmatrix} -0.2699 & 0.20688 \\ 0 & -4.7857 \end{bmatrix}, \quad \boldsymbol{S_1^{(L)}} = \begin{bmatrix} 0.063027 & 0 \\ 0.091873 & 4.6938 \end{bmatrix}.$$

With these parameters the utilization of the system was 0.87143.

The queue length moments of the low priority queue obtained by the presented method are depicted in Table 2.

The execution times belonging to the three methods involved in the comparison are summarized by Table 3. Both [1] and [6] require the generation

	1st moment	2nd moment	3rd moment	lag-1 ac.
High pr. arrival	0.7	2	13	0.3
Low pr. arrival	1	3	16	0.2
High pr. service	0.4	4	92	0.1
Low pr. service	0.3	0.8	8	0.1

Table 1: Parameters of the MAPs for the numerical example

k	kth moment of the queue length
0	1
1	115.78912
2	40074.713
3	21967427
4	1.6419872×10^{10}
5	1.5518066×10^{13}
6	1.7714857×10^{16}
7	2.3689384×10^{19}
8	3.630129×10^{22}

Table 2: Queue length moments of the low priority queue

of matrix series $\boldsymbol{R_k^{(L)}}$. In this particular example 1513 elements of this matrix series were calculated to achieve the stopping criteria that the largest element of the last $\boldsymbol{R_k^{(L)}}$ matrix is less then 10^{-9}. This alone is a significant computational effort, it took 54 seconds on our PC. Next, the solution of the M/G/1 system corresponding to the zero level is required for calculating matrix $\boldsymbol{G_{H_0}}$ and the $\pi_{0,i}$ probabilities. Our simple iterative algorithm to solve the coupled matrix quadratic equations turned out to be more efficient to calculate $\boldsymbol{G_{H_0}}$ than the Newton iteration based M/G/1 type solver. Regarding the $\pi_{0,i}$ probabilities, the method in [1] requires the whole $\pi_{0,i},\ i \geq 0$ distribution, while the other two methods require only some elements of it. To be precise, the method in [6] needs $\pi_{0,0}$ and $\pi'_0 = \sum_{i=0}^{\infty} i\pi_{0,i}$, that are calculated by a system of linear equations (with coefficient matrix containing infinite sums), while the presented method needs $\pi_{0,0}$ and $\pi_{0,\star}$ given by the ETAQA method with all infinite sums eliminated. Finally, the last row in the Table shows the time between the analysis of the zero level and the completion of the queue length moments. Again, [1] is the slowest, since it operates with the huge matrix $\boldsymbol{R}$. [6] is much better, because it needs to solve only a couple of much smaller systems of linear equations. Finally, the new method is the fastest here as well, since the coefficient matrices of the arising linear systems of equations have no infinite summations.

The speed advantage of the presented method becomes more pronounced when the size of the MAPs increases. The small example we studied so far has only 16 phases (as the arrival process has 4 phases, and the service processes of both the high and low priority class have 2 phases). Table 4 shows the analysis times with more phases. To get 32 phases we increased the MAP of

	[1]	[6]	**new method**
generating $\boldsymbol{R}_{\boldsymbol{k}}^{(\boldsymbol{L})}$ matrices:	54.3s	54.3s	-
obtaining $\boldsymbol{G}_{\boldsymbol{H}_{\mathbf{0}}}$:	0.3s	0.3s	**0.1s**
analysis of level zero:	2.6s	0.05s	**0.004s**
queue length moments:	87.7s	0.2s	**0.005s**
Total execution time:	145s	54.8s	**0.11s**

Table 3: Execution time analysis

the low priority arrivals by two as $\boldsymbol{D}_{\mathbf{0}}^{(\boldsymbol{L})\times\mathbf{2}} = \boldsymbol{D}_{\mathbf{0}}^{(\boldsymbol{L})} \oplus \boldsymbol{D}_{\mathbf{0}}^{(\boldsymbol{L})}/2$ and $\boldsymbol{D}_{\mathbf{1}}^{(\boldsymbol{L})\times\mathbf{2}} = \boldsymbol{D}_{\mathbf{1}}^{(\boldsymbol{L})} \oplus \boldsymbol{D}_{\mathbf{1}}^{(\boldsymbol{L})}/2$ (the $\boldsymbol{D}_{\mathbf{0}}, \boldsymbol{D}_{\boldsymbol{H}}$ and $\boldsymbol{D}_{\boldsymbol{L}}$ matrices of the MMAP input of the queue are computed as before). To get 64 phases the MAPs of arrivals of both classes have been increased in the same way. The method of [1] was not able to handle more than 16 phases, because matrix $\boldsymbol{R}$ in [1], page 30, did not fit into the 4 GB of memory we had. As it can be seen on Table 4, our method is clearly faster, mainly by avoiding the calculation of the $\boldsymbol{R}_{\boldsymbol{k}}^{(\boldsymbol{L})}$ matrices.

	16 phases	32 phases	64 phases
method of [1]	145s	-	-
method of [6]	54.8s	132s	2612s
new method	**0.11s**	**0.33s**	**6s**

Table 4: Execution times vs. the number of phases

(Note that in all examples in this paper, except for this one, the small input has been used with 16 phases.)

4.2. Estimation of the queue length distribution based on moments

The execution times studied above correspond to the first moment only. Our method turned out to be very robust and is able to calculate a large number of moments without any difficulties. As depicted on Figure 1, the execution time does increase only slightly with the number of moments.

