Analyzing multi-class, multi-server queueing systems with preemptive priorities

Andrei Sleptchenko, Aart van Harten
& Matthieu van der Heijden

WP-77

BETA-publicatie: WP-77
ISSN: 1386-9213
NUGI: 684
Enschede: June 2002

Analyzing multi-class, multi-server queueing systems with preemptive priorities

Andrei Sleptchenko*, Aart van Harten, Matthieu van der Heijden
Faculty of Technology anf Management, University of Twente,
P.O. box 217, 7500 AE Enschede, The Netherlands.

Abstract

In this paper we consider a multi-class, multi-server queueing system with preemptive priorities. We distinguish two groups of priority classes that consist of multiple items, each having their own arrival and service rate. We assume Poisson arrival processes and exponentially distributed service times. We derive an approximate method to estimate the steady state probabilities with an approximation error that can be made as small as desired at the expense of some more numerical matrix iterations. Based on these probabilities, we can derive approximations for a wide range of relevant performance characteristics, such as the expected postponement time for each item class and the first and second moment of the number of items of a certain type in the system. We illustrate our method with some numerical examples. Comparison to simulation results shows that with a moderate number of matrix iterations (~ 20) we can estimate key performance measures, such as the mean and variance of the number of items in the system, with an error less than 1% in most cases.

Keywords: Multi-server, multi-class queue; Peemptive priority

1 Introduction.

Multi-server priority queueing systems arise in various applications, such as computer and telecommunication systems. We encountered this model during our research on spare parts logistics for repairable items (Sherbrooke [13], Sleptchenko et al. [12]). In such situations we aim to minimize holding costs for spare parts stocks given a fixed system availability, or to maximize system availability given a fixed budget for spare parts. The queueing models are necessary to model repair facilities. Particularly, we need the first two moments of the number of items in the repair shop. By introducing priority rules into these spare part models, we aim to gain efficiency in stock optimization, since we can give high priority to expensive items, thereby reducing stock of these expensive items while increasing the stocks of (low priority) cheap items. In this way, we hope to attain the target system availability at lower costs.

A repair shop in spare part networks is generally able to handle multiple items, that we classify as either high priority or low priority items. Each item has its own arrival rate and service time distribution. As a consequence, we need to model a repair shop by a (multi-server) priority queueing system with two priority classes, where each class consists of multiple subclasses (item types). An algorithm to determine performance characteristics of such multi-server, multi-class priority queueing systems is not available in the literature as far as we know. Therefore, we develop our own algorithm in this paper, assuming Poisson arrivals and exponential service times. We expect that such an algorithm can be used for other applications as well, such as computer, communication and production systems. Although we proceed from a model with

*Corresponding author: a.sleptchenko@sms.utwente.nl (A.Sleptchenko)

exponential service times and two priority groups it is clear that we can in principle extend our method to analyze systems with more priority classes and with hyperexponential service times.

There is quite some literature on single server priority queueing systems. However, multi-server priority queueing systems have received much less attention. Recently, such models have been studied by Mitrani et al [10], Gail et al. [3, 4], Kao et al. [7, 8], Kella and Yechiali [9] and Wagner [15, 16, 17]. Results are available both if preemption is allowed, so high priority items may interrupt the service of low priority items, or not.

The non-preemptive queues are analyzed most extensively by Wagner [16], who considers multi-server nonpreemptive priority systems with Markovian arrival process, service times having phase type distributions and both finite or infinite queueing space. Another interesting approach to non-preemptive priority queues is proposed by Kao and Wilson [8]. They apply a power-series approach to estimate the performance characteristics. The power-series approach has been introduced by Hooghiemstra et al. [5] and has been applied before to solve a variety of queueing problems – particularly those with multidimensional state space. Kao and Wilson [8] apply this approach to a multi-server queue with two priority classes and no preemption. They compare the performance of their method to a matrix analytic approach as described in Kao and Narayanan [7]. The power series approach is interesting, because it can easily be implemented and it can be extended to include more than two priority classes and to preemptive priorities in theory. However these extensions cause an enormous growth of memory requirements and computation time.

Among the papers on preemptive priorities, we mention the approximation approach of Buzen [1] and the generating function approach as proposed by Mitrani et al [10] and by Gail et al. [4]. The basic idea of the Buzen's approximation approach is to replace k servers by a single server that works k times as fast and to use a correction factor, being the ratio of the waiting times when the same trick would be applied to the non-priority multi-server queue. Although this approach is attractive because of its simplicity and its extendability to general service times, it was done only for the first moments of number of items in the system. In contrast to Buzen's idea, the generating function approach gives exact results for the first two moments of the number of items in the system. However, these approaches ([4, 10]) can only be applied to cases with two classes.

To analyze multi-class, multi-server queues with two priority groups each containing several item classes, we proceed as follows. First we construct the equilibrium state equations (section 2). To solve this set of equations, we develop an approximate approach, which gives estimates of the system states probabilities. From a computational point of view, an exact solution is difficult to achieve, but it can be approximated as close as desired at the expense of moderate numerical work. To demonstrate this we shall heavily rely upon in-depth insight in the structure of the exact solution. We distinguish three regions when solving the equilibrium equations, namely (1) states with at least one high priority item in the queue, (2) states with only low priority items in the queue, and (3) states in which the queue is empty. We deal with each region separately. In section 3 we solve the equilibrium equations in region 1 (high priority items in the queue). Next, we show how to deal with the remaining equations (no high priority items in the queue) in section 4. For these equations, we can only approximate the system state probabilities. Using the (approximate) state probabilities, we can derive various system performance characteristics (expected waiting times per type, expected queue length per type and even correlations between types, expected postponement time per type). In section 5, we show as an example how to derive the first two moments of the number of items in the system for each type and the expected postponement time per type. We compare our approximations to simulation results in section 6. Finally, we present our conclusions and we discuss some model extensions in section 7. Some of the details of proofs and derivations are given in appendices so that the general line through

the paper becomes clearer.

2 The Model.

2.1 Definitions and notation.

As mentioned before, customers are processed according to a preemptive priority rule, i.e. when a high priority (hp) item arrives and no server is available, one of the low priority (lp) items in service (if there is one) is taken out of service (postponed) to allow the high priority item to be served. As preemption discipline we suppose that if preemption occurs each lp item in service is chosen with equal probability. When a server comes available again, one of the postponed items is taken back into the service (resumed). We also suppose a resume discipline with equal probability for each of the postponed items. We denote the number of item classes with high (low) priority by N^h (N^l). High priority jobs from subclass i arrive according to a Poisson process with rate λ_i^h and low priority jobs from subclass j arrive with rate λ_j^l. The service times of the subclasses are exponentially distributed with rates μ_i^h and μ_j^l for high and low priority item classes, respectively. Because of the memoryless property of the exponential distribution, it does not make a difference whether postponed jobs are resumed from the moment of interruption on or whether they are restarted completely. All servers are equal, and if multiple servers are available to process a job, each available server has an equal chance to get this job. We use ρ_i^h, ρ_j^l to denote the utilization rates of high and low priority item classes in the system. We denote the total number of high priority items by n and the number items in high priority type i by n_i. For the low priority items, we will use the notation m and m_j respectively.

We characterize the system state by five vectors of dimensions N^h and N^l, where the components of each vector refer to the (high and low priority) subclasses. The first four vectors are obvious, i.e. vectors containing information about the items in queue and in service:

$\overline{s}^h$ and $\overline{s}^l$ – vectors containing the number of high and low priority items in service per item class.

$\overline{w}^h$ and $\overline{w}^l$ – vectors containing the number of high and low priority items in the queue waiting for first service per item class.

Next, we need one more vector to keep track of low priority items that have been withdrawn from service when a high priority item arrived. This vector is necessary since items with longer processing time will be withdrawn (postponed) more often and then the probability to have in front of server one of these slow moving items is higher. Thus, the fifth vector is:

$\overline{r}^l$ – vector containing the number of postponed low priority items per item class.

Then the systems state probabilities are denoted by $P_{n,m}\left(\overline{w}^h, \overline{s}^h, \overline{w}^l, \overline{s}^l, \overline{r}^l\right)$.

Other general notations used throughout the paper are:

Λ^h, Λ^l, μ^h, μ^l – sums of the arrival rates and service rates for each class, i.e. $\Lambda^h = \sum_{i=1}^{N^h} \lambda_i^h$, $\Lambda^l = \sum_{i=1}^{N^l} \lambda_i^l$ and $\mu^h = \Lambda^h \Big/ \sum_{i=1}^{N^h} \frac{\lambda_i^h}{\mu_i^h} = \frac{\Lambda^h}{k\rho^h}$, $\mu^l = \Lambda^l \Big/ \sum_{i=1}^{N^l} \frac{\lambda_i^l}{\mu_i^l} = \frac{\Lambda^l}{k\rho^l}$, where general utilization rates for each class are $\rho^h = \frac{\Lambda^h}{k\mu^h}$, $\rho^l = \frac{\Lambda^l}{k\mu^l}$, and the total utilization rate is $\rho = \rho^h + \rho^l$.

a_i^h, a_i^l – fractions of arrival rates, i.e. $a_i^h = \frac{\lambda_i^h}{\Lambda^h}$, $a_i^l = \frac{\lambda_i^l}{\Lambda^l}$.

δ_i^h, δ_i^l – perturbations of service rates, i.e. $\left(1+\delta_i^h\right)=\frac{\mu_i^h}{\mu^h}$, $\left(1+\delta_i^l\right)=\frac{\mu_i^l}{\mu^h}$.

γ – fraction of general service rates of high and low priority items, i.e. $\gamma=\frac{\mu^l}{\mu^h}$.

$\overline{\mu}\left(\overline{s}^h,\overline{s}^l\right)$ – sum of service rate of all items in service, i.e. $\overline{\mu}\left(\overline{s}^h,\overline{s}^l\right)=\sum_{i=1}^{N^h}s_i^h\mu_i^h+\sum_{i=1}^{N^l}s_i^l\mu_i^l$

$\overline{\delta}\left(\overline{s}^h,\overline{s}^l\right)$ – sum of perturbations of service rate of all items in service,
i.e. $\overline{\delta}\left(\overline{s}^h,\overline{s}^l\right)=\frac{1}{k}\left(\sum_{i=1}^{N^h}s_i^h\delta_i^h+\sum_{i=1}^{N^l}s_i^l\delta_i^l\right)$

x_i – the i^{th} component of any vector $\overline{x}$.

$\overline{e}_i^h$ ($\overline{e}_i^l$) – a vector of dimension N^h (N^l) with component i equal to 1 and all other components equal to 0; this vector is used to indicate the changes in vectors $\overline{w}^h$ and $\overline{s}^h$ ($\overline{w}^l$, $\overline{r}^l$ and $\overline{s}^l$) during transitions from state to state.

e_{ij}^h (e_{ij}^l) – denotes the j^{th} component of the vector e_i^h (e_i^l), so e_{ij}^h (e_{ij}^l) $=1$ if $i=j$ and 0 otherwise.

$|\overline{x}|$ – denotes the sum of all components of any vector $\overline{x}$.

We will introduce the remaining notation later on. The full list of notations can be found in the appendix.

2.2 Stationary state equations.

In this section, we will write down the equilibrium state equations for the continuous time Markov chain. That is, the net exchange of probability in an infinitesimal interval from a given state with its neighbors has to be zero in an equilibrium situation. Neighbors of a state $\left(\overline{w}^h,\overline{s}^h,\overline{w}^l,\overline{s}^l,\overline{r}^l\right)$ with n clients of high priority and m clients of low priority are states to/from which a one-step transition is possible, either by an item arrival or by a service completion. According to the numbers n and m of items in the system, the system states (so the equilibrium equations) can be divided into three areas (see figure 1):

I. there is at least one high priority item in the queue ($n>k$, $m\geq 0$),

II. all servers are busy, but there is no high priority item in the queue ($n\leq k$, $n+m\geq k$),

III. there is at least one server available ($n+m<k$),

All these subspaces have different equilibrium equations. Besides, we have to consider the equations for the two boundaries between the regions separately.

In **area I** ($n>k$, $m\geq 0$), we have states with all servers occupied and high priority items in the queue, so no low priority items are in service ($\overline{s}^l=0$). Therefore the transitions from the neighbors of state $\left(\overline{w}^h,\overline{s}^h,\overline{w}^l,\overline{0},\overline{r}^l\right)$ are due to:

(1) the arrival of high and low priority jobs that enter the queue

(2) the service completion of a high priority job; in this case, we consider all combinations (i,j) where i represents the subclass of the item for which service is completed and j is the subclass of the high priority item that enters service.

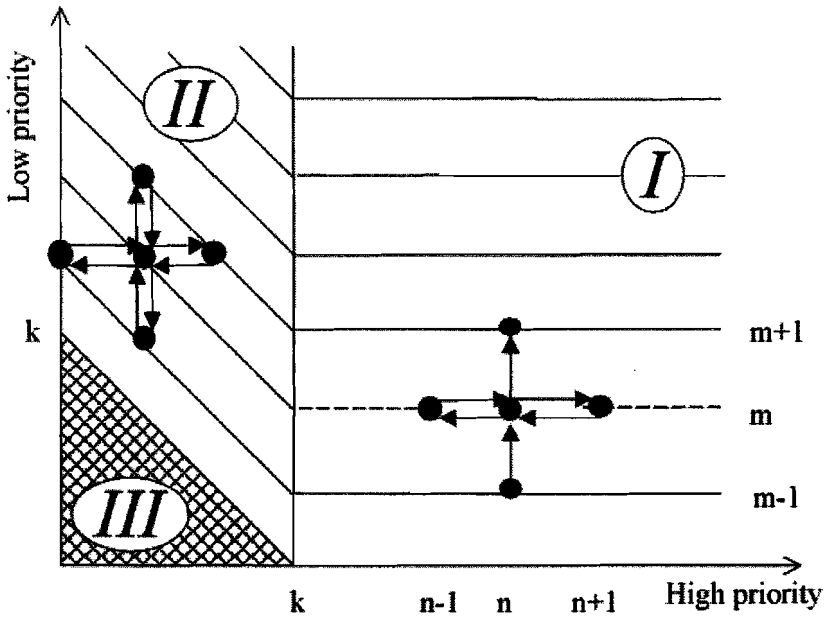

Figure 1: Three sets of probability states

Hence the equilibrium equations are:

$$\begin{aligned}
&\left(\Lambda^h + \Lambda^l + \overline{\mu}\left(\overline{s}^h, \overline{0}\right)\right) P_{n,m}\left(\overline{w}^h, \overline{s}^h, \overline{w}^l, \overline{0}, \overline{r}^l\right) \\
&= \sum_{i=1}^{N^h} \lambda_i^h P_{n-1,m}\left(\overline{w}^h - \overline{e}_i^h, \overline{s}^h, \overline{w}^l, \overline{0}, \overline{r}^l\right) + \sum_{i=1}^{N^l} \lambda_i^l P_{n,m-1}\left(\overline{w}^h, \overline{s}^h, \overline{w}^l - \overline{e}_i^l, \overline{0}, \overline{r}^l\right) \qquad (1) \\
&+ \sum_{i=1}^{N^h}\sum_{j=1}^{N^h} \frac{w_j^h + 1}{|\overline{w}^h| + 1}\left(s_i^h + 1 - e_{ij}^h\right) \mu_i^h P_{n+1,m}\left(\overline{w}^h + \overline{e}_j^h, \overline{s}^h + \overline{e}_i^h - \overline{e}_j^h, \overline{w}^l, \overline{0}, \overline{r}^l\right)
\end{aligned}$$

where $\frac{w_j^h+1}{|\overline{w}^h|+1}$ characterizes the probability that the high priority item with class j is in front of the queue.

In **area II** ($n \leq k$, $n+m \geq k$) the equilibrium equations are different for the internal states (i.e. $n < k$, $n+m > k$) and for the boundary states (i.e. $n = k$, $n+m > k$ for II-III and $n+m = k$ for I-II).

In the **internal states of area II** ($n < k$, $n+m > k$) all servers are busy, low priority items are in service and no high priority items are in the queue (i.e. $\overline{s}^l \neq 0$, $\overline{w}^h = 0$). So, the transitions from the neighbors of state $\left(\overline{0}, \overline{s}^h, \overline{w}^l, \overline{s}^l, \overline{r}^l\right)$ occur due to

(1) the arrival of a low priority job that enters the queue,

(2) the arrival of a high priority job that is served directly, thereby preempting a low priority job and changing the vector $\overline{r}^l$; we assume that low priority items to be withdrawn from service are selected randomly, so the probability that an item of subclass j is selected equals $\frac{s_j^l+1}{|\overline{s}^l|+1}$,

(3) service completion of a (high or low priority) item type i without postponed items in the queue ($\overline{r}^l = \overline{0}$), so a new service of a low priority item type j is started; note that subclass j is at the front of the queue with probability $\frac{w_j^l+1}{|\overline{w}^l|+1}$,

(4) service completion of a (high or low priority) item type i with postponed items in the queue ($\overline{r}^l \neq \overline{0}$), so the service of a postponed low priority item type j is continued; note that

subclass j is at the front of the queue with probability $\frac{r_j^l+1}{|\overline{r}^l|+1}$.

When writing down the equilibrium equations, we use the fact that the service rates of the postponed items and the items in the queue are the same due to the memoryless property of exponential distribution of the service times. Then, the equilibrium equations are:

$$\begin{aligned}
&\left(\Lambda^h+\Lambda^l+\overline{\mu}\left(\overline{s}^h,\overline{s}^l\right)\right)P_{n,m}\left(\overline{0},\overline{s}^h,\overline{w}^l,\overline{s}^l,\overline{r}^l\right)=\sum_{i=1}^{N^l}\lambda_i^l P_{n,m-1}\left(\overline{0},\overline{s}^h,\overline{w}^l-\overline{e}_i^l,\overline{s}^l,\overline{r}^l\right)\\
&+\sum_{i=1}^{N^h}\sum_{j=1}^{N^l}\lambda_i^h\frac{s_j^l+1}{|\overline{s}^l|+1}P_{n-1,m}\left(\overline{0},\overline{s}^h-\overline{e}_i^h,\overline{w}^l,\overline{s}^l+\overline{e}_j^l,\overline{r}^l-\overline{e}_j^l\right)\\
&+I\left(\left|\overline{r}^l\right|=0\right)\sum_{i=1}^{N^h}\sum_{j=1}^{N^l}\frac{w_j^l+1}{|\overline{w}^l|+1}\left(s_i^h+1\right)\mu_i^h P_{n+1,m}\left(\overline{0},\overline{s}^h+\overline{e}_i^h,\overline{w}^l+\overline{e}_j^l,\overline{s}^l-\overline{e}_j^l,\overline{0}\right)\\
&+I\left(\left|\overline{r}^l\right|=0\right)\sum_{i=1}^{N^l}\sum_{j=1}^{N^l}\frac{w_j^l+1}{|\overline{w}^l|+1}\left(s_i^l+1-e_{ij}^l\right)\mu_i^l P_{n,m+1}\left(\overline{0},\overline{s}^h,\overline{w}^l+\overline{e}_j^l,\overline{s}^l+\overline{e}_i^l-\overline{e}_j^l,\overline{0}\right)\\
&+\sum_{i=1}^{N^h}\sum_{j=1}^{N^l}\frac{r_j^l+1}{|\overline{r}^l|+1}\left(s_i^h+1\right)\mu_i^h P_{n+1,m}\left(\overline{0},\overline{s}^h+\overline{e}_i^h,\overline{w}^l,\overline{s}^l-\overline{e}_j^l,\overline{r}^l+\overline{e}_j^l\right)\\
&+\sum_{i=1}^{N^l}\sum_{j=1}^{N^l}\frac{r_j^l+1}{|\overline{r}^l|+1}\left(s_i^l+1-e_{ij}^l\right)\mu_i^l P_{n,m+1}\left(\overline{0},\overline{s}^h,\overline{w}^l,\overline{s}^l+\overline{e}_i^l-\overline{e}_j^l,\overline{r}^l+\overline{e}_j^l\right)
\end{aligned}\tag{2}$$

where $I(\cdot)$ denotes the indicator function, so the value of the function is 1 if the statement between parentheses is true and 0 otherwise.

Then, we consider the **border between the areas I and II** ($n=k$). That is, the number of high priority jobs in the system equals the number of servers. So no high priority jobs are waiting in the queue and no low priority jobs are in service. Then the transitions to the state $\left(\overline{0},\overline{s}^h,\overline{w}^l,\overline{0},\overline{r}^l\right)$ are:

(1) the arrival of a high priority job i that enters service immediately, thereby preempting the single low priority job j that was in service,

(2) the arrival of a low priority job that enters the queue,

(3) the service completion of a high priority job i, causing that the single high priority job in the queue (type j) is being served; note that a low priority job cannot be completed, since all servers are busy with high priority jobs.

So the equilibrium equations are:

$$\begin{aligned}
&\left(\Lambda^h+\Lambda^l+\overline{\mu}\left(\overline{s}^h,\overline{s}^l\right)\right)P_{n,m}\left(\overline{0},\overline{s}^h,\overline{w}^l,\overline{0},\overline{r}^l\right)=\\
&\sum_{i=1}^{N^h}\sum_{j=1}^{N^l}\lambda_i^h P_{n-1,m}\left(\overline{0},\overline{s}^h-\overline{e}_i^h,\overline{w}^l,\overline{e}_j^l,\overline{r}^l-\overline{e}_j^l\right)+\sum_{i=1}^{N^l}\lambda_i^l P_{n,m-1}\left(\overline{0},\overline{s}^h,\overline{w}^l-\overline{e}_i^l,\overline{0},\overline{r}^l\right)\\
&+\sum_{i=1}^{N^h}\sum_{j=1}^{N^h}\left(s_i^h+1-e_{ij}^h\right)\mu_i^h P_{n+1,m}\left(e_j,\overline{s}^h-\overline{e}_j^h+\overline{e}_i^h,\overline{w}^l,\overline{0},\overline{r}^l\right)
\end{aligned}\tag{3}$$

Next, we consider the **border between the areas II and III** ($n+m=k$, $n<k$). That is, all servers are occupied (with high and/or low priority jobs), the queue is empty and there is at least one low priority item in service. Then the transitions to the state $\left(\overline{0},\overline{s}^h,\overline{0},\overline{s}^l,\overline{0}\right)$ are:

(1) the arrival of a (high or low priority) job i that enters service immediately,

(2) the service completion of a high priority job i, causing that a single *low* priority job in the queue (type j) is being served; note that a low priority job that is taken into service may be either a new service or a postponed service,

(3) the service completion of a low priority job i, causing that the single low priority job in the queue (type j) is being served; again, a low priority job that is taken into service may be either a new or a postponed service; note that a high priority job cannot be in the queue if a low priority job is being served.

$$
\begin{aligned}
&\left(\Lambda^h + \Lambda^l + \overline{\mu}\left(\overline{s}^h, \overline{s}^l\right)\right) P_{n,m}\left(\overline{0}, \overline{s}^h, \overline{0}, \overline{s}^l, \overline{0}\right) = \\
&\sum_{i=1}^{N^h} \lambda_i^h P_{n-1,m}\left(\overline{0}, \overline{s}^h - \overline{e}_i^h, \overline{0}, \overline{s}^l, \overline{0}\right) + \sum_{i=1}^{N^l} \lambda_i^l P_{n,m-1}\left(\overline{0}, \overline{s}^h, \overline{0}, \overline{s}^l - \overline{e}_i^l, \overline{0}\right) \\
&+ I\left(\left|\overline{r}^l\right| = 0\right) \sum_{i=1}^{N^h} \sum_{j=1}^{N^l} \left(s_i^h + 1\right) \mu_i^h P_{n+1,m}\left(\overline{0}, \overline{s}^h + \overline{e}_i^h, \overline{e}_j^l, \overline{s}^l - \overline{e}_j^l, \overline{0}\right) \\
&+ \sum_{i=1}^{N^h} \sum_{j=1}^{N^l} \left(s_i^h + 1\right) \mu_i^h P_{n+1,m}\left(\overline{0}, \overline{s}^h + \overline{e}_i^h, \overline{0}, \overline{s}^l - \overline{e}_j^l, \overline{e}_j^l\right) \qquad (4) \\
&+ I\left(\left|\overline{r}^l\right| = 0\right) \sum_{i=1}^{N^l} \sum_{j=1}^{N^l} \left(s_i^l + 1 - e_{ij}\right) \mu_i^l P_{n,m+1}\left(\overline{0}, \overline{s}^h, \overline{e}_j^l, \overline{s}^l + \overline{e}_i^l - \overline{e}_j^l, \overline{0}\right) \\
&+ \sum_{i=1}^{N^l} \sum_{j=1}^{N^l} \left(s_i^l + 1 - e_{ij}\right) \mu_i^l P_{n,m+1}\left(\overline{0}, \overline{s}^h, \overline{0}, \overline{s}^l + \overline{e}_i^l - \overline{e}_j^l, \overline{e}_j^l\right)
\end{aligned}
$$

Finally we consider probability states in **area III** ($n + m < k$). These states have empty queue and hence no items are postponed ($\overline{r}^l = \overline{0}$). Now the transitions from the neighbors of the state $\left(\overline{0}, \overline{s}^h, \overline{w}^l, \overline{s}^l, \overline{r}^l\right)$ become much simpler. They consist of:

(1) the arrival of a high priority job that enters service immediately,

(2) the arrival of a low priority job that enters service immediately,

(3) the completion of a (high or low) priority job without starting a new job because the queue is empty.

Hence we find:

$$
\begin{aligned}
&\left(\Lambda^h + \Lambda^l + \overline{\mu}\left(\overline{s}^h, \overline{s}^l\right)\right) P_{n,m}\left(\overline{0}, \overline{s}^h, \overline{0}, \overline{s}^l, \overline{0}\right) = \\
&\sum_{i=1}^{N^h} \lambda_i^h P_{n-1,m}\left(\overline{0}, \overline{s}^h - \overline{e}_i^h, \overline{0}, \overline{s}^l, \overline{0}\right) + \sum_{i=1}^{N^l} \lambda_i^l P_{n,m-1}\left(\overline{0}, \overline{s}^h, \overline{0}, \overline{s}^l - \overline{e}_i^l, \overline{0}\right) \\
&+ \sum_{i=1}^{N^h} \left(s_i^h + 1\right) \mu_i^h P_{n+1,m}\left(\overline{0}, \overline{s}^h + \overline{e}_i^h, \overline{0}, \overline{s}^l, \overline{0}\right) + \sum_{i=1}^{N^l} \left(s_i^l + 1\right) \mu_i^l P_{n,m+1}\left(\overline{0}, \overline{s}^h, \overline{0}, \overline{s}^l + \overline{e}_i^l, \overline{0}\right)
\end{aligned}
$$

In the next sections, we will show how we can solve these equilibrium equations thereby obtaining the exact system state probabilities. We will address the areas I, II and III in Section 3 and Section 4.

3 System states with high priority items in queue ($n > k$).

In this section we focus on area I, so there is at least one high priority item in the queue.

3.1 Reducing the set of equations

First, we note that we can rewrite these state probabilities by conditioning on the total number of high priority items in the queue. So, $P_{n,m}\left(\overline{w}^h, \overline{s}^h, \overline{w}^l, \overline{0}, \overline{r}^l\right) = P'_{n,m}\left(\left|\overline{w}^h\right|, \overline{s}^h, \overline{w}^l, \overline{0}, \overline{r}^l\right) \times$ Pr{distribution of high priority items in queue over subclasses $= \overline{w}^h$ | number of high priority items in queue$= \left|\overline{w}^h\right|$}. Given the total number of high priority items in the queue, the items are distributed over the item subclasses according to a multinomial distribution, $\left|\overline{w}^h\right|! \prod_{i=1}^{N^h} \frac{\left(a_i^h\right)^{w_i^h}}{w_i^h!}$. The expression $P'_{n,m}\left(\left|\overline{w}^h\right|, \overline{s}^h, \overline{w}^l, \overline{0}, \overline{r}^l\right)$ describes the probability that (1) the distribution of high priority items in service is given by the vector $\overline{s}^h$, (2) the low priority items in queue are given by the vectors $\overline{w}^l$ and $\overline{r}^l$, (3) no low priority items are in service, and (4) the total number of high priority items in the queue equals $\left|\overline{w}^h\right|$. Because it holds that $\left|\overline{w}^h\right| = n - k$, we see that this expression is in fact independent of $\left|\overline{w}^h\right|$. Therefore we can omit the parameter $\left|\overline{w}^h\right|$ and rewrite the equilibrium equations for the states with $n > k$ using the product form:

$$P_{n,m}\left(\overline{w}^h, \overline{s}^h, \overline{w}^l, \overline{0}, \overline{r}^l\right) = \left|\overline{w}^h\right|! \prod_{i=1}^{N^h} \frac{\left(a_i^h\right)^{w_i^h}}{w_i^h!} P'_{n,m}\left(\overline{s}^h, \overline{w}^l, \overline{0}, \overline{r}^l\right). \tag{5}$$

In this way, we reduce the number of state probabilities to be solved.