Although only the first few moments have intuitive meaning, it is still reasonable to calculate a large number of moments. According to [19] it is possible to give upper and lower bounds for the queue length distribution based on the moments. Figure 2 depicts the bounds with increasing number of moments involved into the estimation. We were not able to utilize more than 21 moments due to numerical reasons: [19] needs to solve two linear systems, one of them has a Hankel, the other has a Vandermonde coefficient matrix. With a large number of moments both are ill-conditioned due to the limited accuracy of the machine representation of floating point numbers.

5. Conclusion

The method presented in this paper calculates queue length moments of the MMAP/MAP/1 preemptive priority queue in a faster and more accurate

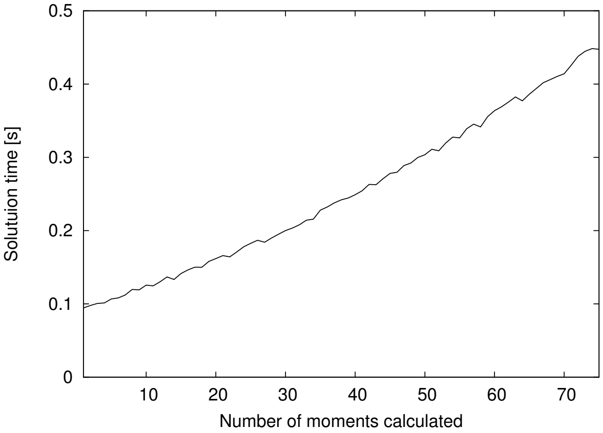

Figure 1: Solution time vs. the number of moments calculated

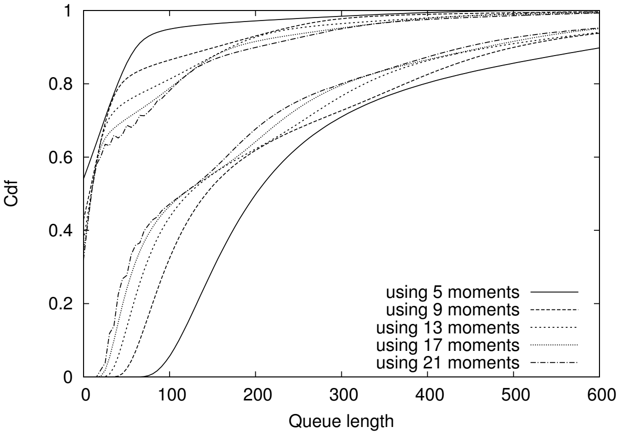

Figure 2: Upper and lower bounds for the queue length distribution based on moments

way than former methods. To achieve these results this paper reveals some yet unknown relations between fundamental matrices of preemptive priority queues.

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

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

[3] J.-A. Zhao, B. Li, X.-R. Cao, I. Ahmad, A matrix-analytic solution for the DBMAP/PH/1 priority queue, Queueing Systems 53 (2006) 127–145.

[4] T. Takine, T. Hasegawa, The nonpreemptive priority MAP/G/1 queue with two classes of customers, Journal of Operations Research Society of Japan 39 (2) (1996) pp. 266–290.

[5] T. Takine, The nonpreemptive priority MAP/G/1 queue, Operations Research 47 (6) (1999) pp. 917–927.

[6] A. Horváth, G. Horváth, M. Telek, A traffic based decomposition of two-class queueing networks with priority service, Comput. Netw. 53 (2009) 1235–1248.

[7] S. Asmussen, G. Koole, Marked point processes as limits of Markovian arrival streams, J. Appl. Probab. 30 (1993) 365–372.

[8] Q.-M. He, M. F. Neuts, Markov chains with marked transitions, Stochastic Processes and their Applications 74 (1) (1998) 37 – 52.

[9] G. Latouche, V. Ramaswami, Introduction to matrix analytic methods in stochastic modeling, ASA-SIAM series on statistics and applied probability, Society for Industrial and Applied Mathematics, 1999.

[10] V. Ramaswami, A stable recursion for the steady state vector in Markov chains of M/G/1-type, Commun. Statist. Stochastic Models 4 (1988) 183–188.

[11] M. F. Neuts, Structured Stochastic Matrices of M/G/1 Type and Their Applications, CRC Press, 1989.

[12] G. Ciardo, W. Mao, A. Riska, E. Smirni, ETAQA-MG1: an efficient technique for the analysis of a class of M/G/1-type processes by aggregation, PERFORMANCE EVALUATION 57 (3) (2004) 235–260.

[13] G. Horváth, B. V. Houdt, M. Telek, Commuting matrices in MAP/MAP/1 queues, Tech. rep. (2011).

[14] F. Ding, T. Chen, On iterative solutions of general coupled matrix equations, SIAM J. Control Optim. 44 (2006) 2269–2284.

[15] S.-K. Li, T.-Z. Huang, Two matrix iterative methods for solving general coupled matrix equations, in: Proceedings of the 2010 International Conference on Computational and Information Sciences, ICCIS '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 384–387.

[16] L. Xie, H. Yang, Y. Liu, F. Ding, Iterative solutions for general coupled matrix equations with real coefficients, in: American Control Conference (ACC), 2011, 2011, pp. 669 –674.

[17] D. A. Bini, B. Meini, S. Steffé, B. Van Houdt, Structured Markov chains solver: software tools, in: Proceeding from the 2006 workshop on Tools for solving structured Markov chains, SMCtools '06, ACM, New York, NY, USA, 2006.

[18] L. Bodrog, A. Heindl, G. Horváth, M. Telek, A Markovian canonical form of second-order matrix-exponential processes., European Journal of Operational Research 190 (2) (2008) 459–477.

[19] S. Rácz, Á. Tari, M. Telek, A moments based distribution bounding method, Mathematical and Computer Modelling 43 (1112) (2006) 1367 – 1382.