We substitute the relation (5) in equation (1) and we divide both sides by $k\mu^h$ to obtain equations that can be transformed in matrix form. This yields:

$$\begin{aligned}
&\left(1 + \overline{\delta}\left(\overline{s}^h, \overline{0}\right) + \rho^h + \gamma\rho^l\right) \left|\overline{w}^h\right|! \prod_{i=1}^{N^h} \frac{\left(a_i^h\right)^{w_i^h}}{w_i^h!} P'_{n,m}\left(\overline{s}^h, \overline{w}^l, \overline{0}, \overline{r}^l\right) \\
&= \rho^h \sum_{i=1}^{N^l} a_i^h \left(\left|\overline{w}^h\right| - 1\right)! \frac{\left(a_i^h\right)^{w_i^h - 1}}{\left(w_i^h - 1\right)!} \prod_{\substack{j=1 \\ j \neq i}}^{N^h} \frac{\left(a_j^h\right)^{w_j^h}}{w_j^h!} P'_{n-1,m}\left(\overline{s}^h, \overline{w}^l, \overline{0}, \overline{r}^l\right) \\
&+ \gamma\rho^l \sum_{i=1}^{N^l} a_i^l \left|\overline{w}^h\right|! \prod_{j=1}^{N^h} \frac{\left(a_j^h\right)^{w_j^h}}{w_j^h!} P'_{n,m-1}\left(\overline{s}^h, \overline{w}^l - e_i^l, \overline{0}, \overline{r}^l\right) \\
&+ \sum_{i=1}^{N^l} \sum_{j=1}^{N^l} \left(s_i^h + 1 - e_{ij}^h\right) \mu_i^h \frac{w_j^h + 1}{\left|\overline{w}^h\right| + 1} \frac{a_j^h \left(\left|\overline{w}^h\right| + 1\right)}{w_j^h + 1} \left|\overline{w}^h\right|! \prod_{q=1}^{N^h} \frac{\left(a_q^h\right)^{w_q^h}}{w_q^h!} P'_{n+1,m}\left(\overline{s}^h + e_i^h - e_j^h, \overline{w}^l, \overline{0}, \overline{r}^l\right)
\end{aligned}$$

where $\gamma = \frac{\mu^l}{\mu^h}$.

Because we use the product form (5), we have to sum up all the equations for the state probabilities satisfying $\sum w_i^h = \left|\overline{w}^h\right|$. Taking into account that $\sum a_i^h = 1$, we can write all equilibrium equation with the same n and m in a matrix form:

$$\begin{aligned}
&\left(\left(1 + \rho^h + \gamma\rho^l\right)\mathbf{I} + \overline{\boldsymbol{\delta}}^h\right) \mathbf{P}_{n,m}\left(\overline{w}^l, \overline{0}, \overline{r}^l\right) = \rho^h \mathbf{P}_{n-1,m}\left(\overline{w}^l, \overline{0}, \overline{r}^l\right) \\
&+ \gamma\rho^l \sum_{j=1}^{N^l} a_j \mathbf{P}_{n,m-1}\left(\overline{w}^l - e_j, \overline{0}, \overline{r}^l\right) + A\mathbf{P}_{n+1,m}\left(\overline{w}^l, \overline{0}, \overline{r}^l\right)
\end{aligned} \tag{6}$$

where $\mathbf{P}_{n+1,m}\left(\overline{w}^l,\overline{0}\right)$ are vectors containing probabilities $P'_{n,m}\left(\overline{s}^h,\overline{w}^l,\overline{0},\overline{r}^l\right)$ as components. The dimension of vectors $\mathbf{P}_{n+1,m}\left(\overline{w}^l,\overline{0}\right)$ is equal to amount of different server states given that all servers are occupied with hp items, i.e. this dimension is equal to $d\left(N^h,k\right)\left(\sum_{i=0}^{k} d\left(N^l,i\right)\right)$, with $d(x,y)=\binom{x+y+1}{y}$. $\overline{\delta}^h$ and A are linear operators on a $d\left(N^h,k\right)\left(\sum_{i=0}^{k} d\left(N^l,i\right)\right)$ – dimensional linear space:

$$\begin{aligned}\overline{\delta}^h\mathbf{P}_{n+1,m}\left(\overline{w}^l,\overline{0}\right)\left[\overline{s}^h,\overline{r}^l\right] &= \left(1+\overline{\delta}\left(\overline{s}^h,\overline{0}\right)+\rho^h+\gamma\rho^l\right)\mathbf{P}_{n+1,m}\left(\overline{w}^l,\overline{0}\right)\left[\overline{s}^h,\overline{r}^l\right]\\ A\mathbf{P}_{n+1,m}\left(\overline{w}^l,\overline{0}\right)\left[\overline{s}^h,\overline{r}^l\right] &= \sum_{i=1}^{N^l}\sum_{j=1}^{N^l}\left(s_i^h+1-e_{ij}\right)\left(1+\delta_i^h\right)a_j^h\mathbf{P}_{n+1,m}\left(\overline{w}^l,\overline{0}\right)\left[\overline{s}^h+e_i-e_j,\overline{r}^l\right].\end{aligned}$$

Solving this matrix equation we can find all state probabilities with $(n>k)$. To solve it, it is worth noticing that $\overline{r}^l$ only serves as an index, where equations with different indices $\overline{r}^l$ are decoupled In the next lemma the structure of the solution of this equation is explained.

Lemma 1
Define the matrix-function $\mathbf{Z}\left(\rho^h,\rho^l,\gamma;\xi\right)$ *as the solution of*

$$\left(\left(1+\rho^h+\gamma\rho^l\right)\mathbf{I}+\overline{\delta}^h\right)=\rho^h\mathbf{Z}+\gamma\rho^l\xi+A\mathbf{Z}^{-1},\quad \left|\sigma\left(\mathbf{Z}\right)\right|>1 \tag{7}$$

Then

$$\begin{aligned}&P_{n,m}\left(\overline{w}^h,\overline{s}^h,\overline{w}^l,\overline{0},\overline{r}^l\right) \qquad (8)\\ &=\left|\overline{w}^h\right|!\prod_{i=1}^{N^h}\frac{\left(a_i^h\right)^{w_i^h}}{w_i^h!}\left(\prod_{j=1}^{N^l}\frac{\left(a_j^l\right)^{w_j^l}}{w_j^l!}\right)\left(\frac{d}{d\xi}\right)^{\left|\overline{w}^l\right|}\left[\left(\mathbf{Z}^{-1}\left(\xi\right)\right)^{n-k}\mathbf{C}\left(\xi\right)\right]_{\xi=0}\left[\overline{s}^h,\overline{r}^l\right]\end{aligned}$$

satisfies all equations for $m\geq 0$, $n>k$.

Note that $\sum_{\overline{w}^h,\overline{w}^l}P\left(\overline{w}^h,\overline{s}^h,\overline{w}^l,\overline{0},\overline{r}^l\right)=\left[\left(\mathbf{Z}^{-1}\left(\xi\right)\right)^{n-k}\mathbf{C}\left(\xi\right)\right]_{\xi=1}\left[\overline{s}^h,\overline{r}^l\right]$. *The notation* $\left[\overline{s}^h,\overline{r}^l\right]$ *in the right hand side refers to the indicated vector component.*

Details of the proof are given in appendix 1.

The probabilities of the system states constructed in this section have a differential form, therefore we will need derivatives of the matrix $\mathbf{Z}$. To find these derivatives is not an easy task since we can not derive an analytical form of the matrix $\mathbf{Z}$, but we can use the equation (7) to find such derivatives iteratively. Details are given in appendix 2.

4 System states with no high priority items in queue $(n\leqslant k)$.

In this section we describe the solution of the equilibrium equations for the states with only low priority items in the queue $(n\leq k)$.

4.1 Reducing the set of equations

As in the previous section, we can reduce the set of equations by writing the state probabilities in product form, conditioning on the total number of low priority items in the queue $\left|\overline{w}^l\right|$. Given this total number, the number of jobs in the queue per low priority subclass has a multinominal distribution with parameters $\left|\overline{w}^l\right|$ and $a_i^l=\lambda_i^l/\Lambda^l$, $i=1\ldots N^l$. So we can write:

$$P_{n,m}\left(\overline{0},\overline{s}^h,\overline{w}^l,\overline{s}^l,\overline{r}^l\right)=\left|\overline{w}^l\right|!\prod_{i=1}^{N^l}\frac{\left(a_i^l\right)^{w_i^l}}{w_i^l!}P''_{n,m}\left(\overline{s}^h,\overline{s}^l,\overline{r}^l\right).$$

Note that we can omit two parameters, namely the number of high priority items in the queue per subclass (these are always zero) and the number of low priority items in the queue $\left|\overline{w}^l\right|$. For the latter, it holds that $\left|\overline{w}^l\right|=m-\left|\overline{s}^l\right|-\left|\overline{r}^l\right|$.

Again, we substitute this product form into the equations (2) for $n<k$, $m>k-n$, and next we sum up the equations for all system states having the total number of (non-postponed) low priority items in the queue equal to $\left|\overline{w}^l\right|$. After dividing the resulting equations by $k\mu^h$, we obtain the following equations for $P''_{n,m}(\cdot,\cdot)$:

$$\begin{aligned}
&\left(1+\overline{\delta}\left(\overline{s}^h,\overline{s}^l\right)+\rho^h+\gamma\rho^l\right)\left(\sum_{i=1}^{N^l}a_i^l\right)^{\left|\overline{w}^l\right|}P''_{n,m}\left(\overline{s}^h,\overline{s}^l,\overline{r}^l\right)=\\
&\rho^h\left(\sum_{i=1}^{N^l}a_i^l\right)^{\left|\overline{w}^l\right|}\sum_{i=1}^{N^h}a_i^h\sum_{j=1}^{N^l}\frac{s_j^l+1}{\left|s^l\right|+1}P''_{n-1,m}\left(\overline{s}^h-e_i,\overline{s}^l+e_j,\overline{r}^l-e_j\right)\\
&+\left(\sum_{i=1}^{N^l}a_i^l\right)^{\left|\overline{w}^l\right|}\gamma\rho^l P''_{n,m-1}\left(\overline{s}^h,\overline{s}^l,\overline{r}^l\right)\\
&+I\left(\left|\overline{r}^l\right|=0\right)\left(\sum_{i=1}^{N^l}a_i^l\right)^{\left|\overline{w}^l\right|}\sum_{i=1}^{N^h}\sum_{j=1}^{N^l}\left(s_i^h+1\right)\left(1+\delta_i^h\right)a_j^l P''_{n+1,m}\left(\overline{s}^h+e_i^h,\overline{s}^l-e_j^l,\overline{0}\right)\\
&+I\left(\left|\overline{r}^l\right|=0\right)\left(\sum_{i=1}^{N^l}a_i^l\right)^{\left|\overline{w}^l\right|}\sum_{i=1}^{N^l}\sum_{j=1}^{N^l}a_i^l\left(s_i+1-e_{ij}^l\right)\left(1+\delta_i^l\right)P''_{n,m+1}\left(\overline{s}^h,\overline{s}^l+e_i^l-e_j^l,\overline{0}\right)\\
&+\left(\sum_{i=1}^{N^l}a_i^l\right)^{\left|\overline{w}^l\right|}\sum_{i=1}^{N^h}\sum_{j=1}^{N^l}\left(s_i^h+1\right)\left(1+\delta_i^h\right)\frac{r_j^l+1}{\left|r^l\right|+1}a_j^l P''_{n+1,m}\left(\overline{s}^h+e_i^h,\overline{s}^l-e_j^l,\overline{r}^l+e_j^l\right)\\
&+\left(\sum_{i=1}^{N^l}a_i^l\right)^{\left|\overline{w}^l\right|}\sum_{i=1}^{N^l}\sum_{j=1}^{N^l}\left(s_i^l+1-e_{ij}^l\right)\left(1+\delta_i^l\right)\frac{r_j^l+1}{\left|r^l\right|+1}P''_{n,m+1}\left(\overline{s}^h,\overline{s}^l+e_i^l-e_j^l,\overline{r}^l+e_j^l\right)
\end{aligned}\tag{9}$$

where $\sum_{i=1}^{N^l}a_i^l=1$, and $\gamma=\frac{\mu^l}{\mu^h}$.

Again, this results in a matrix equation for $n<k$, $m\leq k-n$, namely:

$$\mathbf{D}_{n,m}\mathbf{P}_{n,m}=\mathbf{F}_{n,m}\mathbf{P}_{n-1,m}+\mathbf{E}_{n,m}\mathbf{P}_{n,m-1}+\mathbf{B}_{n,m}\mathbf{P}_{n+1,m}+\mathbf{G}_{n,m}\mathbf{P}_{n,m+1}$$

where the operators $D_{n,m}$, $F_{n,m}$, $E_{n,m}$, $B_{n,m}$ and $G_{n,m}$ on the vectors $\zeta\left[\overline{s}^h,\overline{s}^l\right]$ are respectively defined as:

$$\mathbf{D}_{n,m}\zeta\left[\overline{s}^h,\overline{s}^l,\overline{r}^l\right]\stackrel{def}{=}\begin{cases}\left(1+\overline{\delta}\left(\overline{s}^h,\overline{s}^l\right)+\rho^h+\gamma\rho^l\right)\zeta\left[\overline{s}^h,\overline{s}^l,\overline{r}^l\right], & n+m\geq k\\ \left(\frac{n+m}{k}+\overline{\delta}\left(\overline{s}^h,\overline{s}^l\right)+\rho^h+\gamma\rho^l\right)\zeta\left[\overline{s}^h,\overline{s}^l,\overline{0}\right], & n+m<k\end{cases}$$

$$\mathbf{F}_{n,m}\zeta\left[\overline{s}^h,\overline{s}^l,\overline{r}^l\right]\stackrel{def}{=}\begin{cases}\rho^h\sum_{i=1}^{N^h}\sum_{j=1}^{N^l}\frac{s_j^l+1}{\left|s^l\right|+1}a_i^h\zeta\left[\overline{s}^h-e_i^h,\overline{s}^l+e_j^l,\overline{r}^l-e_j^l\right], & n+m>k\\ \rho^h\sum_{i=1}^{N^h}a_i^h\zeta\left[\overline{s}^h-e_i^h,\overline{s}^l\right], & n+m\leq k\end{cases}$$

$$\mathbf{E}_{n,m}\zeta\left[\overline{s}^h,\overline{s}^l,\overline{r}^l\right]\stackrel{def}{=}\begin{cases}\gamma\rho^l\zeta\left[\overline{s}^h,\overline{s}^l,\overline{r}^l\right], & n+m>k\\ \gamma\rho^l\sum_{i=1}^{N^l}a_i^l\zeta\left[\overline{s}^h,\overline{s}^l-e_i^l,\overline{0}\right], & n+m\leq k\end{cases}$$

$$\mathbf{B}_{n,m}\zeta\left[\overline{s}^h,\overline{s}^l,\overline{r}^l\right] \stackrel{def}{=} \begin{cases} \frac{1}{k}\sum\limits_{i=1}^{N^h}\sum\limits_{j=1}^{N^l}\left(s_i^h+1\right)\left(1+\delta_i^h\right)\frac{r_j^l+1}{|r^l|+1}\zeta\left[\overline{s}^h+e_i^h,\overline{s}^l-e_j^l,\overline{r}^l+e_j^l\right], & n+m\geq k \\ \frac{1}{k}\sum\limits_{i=1}^{N^h}\sum\limits_{j=1}^{N^l}a_j^l\left(s_i^h+1\right)\left(1+\delta_i^h\right)\zeta\left[\overline{s}^h+e_i^h,\overline{s}^l-e_j^l,\overline{0}\right], & n+m\geq k,\ \overline{r}^l=\overline{0} \\ \frac{1}{k}\sum\limits_{i=1}^{N^h}\left(s_i^h+1\right)\left(1+\delta_i^h\right)\zeta\left[\overline{s}^h+e_i^h,\overline{s}^l,\overline{0}\right], & n+m<k \end{cases}$$

$$\mathbf{G}_{n,m}\zeta\left[\overline{s}^h,\overline{s}^l,\overline{r}^l\right] \stackrel{def}{=} \begin{cases} \frac{1}{k}\sum\limits_{i=1}^{N^l}\sum\limits_{j=1}^{N^l}\left(s_i^l+1-e_{ij}^l\right)\left(1+\delta_i^l\right)\frac{r_j^l+1}{|r^l|+1}\zeta\left[\overline{s}^h,\overline{s}^l+e_i^l-e_j^l,\overline{r}^l+e_j^l\right], & n+m\geq k \\ \frac{1}{k}\sum\limits_{i=1}^{N^l}\sum\limits_{j=1}^{N^l}a_i^l\left(s_i+1-e_{ij}^l\right)\left(1+\delta_i^l\right)\zeta\left[\overline{s}^h,\overline{s}^l+e_i^l-e_j^l,\overline{0}\right], & n+m\geq k,\ \overline{r}^l=\overline{0} \\ \frac{1}{k}\sum\limits_{i=1}^{N^l}\left(s_i+1\right)\left(1+\delta_i^l\right)\zeta\left[\overline{s}^h,\overline{s}^l+e_i^l,\overline{0}\right], & n+m<k \end{cases}$$

The equilibrium equations (3) for $n=k$ have a similar form, with the only difference that these equations include the probabilities of the system states with one high priority item in the queue ($n=k+1$), which are equal to $\frac{1}{|\overline{w}^l|!}\left(\frac{d}{d\xi}\right)^{|\overline{w}^l|}\left[\mathbf{Z}^{-1}(\xi)\,\mathbf{C}(\xi)\right]$. So, we can aggregate all equations for states with no high priority items in queue ($n\leq k$) and write them in matrix form. To this end, we write all vectors $\mathbf{P}_{n,m}$ for the states with the same number of items in the system (i.e. $n+m$) into one vector $\mathbb{P}_{\hat{t}}$, with $\hat{t}=n+m$. For example, then the equations for the states with more items in the system than the number of servers ($\hat{t}>k$) can be written as:

$$\begin{pmatrix} \mathbf{D}_{0,m} & & & \mathbf{0} \\ & \mathbf{D}_{1,m-1} & & \\ & & \ddots & \\ \mathbf{0} & & & \mathbf{D}_{k,m-k} \end{pmatrix}\begin{pmatrix} \mathbf{P}_{0,m} \\ \mathbf{P}_{1,m-1} \\ \vdots \\ \mathbf{P}_{k,m-k} \end{pmatrix}$$

$$=\begin{pmatrix} \mathbf{E}_{0,m} & & & & \mathbf{0} \\ \mathbf{F}_{1,m-1} & \mathbf{E}_{1,m-1} & & & \\ & \ddots & \ddots & & \\ \mathbf{0} & & \mathbf{F}_{k,m-k} & \mathbf{E}_{k,m-k} & \end{pmatrix}\begin{pmatrix} \mathbf{P}_{0,m-1} \\ \mathbf{P}_{1,m-2} \\ \vdots \\ \mathbf{P}_{k,m-k-1} \end{pmatrix}$$

$$+\begin{pmatrix} \mathbf{G}_{0,m} & \mathbf{B}_{0,m} & & \mathbf{0} \\ & \mathbf{G}_{1,m-1} & \ddots & \\ & & \ddots & \mathbf{B}_{k-1,m-k+1} \\ \mathbf{0} & & & \mathbf{G}_{k,m-k} \end{pmatrix}\begin{pmatrix} \mathbf{P}_{0,m+1} \\ \mathbf{P}_{1,m} \\ \vdots \\ \mathbf{P}_{k,m-k+1} \end{pmatrix}$$

$$+\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{B}_{k,m-k} \end{pmatrix} m!\left(\frac{d}{d\xi}\right)^m\left[\mathbf{Z}^{-1}(\xi)\,\mathbf{C}(\xi)\right]_{\xi=0}$$

The equations for $n+m\leq k$ have the same form, but they do not include the inhomogeneous term.

Rewriting these equations in a matrix form, where the vector $\mathbb{P}_{\hat{t}}$ is composed of the vectors $\mathbf{P}_{n,m-n}$ we obtain:

$$\mathbb{D}_{\hat{t}}\mathbb{P}_{\hat{t}}=\mathbb{F}_{\hat{t}}\mathbb{P}_{\hat{t}-1}+\mathbb{G}_{\hat{t}}\mathbb{P}_{\hat{t}+1}+\mathbb{B}_{\hat{t}}\frac{1}{(\hat{t}-k)!}\left(\frac{d}{d\xi}\right)^{\hat{t}-k}\left[\mathbf{Z}^{-1}(\xi)\,\mathbf{C}(\xi)\right]_{\xi=0},\quad \hat{t}\geq k \tag{10}$$

$$\mathbb{D}_{\hat{t}}\mathbb{P}_{\hat{t}}=\mathbb{F}_{\hat{t}}\mathbb{P}_{\hat{t}-1}+\mathbb{G}_{\hat{t}}\mathbb{P}_{\hat{t}+1},\quad \hat{t}<k \tag{11}$$

where the matrices $\mathbb{D}_{\hat{t}}$, $\mathbb{F}_{\hat{t}}$ and $\mathbb{G}_{\hat{t}}$ are fixed for $\hat{t}\geq k$ and depend on $\hat{t}$ for $\hat{t}<k$.

The dimension of the equations in (10) does not depend on $\hat{t}$. It can be derived from the number of all combinations of $N^h + N^l$ items on k servers including all combinations of j postponed items $j \leq n \leq k$. We find:

$$\dim(\mathcal{L}_{"\hat{t}>k,n\leq k"}) = \sum_{i=0}^{k} \left[\sum_{n=i}^{k} \binom{N^h + n - 1}{n} \binom{N^l + k - n - 1}{k - n} \right] \binom{N^l + i - 1}{i}. \quad (12)$$

The dimension of the equations (11) depends on $\hat{t}$ and should be equal to the number of all combinations of $N^h + N^l$ items on $\hat{t}$ servers if $\hat{t} \leq k$, i.e.

$$\dim(\mathcal{L}_{"\hat{t}\leq k"}) = \binom{N^h + N^l + \hat{t} - 1}{\hat{t}}, \quad (13)$$

So, we now have two systems of second order linear difference equations. One system (for $\hat{t} > k$) is an inhomogeneous system with a fixed dimension and with fixed coefficients. The second system (for $\hat{t} < k$) has coefficients depending on $\hat{t}$. Also, the system dimension depends on t. In the next two sections we describe how we can construct the solution of these two systems.

4.2 States with only low priority items in queue ($\hat{t} \geq k$, $n \leq k$).

The probabilities of the system states having only low priority items in queue satisfy the system of linear inhomogeneous difference equations of second order with fixed coefficients (10). However, the inhomogeneous term has a differential form, therefore the standard procedure of solving the inhomogeneous equations (solution of homogeneous + partial solution of inhomogeneous) is difficult to apply. Therefore we will look for the solution in a differential form $\mathbb{P}_{t+k} = \frac{1}{t!}\left(\frac{d}{d\xi}\right)^t v(\xi)_{\xi=0}$ where $t = \hat{t} - k$. The substitution of this solution into the equation (10) gives:

$$\mathbb{D}\frac{1}{t!}\left(\frac{d}{d\xi}\right)^t v(\xi)_{\xi=0} = \mathbb{F}\frac{1}{(t-1)!}\left(\frac{d}{d\xi}\right)^{t-1} v(\xi)_{\xi=0} + \mathbb{G}\frac{1}{(t+1)!}\left(\frac{d}{d\xi}\right)^{t+1} v(\xi)_{\xi=0}$$
$$+ \mathbb{B}\frac{1}{t!}\left(\frac{d}{d\xi}\right)^t \left[\mathbf{Z}^{-1}(\xi)\,\mathbf{C}(\xi)\right]_{\xi=0}, \quad t > 0$$

Here we can apply the following equations: $\left(\frac{d}{dx}\right)^t (xf(x))_{x=0} = t\left[\left(\frac{d}{dx}\right)^{t-1} f(x)\right]_{x=0}$ and $\left(\frac{d}{dx}\right)^t \left(x^2 f(x)\right)_{x=0} = t(t-1)\left[\left(\frac{d}{dx}\right)^{t-2} f(x)\right]_{x=0}$ that can be easily proved as shown in the proof of lemma 1 (Appendix 1). These two equations allow us to remove the derivatives from equation (10) and to obtain a new expression of the function $v(\xi)$ for any $t > 0$:

$$\frac{1}{(t+1)!}\left(\frac{d}{d\xi}\right)^{t+1} \left[\xi\mathbb{D}v(\xi) - \xi^2\mathbb{F}v(\xi) - \mathbb{G}v(\xi) - \xi\mathbb{B}\mathbf{Z}^{-1}(\xi)\,\mathbf{C}(\xi)\right]_{\xi=0} = 0, \quad t > 0. \quad (14)$$

The function $\mathbf{C}(\xi)$ can be expressed as a part of the vector-function $v(\xi)$, which corresponds to the states with k high priority items in the system $v_k(\xi)$, i.e. $\mathbf{C}(\xi) = v_k(\xi)$.

The right part of equation (14) should be a function which becomes zero for any $t > 0$, i.e. a linear function. Hence, we obtain another expression for the vector-function $v(\xi)$, that does not contain derivatives, but that contains unknown vectors C_1 and C_2:

$$\xi\mathbb{D}v(\xi) - \xi^2\mathbb{F}v(\xi) - \mathbb{G}v(\xi) - \xi\mathbb{B}\mathbf{Z}^{-1}(\xi)\,v_k(\xi) = C_1\xi + C_2 \quad (15)$$

or

$$\mathbb{H}(\xi)\, v(\xi) = C_1\xi + C_2 \tag{16}$$

The constants C_1 and C_2 can be easily expressed via the probability states $\mathbb{P}_k$, and $\mathbb{P}_{k-1}$, i.e. we have from equation (15) that for $\xi = 0$,

$$-\mathbb{G}v(0) = C_2$$

and recalling that $v(0) = \mathbb{P}_k$, we obtain an equation for C_2:

$$C_2 = -\mathbb{G}\mathbb{P}_k.$$

Next, we can take the derivative of equation (15) in point $\xi = 0$ and obtain

$$\mathbb{D}v(0) - \mathbb{G}v'(0) - \mathbb{B}\mathbf{Z}^{-1}(0)\, v_k(0) = C_1,$$

The left hand side terms of the last equation are equivalent to some terms of equation (10) for $\hat{t} = k$ if we take into account that $\mathbb{P}_k = v(0)$, $\mathbb{P}_{k+1} = v'(0)$ and that the matrices $\mathbb{D}_k$, $\mathbb{G}_k$, $\mathbb{B}_k$ are equivalent to $\mathbb{D}$, $\mathbb{G}$ and $\mathbb{B}$, respectively. Then we find:

$$C_1 = \mathbb{D}_k\mathbb{P}_k - \mathbb{G}_k\mathbb{P}_{k+1} - \mathbb{B}_k\mathbf{Z}^{-1}(0)\, v_k(0) = \mathbb{F}_k\mathbb{P}_{k-1}$$

So, we have

$$C_1 = \mathbb{F}_k\mathbb{P}_{k-1}.$$

In this way, we have defined a function $v(\xi)$ given the probability vectors $\mathbb{P}_k$ and $\mathbb{P}_{k-1}$ and next all probability vectors $\mathbb{P}_{\hat{t}}$ for $\hat{t} = k+1 \ldots \infty$ follow from $\mathbb{P}_k$ and $\mathbb{P}_{k-1}$. However, an essential piece of information has not been used upto now. It is clear that we are looking for decaying solutions $\mathbb{P}_{\hat{t}}$ for $\hat{t} \longrightarrow \infty$. As a consequence $v(\xi)$ should be analytic on a circle with radius $1+\varepsilon$ for some $\varepsilon > 0$. Due to (15) extra conditions have to be satisfied at points ξ inside this circle where $H(\xi)$ is singular. It turns out that there are several such points in general. For example, $\xi = 0$ and $\xi = 1$ are points of this type. It is easy to check that in case $\xi = 0$ any vector with 0 entries whenever $|r^l| > 0$ is in the null space of $\mathbb{G}$. Using the the equilibrium property for subsystems with $\overline{s}^h, \overline{s}^l, \overline{r}^l$ fixed m arbitrary it is not difficult to check that $\mathbf{1}^t = (1, \ldots, 1)$ is a left eigenvalue of $\mathbb{H}(1)$ for the eigenvalue 0. In the next section we shall show that the decay requirement boils down a relation between the initial condition $\mathbb{P}_{k+1}$ and $\mathbb{P}_k$ of the following type:

$$\mathbb{P}_{k+1} = \mathbb{Q}_k\mathbb{P}_k.$$

In the next section it will be shown that this can be done by using a direct method without reference to singular points of $H(\xi)$. This result is crucial in the case with no queue at all.

4.3 Decay of $\mathbb{P}_{\hat{t}}$ for $\hat{t} \longrightarrow \infty$.

From (15) it follows by differentiating $t \geq 2$ times with respect to ξ that:

$$\binom{t}{0}\mathbb{H}(0)\frac{d^t}{d\xi^t}v(0) + \binom{t}{1}\mathbb{H}'(0)\frac{d^{t-1}}{d\xi^{t-1}}v(0) + \cdots + \binom{t}{t}\frac{d^t}{d\xi^t}\mathbb{H}(0)\, v(0) = 0,$$

We can also write this in the form

$$\sum_{i=0}^{t} h_i\mathbb{P}_{k+m-i} = 0,$$

where $\mathbb{P}_{k+i} = \frac{1}{i!}\frac{d^i}{d\xi^i} v(0)$ and $h_i = \frac{1}{i!}\frac{d^i}{d\xi^i}\mathbb{H}(0)$.

An important consequence of this equation is that

$$\mathbb{P}_{k+1} = \mathbf{Q}_k^t \mathbb{P}_k + \mathbf{\Omega}_k^t \mathbb{P}_{k+t} \tag{17}$$

We will derive this result shortly. First we mention that this has the implication that for decaying solutions $\mathbb{P}_{k+t}$ for $t \longrightarrow \infty$ if $\mathbf{\Omega}_k^t$ remains bounded:

$$\mathbb{P}_{k+1} = \mathbb{Q}_k \mathbb{P}_k \text{ with } \mathbb{Q}_k = \lim_{t \longrightarrow \infty} \mathbf{Q}_k^t$$

Let us now show how the matrices $\mathbf{Q}_k^t$ and $\mathbf{\Omega}_k^t$ can be determined.

Using backward recursion from t to $k+1$ we can show that for any t and t^* $(t > t^*)$ a relation

$$\mathbf{\Theta}_0^{t^*} \mathbb{P}_{t+k} + \sum_{i=1}^{t^*} \mathbf{\Theta}_i^{t^*} \mathbb{P}_{k+t^*-i} = 0 \tag{18}$$

is valid. That is:

For $t^* = t-1$ it is clear from the equation for $\mathbb{P}_{t+k}$ that we can take $\mathbf{\Theta}_i^t = h_i$.

For $t^* < t-1$ we have the equation derived in the previous induction step and also the original equation for $\mathbb{P}_{t^*+k}$:

$$\begin{aligned} \mathbf{\Theta}_0^{t^*+1} \mathbb{P}_{t+k} + \mathbf{\Theta}_1^{t^*+1} \mathbb{P}_{k+t^*} + \cdots + \mathbf{\Theta}_{t^*+1}^{t^*+1} \mathbb{P}_k &= 0 \\ h_0 \mathbb{P}_{k+t^*} + \cdots + h_{t^*} \mathbb{P}_k &= 0 \end{aligned}$$

Then multiplying the first equation by $h_0 \left(\mathbf{\Theta}_1^{t^*+1}\right)^{-1}$ and taking the difference we obtain

$$\begin{aligned} h_0 \left(\mathbf{\Theta}_1^{t^*+1}\right)^{-1} \mathbf{\Theta}_0^{t^*+1} \mathbb{P}_{t+k} &+ \left(h_0 \left(\mathbf{\Theta}_1^{t^*+1}\right)^{-1} \mathbf{\Theta}_2^{t^*+1} - h_1\right) \mathbb{P}_{k+t^*-1} \\ &+ \cdots \\ &+ \left(h_0 \left(\mathbf{\Theta}_1^{t^*+1}\right)^{-1} \mathbf{\Theta}_{t^*+1}^{t^*+1} - h_{t^*}\right) \mathbb{P}_k = 0 \end{aligned}$$

or in other terms

$$\mathbf{\Theta}_0^{t^*} \mathbb{P}_{k+t} + \mathbf{\Theta}_1^{t^*} \mathbb{P}_{k+t^*-1} + \cdots + \mathbf{\Theta}_{t^*}^{t^*} \mathbb{P}_k = 0$$

where the matrices $\mathbf{\Theta}_i^{t^*}$ are equal to

$$\begin{aligned} \mathbf{\Theta}_0^{t^*} &= h_0 \left(\mathbf{\Theta}_1^{t^*+1}\right)^{-1} \mathbf{\Theta}_0^{t^*+1} \\ \mathbf{\Theta}_i^{t^*} &= \left(h_0 \left(\mathbf{\Theta}_1^{t^*+1}\right)^{-1} \mathbf{\Theta}_{i+1}^{t^*+1} - h_i\right), \quad i = 1, \ldots, t^* \end{aligned}$$

So, we have shown that for any t and t^* $(t > t^*)$ the relation (18) is valid. Next, taking $t^* = 2$ we obtain

$$\mathbf{\Theta}_0^2 \mathbb{P}_{t+k} + \mathbf{\Theta}_1^2 \mathbb{P}_{k+1} + \mathbf{\Theta}_2^2 \mathbb{P}_k = 0$$

which is the desired result if we identify $\mathbf{\Omega}_k^t = -\left(\mathbf{\Theta}_1^2\right)^{-1} \mathbf{\Theta}_0^2$ and $\mathbf{Q}_k^t = -\left(\mathbf{\Theta}_1^2\right)^{-1} \mathbf{\Theta}_2^2$. Herewith the relation (17) is shown.

Note that from a computational point of view the matrices $\mathbf{Q}_k^t$ and $\mathbf{\Omega}_k^t$ can be computed using a straightforward iteration procedure. In this iteration we check the boundedness of $\mathbf{\Omega}_k^t$

and numerical evidence shows that this is the case in all cases we considered, moreover we found that

$$\lim_{t\longrightarrow\infty}\left\|\Omega_k^t\right\| \approx \omega\,(1-\rho)$$

where ω is some constant depending on other system parameters. Hence by taking t sufficiently large we then have a good approximation of the matrix $\mathbb{Q}_k$ that will play a role in the sequel. Furthermore, $\rho \longrightarrow 1$ is not a problem for the convergence in the iterations.

In the computation of performance measures we shall also need information about the function $v(\xi)$ at $\xi = 1$. As an intermezzo we describe in the next subsection how such information can now easily be obtained.

4.4 On the vectors $v\,(1)$, $v'\,(1)$ and $v''\,(1)$.

Let us to discuss now the procedure to find $v\,(1)$, $v'\,(1)$ and $v''\,(1)$, which we need for the performance characteristics. This procedure is non-trivial due to singularity of the matrix $\mathbb{H}\,(1)$. Therefore we cannot find the vector $v\,(1)$ just by inverting the equation (15) at $\xi = 1$

$$\mathbb{H}\,(1)\,v\,(1) = \mathbb{F}_k\mathbb{P}_{k-1} - \mathbb{G}\mathbb{P}_k \tag{19}$$

We have to apply a procedure using the Taylor expansion. Namely we derive from (15)

$$\mathbb{H}\,(1)\,v'\,(1) + \mathbb{H}'\,(1)\,v\,(1) = \mathbb{F}_k\mathbb{P}_{k-1}. \tag{20}$$

In this equation we have to get rid of the term with $v'\,(1)$. It can be done if we project in (20) both sides onto the null-space of $\mathbb{H}\,(1)$. Hence, we multiply this equation by matrix Pr_1 constructed as:

$$\mathrm{Pr}_1 = \varepsilon\, S \begin{pmatrix} 1 & \\ & \mathbf{0} \end{pmatrix} S^{-1},$$

where S is matrix of eigenvectors of $\mathbb{H}\,(1)$ and the block with 1 on the diagonal corresponds to the zero eigenvalue and the block with 0 on the diagonal to non-zero eigenvalues; ε is some number > 0 that can be used for scaling purposes. It is easy to show that multiplication of matrices Pr_1 and $\mathbb{H}\,(1)$ gives zero:

$$\mathrm{Pr}_1\,\mathbb{H}\,(1) = \varepsilon\, S \begin{pmatrix} 1 & \\ & \mathbf{0} \end{pmatrix} S^{-1}\mathbb{H}\,(1) = \varepsilon\, S \begin{pmatrix} 1 & \\ & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \\ & * \end{pmatrix} S^{-1} = \mathbf{0}.$$

So, we have now a new system of linear equations

$$\left[\mathbb{H}\,(1) + \mathrm{Pr}_1\,\mathbb{H}'\,(1)\right] v\,(1) = (I + \mathrm{Pr}_1)\,\mathbb{F}_k\mathbb{P}_{k-1} - \mathbb{G}\mathbb{P}_k$$

where the matrix $[\mathbb{H}\,(1) + \mathrm{Pr}_1\,\mathbb{H}'\,(1)]$ turns out to be non-singular. In all our experiments it turned out that 0 is a single eigenvalue of $\mathbb{H}\,(1)$ and as we saw previously the corresponding left eigenvector is $\mathbf{1}^t = (1, ..., 1)$. As a consequence the matrix Pr_1 can be constructed as a matrix with elements of one row (any row) equal to 1.

$$\mathrm{Pr}_1 = \begin{pmatrix} 1 & \cdots & 1 \\ & \ddots & \\ 0 & \cdots & 0 \end{pmatrix}$$

In the same way we can find the derivatives $v'\,(1)$, $v''\,(1)$, etc. That is

$$\begin{aligned} \left[\mathbb{H}\,(1) + 2\,\mathrm{Pr}_1\,\mathbb{H}'\,(1)\right] v'\,(1) &= -\left[\mathbb{H}'\,(1) + \mathrm{Pr}_1\,\mathbb{H}''\,(1)\right] v\,(1) + \mathbb{F}_k\mathbb{P}_{k-1} \\ \left[\mathbb{H}\,(1) + 3\,\mathrm{Pr}_1\,\mathbb{H}'\,(1)\right] v''\,(1) &= -\left[2\mathbb{H}'\,(1) + 3\,\mathrm{Pr}_1\,\mathbb{H}''\,(1)\right] v'\,(1) - \left[\mathbb{H}''\,(1) + \mathrm{Pr}_1\,\mathbb{H}'''\,(1)\right] v\,(1) \end{aligned}$$

4.5 States with empty queue ($\hat{t} < k$).

The equilibrium equations for $\hat{t} < k$ do not have an inhomogeneous term:

$$\mathbb{D}_{\hat{t}}\mathbb{P}_{\hat{t}} = \mathbb{F}_{\hat{t}}\mathbb{P}_{\hat{t}-1} + \mathbb{G}_{\hat{t}}\mathbb{P}_{\hat{t}+1}$$

However, now the matrices $\mathbb{D}_{\hat{t}}$, $\mathbb{F}_{\hat{t}}$, $\mathbb{G}_{\hat{t}}$ depend on the value of $\hat{t}$. We can find $\mathbb{P}_{\hat{t}}$ for $\hat{t} < k$ using a recurrent expression. The complete solution can be represented as:

$$\mathbb{P}_{\hat{t}} = \mathbb{Q}_{\hat{t}-1}\mathbb{P}_{\hat{t}-1} = \mathbb{Q}_{\hat{t}-1}\mathbb{Q}_{\hat{t}-2}\cdots\mathbb{Q}_0\mathbb{P}_0$$

where $\mathbb{Q}_{\hat{t}}$ follows recursively from

$$\mathbb{Q}_{\hat{t}-1} = \left(\mathbb{D}_{\hat{t}} - \mathbb{G}_{\hat{t}}\mathbb{Q}_{\hat{t}}\right)^{-1}\mathbb{F}_{\hat{t}}, \quad \hat{t} = k, \ldots, 1$$

In this recurrent relation, the matrix $\mathbb{Q}_k$ was found in the previous section. The free constant $\mathbb{P}_0$ is determined by

$$\sum_{\overline{w}^h, \overline{s}^h, \overline{w}^l, \overline{s}^l, \overline{r}^l} P_{n,m}\left(\overline{w}^h, \overline{s}^h, \overline{w}^l, \overline{s}^l, \overline{r}^l\right) = 1$$

or

$$\sum_{\hat{t}\le k} P_{n,m}\left(\overline{0}, \overline{s}^h, \overline{0}, \overline{s}^l, \overline{0}\right) + \sum_{|r^l|\le n\le k,\ \hat{t}>k} P_{n,m}\left(\overline{0}^h, \overline{s}^h, \overline{w}^l, \overline{s}^l, \overline{r}^l\right) + \sum_{n>k, |r^l|\le k} P_{n,m}\left(\overline{w}^h, \overline{s}^h, \overline{w}^l, \overline{0}, \overline{r}^l\right) = 1$$

It is easy to show that the sum $\sum_{|r^l|\le n\le k,\ \hat{t}>k} P_{n,m}\left(\overline{0}^h, \overline{s}^h, \overline{w}^l, \overline{s}^l, \overline{r}^l\right)$ is equal to $v(1)$. From the previous section we know that $v(1)$ can be expressed in $\mathbb{P}_{k-1}$ and $\mathbb{P}_k$. Hence $v(1) = \mathbb{P}_0 v^0(1)$ where $v^0(1)$ can simply be expressed in $\mathbb{F}_k\,\mathbb{G}$ and products of the sort $\mathbb{Q}_{\hat{t}}\cdots\mathbb{Q}_0$. Now $\mathbb{P}_0$ follows from:

$$\mathbb{P}_0\{1 + \sum_{\hat{t}=0}^{k-2}\langle \mathbf{1}_{\hat{t}}, \mathbb{Q}_{\hat{t}}\cdots\mathbb{Q}_0\rangle + \left\langle \mathbf{1}_{n\le k,\hat{t}\ge k}, v^0(1)\right\rangle + \left\langle \mathbf{1}_{n>k}, \left(I - \mathbf{Z}^{-1}(1)\right)v_k^0(1)\right\rangle\} = 1,$$

where $v_k^0(1)$ is part of vector $v^0(1)$ with k high priority items in the system and $\mathbf{1}_*$ is a vector with all elements equal to 1 and with a dimension corresponding to dimension of the state spaces (cf. expressions 12-13), i.e. $\dim(\mathbf{1}_{\hat{t}}) = \binom{N^h+N^l+\hat{t}-1}{\hat{t}}$, $\dim\left(\mathbf{1}_{n\le k,\hat{t}>k}\right) = \sum_{i=0}^{k}\left[\sum_{n=i}^{k}\binom{N^h+n-1}{n}\binom{N^l+k-n-1}{k-n}\right]\times \binom{N^l+i-1}{i}$ and $\dim(\mathbf{1}_{n>k}) = \binom{N^h+k-1}{k}$

Hence, we found the probability states for $\hat{t} = 0\ldots k+1$ and the values of the first derivative of the function $v(1)$ that we need to calculate the performance measures for the queueing system.

5 Performance measures.

In this section we will concentrate on the performance criteria for the low priority items in the system, since performance indicators for the high priority items can be calculated using the non-priority multi-class, multi-serve queue analysis by Van Harten and Sleptchenko [11]. The latter is possible due to the preemptive priority rule in our system, i.e. low priority items don't influence processing of high priority items and therefore they can be ignored.

We show here how to calculate performance indicators like the mean number of low priority items of type i in the queue, the mean number of the low priority items of type i in the postponed

state and the first two moments of the total number of the low priority items of type i in the system. We denote the number of low priority items of subclass i in the queue as q_i^l, in the postponed state as PS_i^l and in the system as R_i^l. Such performance indicators play a role in spare part service networks as discussed in the introduction (Sherbrooke [13], Sleptchenko et al. [12]). Other interesting performance measures are the expected number of postponements of per item of type i, the expected residential time of items of type i in service, queue and postponement, respectively.

The mean number of the low priority item i in the queue can be found as sum of all probability states with low priority items in the queue (i.e. zones I and II) multiplied by the number of low priority item i in the queue:

$$E\left[q_i^l\right] = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} \sum_{\substack{\overline{w}^h,\overline{s}^h \\ |\overline{w}^h|+|\overline{s}^h|=n}} \sum_{\substack{\overline{w}^l,\overline{s}^l,\overline{r}^l \\ |\overline{w}^h|+|\overline{r}^l|+|\overline{s}^l|=m}} w_i^l P_{n,m}\left(\overline{w}^h,\overline{s}^h,\overline{w}^l,\overline{s}^l,\overline{r}^l\right).$$

In this expression various terms can be simplified via the function $v(\xi)$ and via the matrix $\mathbf{Z}(\xi)$ using Taylor expansion:

$$\begin{aligned}
E\left[q_i^l\right] &= \sum_{n=k+1}^{\infty}\sum_{m=0}^{\infty} \sum_{\substack{\overline{w}^h,\overline{s}^h \\ |\overline{w}^h|+|\overline{s}^h|=n,|\overline{s}^h|=k}} \sum_{\substack{\overline{w}^l,\overline{r}^l \\ |\overline{w}^l|+|\overline{r}^l|=m,|\overline{r}^l|\le k}} w_i^l P_{n,m}\left(\overline{w}^h,\overline{s}^h,\overline{w}^l,\overline{0},\overline{r}^l\right) \\
&\quad + \sum_{n=0}^{k}\sum_{m=k-n}^{\infty} \sum_{\substack{\overline{s}^h \\ |\overline{s}^h|=n}} \sum_{\substack{\overline{w}^l,\overline{r}^l,\overline{s}^l \\ |\overline{w}^l|+|\overline{r}^l|+|\overline{s}^l|=m,|\overline{r}^l|\le n}} w_i^l P_{n,m}\left(\overline{0},\overline{s}^h,\overline{w}^l,\overline{s}^l,\overline{r}^l\right) \\
&= \sum_{m=1}^{\infty}\sum_{n=k+1}^{\infty} \sum_{\substack{\overline{s}^h \\ |\overline{s}^h|=k}} \sum_{\substack{\overline{r}^l \\ |\overline{r}^l|\le k}} a_i^l \frac{1}{\left(|\overline{w}^l|-1\right)!}\left(\frac{d}{d\xi}\right)^{|\overline{w}^l|}\left[\left(\mathbf{Z}(\xi)^{-1}\right)^{n-k} v_k(\xi)\right]_{\xi=0} \\
&\quad + \sum_{n=0}^{k}\sum_{m'=k-n}^{\infty} \sum_{w_i^l,r_i^l,s_i^l} w_i^l m'! \prod_{i=1}^{N^l} \frac{\left(a_i^l\right)^{w_i^l}}{w_i^l!}\left[\frac{1}{m'!}\frac{d^{m'}}{d\xi^{m'}} v(\xi)\right]_{\xi=0}
\end{aligned}$$

which finally gives us

$$\begin{aligned}
E\left[q_i^l\right] &= -a_i^l\left\langle \mathbf{1}_{n>k}, \left(\mathbf{Z}(1)-\mathbf{I}\right)^{-1}\mathbf{Z}'(1)\left(\mathbf{Z}(1)-\mathbf{I}\right)^{-1} v_k(1)\right\rangle \\
&\quad + a_i^l\left\langle \mathbf{1}_{n>k}, \left(\mathbf{Z}(1)-\mathbf{I}\right)^{-1}\mathbf{Z}(1)\, v_k'(1)\right\rangle + a_i^l\left\langle \mathbf{1}_{n\le k,\hat{t}\ge k}, v'(1)\right\rangle
\end{aligned}$$

Here v_k refers to the vector components of v with $\left|\overline{s}^h\right| = k$. The notation $\mathbf{1}_*$ is used for a vector with components 1 of the dimension indicated by the subscript as introduced before.

In an analogous way we obtain

$$E\left[PS_i^l\right] = \left\langle \chi_{n>k}^{r_i^l}, v_k(1)\right\rangle + \left\langle \chi_{n\le k,\hat{t}\ge k}^{r_i^l}, v(1)\right\rangle,$$

where $\chi_{n>k}^{r_i^l}$ and $\chi_{n\le k,\hat{t}\ge k}^{r_i^l}$ have as components the amounts of postponed low priority items i corresponding with the vector component $(\overline{s}^h,\overline{s}^l,\overline{r}^l)$.

Using similar vectors $\chi_{n\le k,\hat{t}\ge k}^{s_i^l}$ and $\chi_{\hat{t}}^{s_i^l}$, which keep track of amounts of the low priority items in service we can write down an expression for the mean number of items of type i in service:

$$E\left[SR_i^l\right] = \sum_{\hat{t}=0}^{k-2}\left\langle \chi_{\hat{t}}^{s_i^l}, \mathbb{Q}_{\hat{t}}\cdots\mathbb{Q}_0\mathbb{P}_0\right\rangle + \left\langle \chi_{n\le k,\hat{t}\ge k}^{s_i^l}, v(1)\right\rangle,$$

Note that the mean number of items of type i in service can also be estimated via Little's law, i.e.

$$E\left[SR_i^l\right] = \lambda_i^l \Big/ \mu_i^l$$

We can use this as a test to check the accuracy of our numerical approximations in the next section.

The expected value of R_i^l is composed of these three terms:

$$E\left[R_i^l\right] = E\left[SR_i^l\right] + E\left[q_i^l\right] + E\left[PS_i^l\right]$$

Next Little's law can be applied to calculate performance indicators as the mean waiting time $E\left[W_i^l\right]$, the mean postponement time $E\left[PsTime_i^l\right]$ and the mean sojourn time $E\left[SJTime_i^l\right]$:

$$\begin{aligned} \lambda_i^l E\left[W_i^l\right] &= E\left[q_i^l\right] \\ \lambda_i^l E\left[PsTime_i^l\right] &= E\left[PS_i^l\right] \\ \lambda_i^l E\left[SJTime_i^l\right] &= E\left[R_i^l\right] \end{aligned}$$

and, of course,

$$E\left[SJTime_i^l\right] = \frac{1}{\mu_i^l} + E\left[W_i^l\right] + E\left[PsTime_i^l\right]$$

Note that $E\left[PsTime_i^l\right]$ has to be interpreted as: the expected total time an item of type i spends in a postponed state between the moment it leaves the queue and the moment its service process is completed. Further, we should note that even though preemption occurs, the expected total service time equals $1/\mu_i^l$, due to Poisson character of the end of service events.

Let us now focus on another interesting quantity: $E[nrPreemptEvent_i^l]$, the expected number of preemption events per item of type i. In order to compute it we need the arrival rate of a low priority item into postponed state. It can be calculated using the state probabilities estimated in the previous sections. The arrival rate of low priority items i into postponed state is equal to arrival rate of high priority items multiplied by the probability that item i was withdrawn from the service:

$$\lambda_i^{ps} = \Lambda^h \sum_{\overline{s}^h,\overline{w}^l,\overline{s}^l,\overline{r}^l} \frac{s_i^l}{\left|\overline{s}^l\right|} P_{n,m}\left(0,\overline{s}^h,\overline{w}^l,\overline{s}^l,\overline{r}^l\right) = \Lambda^h \left\langle \chi^{s_i^l}_{n\le k,\hat{t}\ge k}, v\left(1\right)\right\rangle$$

where components of vectors $\chi^{s_i^l}_{\hat{t}}$ and $\chi^{s_i^l}_{n\le k,\hat{t}\ge k}$ are equal to $\frac{s_i^l}{\left|\overline{s}^l\right|}$ of corresponding vector component $\left(\overline{s}^h,\overline{s}^l,\overline{r}^l\right)$

Comparing the number of preemption events with the number of arrivals over a long interval it is clear that number of preemption events per each item entering the system is equal to:

$$E[nrPreemptEvent_i^l] = \lambda_i^{ps} / \lambda_i^l$$

It is now also possible to compute the expected time between the moment an item of type i is postponed and the next moment when the service process was resumed again, i.e. the expected re-entrance into service time, $E[reenterTime_i^l]$. Using Little's law again, we obtain

$$\lambda_i^{ps} E[ReenterTime_i^l] = E\left[PS_i^l\right]$$

Note that the number of preemption events per each item entering the system can be also estimated as:

$$E[nrPreemptEvent_i^l] = E\left[PsTime_i^l\right] \Big/ E[ReenterTime_i^l]$$

Let us now conclude this section on performance measures with some remarks on the calculation of the second moment of R_i^l. Here we should take into account the correlations between numbers of items in queue, in service and in the postponed states. This can be done analogous to computations for non-priority systems in van Harten and Sleptchenko [11]. After a lengthy computation (details can be provided to the reader upon request) we obtain:

$$
\begin{aligned}
E\left[\left(R_i^l\right)^2\right] = &+ \left(a_i^l\right)^2 \left\langle \mathbf{1}_{n>k}, 2\left[(\mathbf{Z}(1)-\mathbf{I})^{-1}\mathbf{Z}'(1)(\mathbf{Z}(1)-\mathbf{I})^{-1}\mathbf{Z}'(1)(\mathbf{Z}(1)-\mathbf{I})^{-1}\right.\right. \\
&\left.\left. - (\mathbf{Z}(1)-\mathbf{I})^{-1}\mathbf{Z}''(1)(\mathbf{Z}(1)-\mathbf{I})^{-1}\right] v_k(1)\right\rangle \\
&- 2\left(a_i^l\right)^2 \left\langle \mathbf{1}_{n>k}, (\mathbf{Z}(1)-\mathbf{I})^{-1}\mathbf{Z}'(1)(\mathbf{Z}(1)-\mathbf{I})^{-1}\mathbf{Z}'(1)(\mathbf{Z}(1)-\mathbf{I})^{-1} v_k'(1)\right\rangle \\
&+ \left(a_i^l\right)^2 \left\langle \mathbf{1}_{n>k}, (\mathbf{Z}(1)-\mathbf{I})^{-1}\mathbf{Z}(1) v_k''(1)\right\rangle \\
&- a_i^l \left\langle \mathbf{1}_{n>k}, (\mathbf{Z}(1)-\mathbf{I})^{-1}\mathbf{Z}'(1)(\mathbf{Z}(1)-\mathbf{I})^{-1} v_k(1)\right\rangle + a_i^l \left\langle \mathbf{1}_{n>k}, (\mathbf{Z}(1)-\mathbf{I})^{-1} v_k'(1)\right\rangle \\
&- 2a_i^l \left\langle \chi_{n>k}^{r_i^l}, (\mathbf{Z}(1)-\mathbf{I})^{-1}\mathbf{Z}'(1)(\mathbf{Z}(1)-\mathbf{I})^{-1} v_k(1)\right\rangle \\
&+ 2a_i^l \left\langle \chi_{n>k}^{r_i^l}, (\mathbf{Z}(1)-\mathbf{I})^{-1} v_k'(1)\right\rangle + \left\langle \left(\chi_{n>k}^{r_i^l}\right)^2, \left(\mathbf{I}-\mathbf{Z}(1)^{-1}\right)^{-1} v_k(1)\right\rangle \\
&+ \left(a_i^l\right)^2 \left\langle \mathbf{1}_{n\le k,\hat{t}\ge k}, v''(1)\right\rangle + a_i^l \left(\left\langle \mathbf{1}_{n\le k,\hat{t}\ge k}, v'(1)\right\rangle + 2\left\langle \chi_{n\le k,\hat{t}\ge k}^{r_i^l+s_i^l}, v'(1)\right\rangle\right) \\
&+ \left\langle \left(\chi_{n\le k,\hat{t}\ge k}^{r_i^l+s_i^l}\right)^2, v(1)\right\rangle + \sum_{\hat{t}=0}^{k-2} \left\langle \left(\chi_{\hat{t}}^{s_i^l}\right)^2, \mathbb{Q}_{\hat{t}}\cdots\mathbb{Q}_0\mathbb{P}_0\right\rangle
\end{aligned}
$$

In an analogous way second moments of SR_i^l, q_i^l, PS_i^l can be given.

6 Numerical experiments.

To examine the computational efforts necessary to obtain accurate results by the proposed algorithm, we shall first compare the values of first and second moments of numbers of items in the system obtained by the proposed algorithm with different numbers of iterations t in formula (17).

A set of numerical experiments, was designed to see the impact of this numbers of iterations t and the total utilization rate ρ. We chose the following parameter settings for our numerical experiments:

- k is equal to 4,
- N^h and N^l are fixed to 2,
- the arrival rates are fixed and distributed as: $\lambda_1^h = 1.75$, $\lambda_2^h = 2.25$, $\lambda_1^l = 2.75$, $\lambda_2^l = 3.25$
- the service rates satisfy: $\mu_i^h = \mu^h\left(1+\delta_i^h\right)$ and $\mu_i^l = \mu^h\left(1+\delta_i^l\right)$, where the δ_i's are equal to 0.5 for odd and -0.5 for even items.
- μ^h is varied so that ρ is equal to 0.75, 0.85 or 0.95

In figure 2 we plot the maximum error between the mean and the variance of the numbers of low priority items in the system obtained by the algorithm with different amounts of iterations and the mean and the variance obtained after 80 iterations. Under the axis with the number of iterations we indicate the CPU time in seconds required to do these iterations using a Pentium

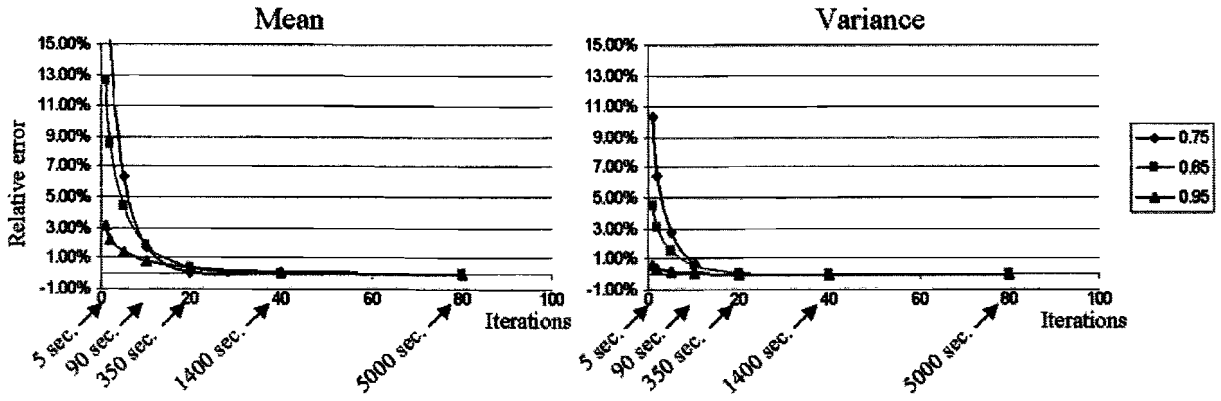

Figure 2: Error in performance estimators of low priority items against amounts of iterations and computation time for cases with 4 item types and 4 servers and different utilisation rates.

III-500 PC with the algorithm implemented in Borlan Delphi 5. This plot shows us how the approximation error decreases with increasing number of iterations, and how much time we need to obtain these errors.

It is an interesting effect that for the systems with higher utilisation rates we can obtain good results for low priority items with less iterations. On the other hand we have an exact method to estimate the performance characteristics of high priority items, that is independent of ρ as for computational effort (cf. van Harten, Sleptchenko [11]). Comparison of these results with the results obtained by a discrete event simulation have also shown us that after 10 iterations in average we come to performance estimators of the low priority items within a 3% confidence interval. Further, we have checked whether the test $E\left[SR_i^l\right] = \lambda_i^l/\mu_i^l$ from the previous section is satisfied. The experiments have shown that after 20 iterations the average relative error is $\sim 0.2\%$ and after 40 iterations is $\sim 0.04\%$

Next, we shall present some results on the influence of the most important parameters of the MCMS priority system on the performance measures as introduced in the previous section. From queueing theory we know that the total utilization rate ρ and the number of servers k are such parameters for any queueing system. However, we have learned from the experiments with the MCMS **non**-priority queueing system (van Harten and Sleptchenko [11]) that the fractions of arrival rates (a_i) and the perturbations of the service times (δ_i) might also seriously influence the performance characteristics of the queueing system.

We did computations for a large set of instances. Since the effects we want to discuss are already present for small systems we shall only present here the results on experiments for 3 servers and 3 items. One of items has high priority and two have low priority. The utilization rate ρ is fixed to 95%. First of all we want to see the influence of difference between the service times. This difference is completely defined by the parameters γ and $\frac{\mu_1^l}{\mu_2^l}$, where μ_1^l and μ_2^l are the service rates of the low priority classes. Also we would like to see the influence of different values of the utilization rate of high priority items. Hence, we vary in fact three parameters: ρ^h, γ and $\frac{\mu_1^l}{\mu_2^l}$. The other parameters are either fixed (e.g. fractions of arrival rates within group of the low priority item are equal to $a_1^l = 0.3$ and $a_2^l = 0.7$), or are completely defined by the other parameters (e.g. δ_1^l and δ_2^l). In this way we have done 18 experiments for γ and $\frac{\mu_1^l}{\mu_2^l}$ equal to 0.5, 1 and 2 and for ρ^h equal to 20% and 60%.

Some interesting performance characteristics are presented in the table 1: the expected total number of items in the system for each low priority subclass $E\left[R_i^l\right]$, the expected number of

Table 1: Expected numbers of low priority items in the system, in the queue, in the postpone state and expected number of postpone events per each low priority item entering the system for different values of γ, ρ^h and $\frac{\mu_1^l}{\mu_2^l}$ (the upper numbers in each cell corresponds to the first low priority item and the numbers under to the second low priority item).

ρ^h		20%				60%			
$\frac{\mu_2}{\mu_3}$	γ	$E\left[R_i^l\right]$	$E\left[q_i^l\right]$	$E\left[PS_i^l\right]$	$\lambda_i^{ps}/\lambda_i^l$	$E\left[R_i^l\right]$	$E\left[q_i^l\right]$	$E\left[PS_i^l\right]$	$\lambda_i^{ps}/\lambda_i^l$
0.5	0.5	6.2	5.0	0.14	0.73	4.3	3.4	0.47	7.66
		13.1	11.7	0.67	0.35	8.9	7.9	0.52	3.73
	1	6.7	5.6	0.10	0.96	5.9	4.9	0.40	7.68
		14.4	13.0	0.11	0.46	12.6	11.6	0.44	3.73
	2	7.8	6.7	0.07	0.78	8.9	8.0	0.33	5.77
		16.9	15.5	0.08	0.38	19.7	18.8	0.35	2.80
1	0.5	5.3	4.5	0.09	4.05	3.8	3.2	0.30	27.9
		12.4	10.6	0.21	2.21	8.9	7.4	0.70	16.1
	1	5.8	5.1	0.06	2.41	5.3	4.8	0.25	16.4
		13.6	11.9	0.15	1.49	12.4	11.1	0.58	10.5
	2	6.9	6.9	0.04	1.39	8.4	7.9	0.20	9.45
		16.1	14.4	0.10	0.93	19.6	18.3	0.47	6.49
2	0.5	5.3	4.8	0.05	6.15	3.7	3.30	0.17	40.6
		13.4	11.2	0.25	4.59	9.4	7.70	0.83	31.8
	1	5.8	5.4	0.04	3.29	5.2	4.9	0.14	21.7
		14.6	12.6	0.19	2.75	13.0	11.4	0.70	18.8
	2	6.9	6.5	0.02	1.76	8.3	8.0	0.11	11.6
		17.0	15.1	0.13	1.60	20.0	18.6	0.57	10.9

items in the queue for each low priority subclass $E\left[q_i^l\right]$, the expected number of the postponed items for each low priority subclass $E\left[PS_i^l\right]$ and the expected number of preemption events per each low priority item entering the system $\lambda_i^{ps}/\lambda_i^l$.

From table 1 we can draw the following conclusions:

- Numbers of items in the postponed state increase with increase of the utilisation rate of the high priority items ρ^h (this is not a trivial result since $\rho = \rho^h + \rho^l$ is constant, hence increasing of ρ^h means decreasing of ρ^l).
- The dependence of the total number of postponed items on the ratio $\frac{\mu_1^l}{\mu_2^l}$ is quite small.
- However the dependence of the total number of items in queue on the ratio $\frac{\mu_1^l}{\mu_2^l}$ is remarkable. Namely the numbers of items in queue (hence, the waiting times) are lower when service times of low priority items are equal ($\mu_1^l = \mu_2^l$), than when perturbations of service times of low priority items occur. This can be interpreted as a sort of Pollaczek-Khintchine effect (cf. Tijms [14]), i.e. the average waiting time is increases when the variability of the service time increseas.
- The numbers of low priority items in queue (hence, the waiting times) are lower when service time of high priority items is shorter (γ is smaller).
- The numbers of the low priority items in the queue decrease with increase of the utilisation rate of high priority items (ρ^h) when the service time of the high priority items is shorter or equal ($\gamma \leq 1$) and increase with the increase of the utilisation rate of high priority items

(ρ^h) when the service time of the high priority items is longer than the average service time of the low priority items ($\gamma > 1$).

It is also possible to derive from this table the time parameters, such as waiting time in the queue and in the postponed state or the total time spent in the system. This can be done using Little's law as was shown in section 5.

To conclude this section we give a sketch of the effect of applying a priority queueing rule. Therefore we vary ρ in the experiment with $N^h = 1, N^l = 2, k = 3$, $\mu_1^l/\mu_2^l = 0.5$, $\gamma = 0.5$ and $\rho^h = 0.6\rho$, $\rho^l = 0.4\rho$, compare the total numbers of items for this experiments in cases with and without priority rules (figure 3).

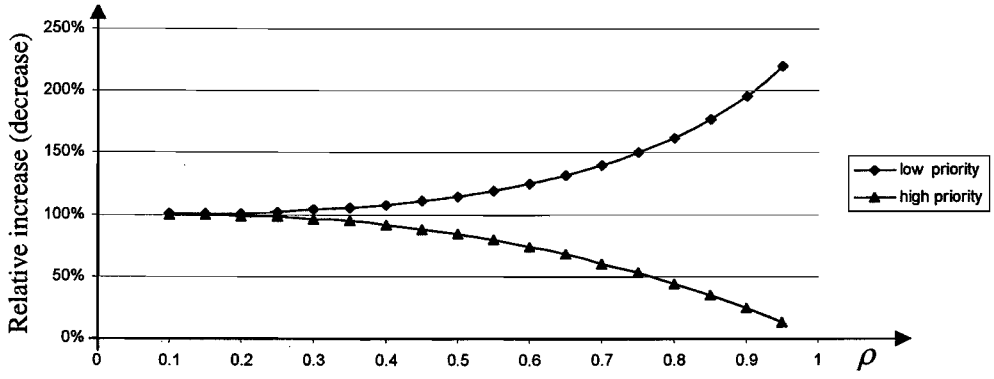

Figure 3: Relative increase (decrease) of the total number of low (high) priority items caused by introduction of priority rules into the queueing system.

The obtained picture shows not only the fact that the introduction of priority rules cause increase (decrease) of the total number of low (high) priority items in the system but also the scale of this increase (decrease).

7 Conclusions and generalizations.

In this paper we derived a method to analyze multi-class $M/M/k$ priority queues with preemptive priority and two priority groups (high and low). Each group of priority can contain several classes of items with different arrival and service rates. The proposed method is based on solution of the stationary state equations. It uses an iteration algorithm. The computational effort to find good approximation depends on the number of types, number of servers and utilization rate (since higher utilization rate requires more iterations). For example, system with 4 subclasses, 4 servers and utilization rate of 95% needs $O(20)$ matrix operations, where dimension of each matrix is 238×238. Due to the increase of the size of matrices the computational effort increases rapidly for large k, N^h and N^l. Approximations are then necessary. As an approximation we then replace groups of items with similar characteristic by one item with average service properties. We have some results, but this is still work in progress.

This method can in principle be extended to solve problems with more priority groups. This is possible iteratively due to preemption property. That is, we can estimate performance estimators for each priority group ignoring all classes with lower priorities and aggregating all classes with high priorities into one high priority group.

Also it can be used for cases where items have hyperexponential (H_x) service times. We can deal with these cases by representing each class as x classes with exponential distributed service

time and adopting the performance estimators for the total number of items in the system among these x classes.

Finally, for applications of MCMS priority queueing theory in spare parts managements we refer to a forthcoming report.

References

[1] Buzen, J.P., A.B.Bondi. The response times of priority classes under preemptive resume in $M/M/m$ queues. Operations Research, Vol.31, No.3, 456-465, 1983.

[2] Devis, R.H. Waiting time distribution of a multi-server priority queueing model. Operations research, 14, 133-136, 1966.

[3] Gail, H.R., S.L.Hantler, B.A.Taylor. Analysis of a non-preemptive priority multiserver queue. Advances in Applied Probability. 20, No.4, 852-879, 1988

[4] Gail, H.R., S.L.Hantler, B.A.Taylor. On preemptive Markovian queue with multiple servers and two priority classes. Mathematical Operations Research. 17, No.2, 365-391, 1992

[5] Hooghiemstra, G., M.S. Keane, S. van der Ree. Power series for stationary distributions of coupled processor models. SIAM Journal of Applied Mathematics. 48, pp. 1159-1166, 1988.

[6] Kanadia, A.S., M.F.Kazmi and A.C.Mitchell. Analysis of a finite capacity non preemptive priority queue. Computers and Operations Research, 11/3, 337-343, 1984.

[7] Kao, E.P.C. and K.S.Narayanan. Computing steady state probabilities of a nonpreemptive priority queue. ORSA Journal on Computing, 2, 211-218, 1990.

[8] Kao, E.P.C., and S.D. Wilson. Analysis of nonpreemptive priority queues with multiple servers and two priority classes. European Journal of Operational Research. 118, 181-193, 1999.

[9] Kella, O., and U. Yechiali. Waiting times in the non-preemptive priority $M/M/c$ queue. Communications in statistics - Stochastic models, 1, 257-262, 1985.

[10] Mitrani, I., P.J.B.King. Multiprocessor systems with preemptive priorities. Performance Evaluations 1, 118-125, 1981.

[11] Harten, A. van, A.Sleptchenko. On multi-class multi-server queueing and spare parts management. submitted in Queueing Systems.

[12] Sleptchenko, A., M.C. van der Heijden, A. van Harten. Effects of finite repair capacity in multi-echelon, multi-indenture service part supply systems, submitted in International Journal of Production Economics.

[13] Sherbrooke, C.C., Optimal inventory modeling of systems: Multi-echelon techniques, Wiley, New York, 1992.

[14] Tijms, H.C. Stochastic Models: An algorithmic approach, Wiley, Chichester, 1994.

[15] Wagner, D. Analysis of mean values of a multi-server model with non-preemptive priorities and non-renewal input. Communications in statistics - Stochastic models. 13(1), 67-84, 1997.

[16] Wagner, D. Waiting time of a finite-capacity multi-server model with non-preemptive priorities. European Journal of Operational Research. 102, 227-241, 1997.

[17] Wagner, D. A finite-capacity multi-server multi-queueing model with non-renewal input. Annals of Operational Research. 79, 63-82, 1998.

Appendix 1: Proof of Lemma 1

We will prove this lemma by induction using the matrix form (6) of the equilibrium equations (1):

$$\left(\left(1+\rho^h+\gamma\rho^l\right)\mathbf{I}+\overline{\delta}^h\right)\mathbf{P}_{n,m}\left(\overline{w}^l,\overline{0}\right)=\rho^h\mathbf{P}_{n-1,m}\left(\overline{w}^l,\overline{0}\right)$$
$$+\gamma\rho^l\sum_{j=1}^{N^l}a_j\mathbf{P}_{n,m-1}\left(\overline{w}^l-e_j,\overline{0}\right)+A\mathbf{P}_{n+1,m}\left(\overline{w}^l,\overline{0}\right)$$

Proof. For $m=0$ equation (6) can be written as

$$\left(\left(1+\rho^h+\gamma\rho^l\right)\mathbf{I}+\overline{\delta}^h\right)\mathbf{P}_{n,0}\left(\overline{0},\overline{0}\right)=\rho^h\mathbf{P}_{n-1,0}\left(\overline{0},\overline{0}\right)+A\mathbf{P}_{n+1,0}\left(\overline{0},\overline{0}\right)$$

which is similar to the multi-class multi-server equilibrium equation having a solution of the form $\mathbf{P}_{n,0}\left(\overline{0},\overline{0}\right)=\left(\mathbf{Z}^{-1}\right)^{n-k}\mathbf{C}$ (see [11]), where $\mathbf{Z}$ should satisfy the equation

$$\left(\left(1+\rho^h+\gamma\rho^l\right)\mathbf{I}+\overline{\delta}^h\right)=\rho^h\mathbf{Z}+A\mathbf{Z}^{-1},\quad|\sigma\left(\mathbf{Z}\right)|>1$$

similar to the equation (7) with $\xi=0$. So we have that the solution in the form (8) is the solution of the equation (1) for $m=0$.

For $m>0$ we first define $\mathbf{P}'_n\left(\overline{x}\right)$ as the solution of:

$$\left(\left(1+\rho^h+\gamma\rho^l\right)\mathbf{I}+\overline{\delta}^h\right)\mathbf{P}'_n\left(\overline{x}\right)=\rho^h\mathbf{P}'_{n-1}\left(\overline{x}\right)+\gamma\rho^l\sum_{j=1}^{N^l}a_j^lx_j\mathbf{P}'_n\left(\overline{x}\right)+A\mathbf{P}'_{n+1}\left(\overline{x}\right)\qquad(21)$$

with $x_j\in[0,1]$, being just a parameter.

It follows that $\mathbf{P}'_n\left(\overline{x}\right)=\left(\mathbf{Z}\left(\rho^h,\rho^l,\gamma;\overline{x}\right)^{-1}\right)^{n-k}\mathbf{C}'\left(\overline{x}\right)$. By differentiation of (21) with respect to x_j we find

$$\left(\left(1+\rho^h+\gamma\rho^l\right)\mathbf{I}+\overline{\delta}^h\right)\frac{d\mathbf{P}'_n\left(\overline{x}\right)}{dx_j}=\rho^h\frac{d\mathbf{P}'_{n-1}\left(\overline{x}\right)}{dx_j}+\gamma\rho^l\sum_{i=1}^{N^l}a_i^lx_i\frac{d\mathbf{P}'_n\left(\overline{x}\right)}{dx_i}$$
$$+A\frac{d\mathbf{P}'_{n+1}\left(\overline{x}\right)}{dx_j}+\gamma\rho a_j^l\mathbf{P}'_n\left(\overline{x}\right)$$

Hence $\left[\frac{d\mathbf{P}'_n\left(\overline{x}\right)}{dx_j}\right]_{\overline{x}=0}$ satisfies the equation (1) for $m=1$.

Using the general property $\left(\frac{d}{dx}\right)^m\left(xf\left(x\right)\right)=m\left(\frac{d}{dx}\right)^{m-1}f\left(x\right)+x\left(\frac{d}{dx}\right)^mf\left(x\right)$ we find by differentiation of (21) m_j-times with respect to each parameter x_j that:

$$\left(\left(1+\rho^h+\gamma\rho^l\right)\mathbf{I}+\overline{\delta}^h\right)\prod_{i=1}^{N^l}\left(\frac{d}{dx_i}\right)^{m_i}\mathbf{P}'_n\left(\overline{x}\right)=\rho^h\prod_{i=1}^{N^l}\left(\frac{d}{dx_i}\right)^{m_i}\mathbf{P}'_{n-1}\left(\overline{x}\right)$$
$$+\gamma\rho^l\sum_{v=1}^{N^l}a_v^lx_v\prod_{i=1}^{N^l}\left(\frac{d}{dx_i}\right)^{m_i}\mathbf{P}'_n\left(\overline{x}\right)+A\prod_{i=1}^{N^l}\left(\frac{d}{dx_i}\right)^{m_i}\mathbf{P}'_{n+1}\left(\overline{x}\right)$$
$$+\gamma\rho^l\sum_{v=1}^{N^l}m_va_v^l\left(\frac{d}{dx_v}\right)^{m_v-1}\prod_{\substack{i=1\\i\neq v}}^{N^l}\left(\frac{d}{dx_i}\right)^{m_i}\mathbf{P}'_n\left(\overline{x}\right)$$

Transforming this expression back to the $\mathbf{P}_{n,m}\left(\overline{w}^l,\overline{0}\right)$ we get:

$$\mathbf{P}_{n,m}\left(\overline{w}^l,\overline{0}\right)=\prod\frac{1}{m_j!}\left(\frac{d}{dx_j}\right)^{m_j}\left[\left(\mathbf{Z}^{-1}\left(\overline{x}\right)\right)^{n-k}\mathbf{C}\left(\overline{x}\right)\right]_{\overline{x}=\overline{0}}$$

Now using the fact that $\frac{d}{dx_j}=\frac{d\xi(\overline{x})}{dx_j}\frac{d}{d\xi}=a_j\frac{d}{d\xi}$ we can change the term $\sum_{j=1}^{N^l}a_j^l x_j$ in the equation (21) to the function $\xi\left(\overline{x}\right)=\sum_{j=1}^{N^l}a_j^l x_j$ and we can rewrite the probability states written as:

$$\mathbf{P}_{n,m}\left(\overline{w}^l,\overline{0}\right)=\prod\frac{\left(a_j^l\right)^{m_j}}{m_j!}\left(\frac{d}{d\xi}\right)^{m}\left[\left(\mathbf{Z}^{-1}\left(\xi\right)\right)^{n-k}\mathbf{C}\left(\xi\right)\right]_{\xi=0}$$

Finally, we recall that $\sum_{m_j=m}m!\prod\frac{\left(a_j^l\right)^{m_j}}{m_j!}=\left(\sum a_j\right)^m$ and we obtain

$$\begin{aligned}&\sum_m\frac{m!}{m!}\prod\frac{\left(a_j^l\right)^{m_j}}{m_j!}\left(\frac{d}{d\xi}\right)^{m}\left[\left(\mathbf{Z}^{-1}\left(\xi\right)\right)^{n-k}\mathbf{C}\left(\xi\right)\right]_{\xi=0}\\&=\sum_m\frac{1}{m!}\underbrace{\left(\sum a_j^l\right)^m}_{=1}\left(\frac{d}{d\xi}\right)^{m}\left[\left(\mathbf{Z}^{-1}\left(\xi\right)\right)^{n-k}\mathbf{C}\left(\xi\right)\right]_{\xi=0}=\left[\left(\mathbf{Z}^{-1}\left(\xi\right)\right)^{n-k}\mathbf{C}\left(\xi\right)\right]_{\xi=1}\end{aligned}$$

as a well-known Taylor series expansion of $\left(\mathbf{Z}\left(\rho^h,\rho^h,\gamma;\xi\right)^{-1}\right)^{n-k}$ around $\xi=1$, where the value of $\mathbf{Z}\left(\rho^h,\rho^h,\gamma;\xi\right)$ is found from

$$\left(\left(1+\rho^h+\gamma\rho^l\right)\mathbf{I}+\overline{\delta}^h\right)=\rho^h\mathbf{Z}+\gamma\rho^l\xi+A\mathbf{Z}^{-1},\quad\left|\sigma\left(\mathbf{Z}\right)\right|>1$$

This equation can be solved as in the case of the non-priority multi-class queue (van Harten and Sleptchenko [11]) using decoupling by $\overline{r}^l$.

So the solution has the form:

$$P_{n,m}\left(\overline{w}^h,\overline{s}^h,\overline{w}^l,\overline{0},\overline{r}^l\right)=\left|\overline{w}^h\right|!\prod_{i=1}^{N^h}\frac{\left(a_i^h\right)^{w_i^h}}{w_i^h!}\left(\prod\frac{\left(a_j^l\right)^{m_j}}{m_j!}\right)\left(\frac{d}{d\xi}\right)^{m}\left[\left(\mathbf{Z}^{-1}\right)^{n-k}\left(\xi\right)\mathbf{C}\left(\xi\right)\right]_{\xi=0}\left[\overline{s}^h,\overline{r}^l\right]$$

■

Appendix 2: On the derivatives of the matrix Z

Unlike a scalar function finding the derivative $\frac{d}{d\xi}\left[\mathbf{Z}\left(\xi\right)^{-1}\right]$ has to take into account the non-commutativity of $\frac{d}{d\xi}\mathbf{Z}\left(\xi\right)$ and $\mathbf{Z}\left(\xi\right)$. This can be in the following way:

$$\frac{d}{d\xi}:I=\mathbf{Z}\left(\xi\right)\mathbf{Z}\left(\xi\right)^{-1}\quad\Longrightarrow\quad 0=\frac{d}{d\xi}\mathbf{Z}\left(\xi\right)\mathbf{Z}\left(\xi\right)^{-1}+\mathbf{Z}\left(\xi\right)\frac{d}{d\xi}\left[\mathbf{Z}\left(\xi\right)^{-1}\right],$$

This provides a first relation between $\frac{d}{d\xi}\left[\mathbf{Z}\left(\xi\right)^{-1}\right]$ and $\frac{d}{d\xi}\mathbf{Z}\left(\xi\right)$. Next we can use the equation defining the matrix $\mathbf{Z}\left(\boldsymbol{\xi}\right)$, i.e. if we differentiate (7) with respect to ξ, we get

$$0=\rho^h\frac{d}{d\xi}\mathbf{Z}\left(\boldsymbol{\xi}\right)+\gamma\rho^l+A\frac{d}{d\xi}\left[\mathbf{Z}\left(\xi\right)^{-1}\right].$$

So, we have a $2\binom{N^h+k-1}{k}\left(\sum_{i=0}^{k}\binom{N^l+i-1}{i}\right)$-dimensional non-singular system of linear equations for $2\binom{N^h+k-1}{k}\left(\sum_{i=0}^{k}\binom{N^l+i-1}{i}\right)$ variables and, therefore, the derivatives $\frac{d}{d\xi}\left[\mathbf{Z}(\xi)^{-1}\right]$ and $\frac{d}{d\xi}\mathbf{Z}(\xi)$ for each value ξ are defined.

We can find the higher order derivatives in a similar way. We have a $2\binom{N^h+k-1}{k}\left(\sum_{i=0}^{k}\binom{N^l+i-1}{i}\right)$-dimensional non-singular system of linear equations for $2\binom{N^h+k-1}{k}\left(\sum_{i=0}^{k}\binom{N^l+i-1}{i}\right)$ variables:

$$\frac{d^m}{d\xi^m}\mathbf{Z}(\xi)\,\mathbf{Z}(\xi)^{-1}+\mathbf{Z}(\xi)\frac{d^m}{d\xi^m}\left[\mathbf{Z}(\xi)^{-1}\right]=-\sum_{i=1}^{m-1}\frac{m!}{i!\,(m-i)!}\frac{d^i}{d\xi^i}\mathbf{Z}(\xi)\frac{d^{m-i}}{d\xi^{m-i}}\left[\mathbf{Z}(\xi)^{-1}\right]$$

$$\rho^h\frac{d^m}{d\xi^m}\mathbf{Z}(\boldsymbol{\xi})+A\frac{d^m}{d\xi^m}\left[\mathbf{Z}(\xi)^{-1}\right]=0$$

and, therefore, the derivatives $\frac{d^m}{d\xi^m}\left[\mathbf{Z}(\xi)^{-1}\right]$ and $\frac{d^m}{d\xi^m}\mathbf{Z}(\xi)$ for each value ξ can be obtained from the lower order derivatives.

Note that the equations for derivatives $\frac{d^m}{d\xi^m}\left[\mathbf{Z}(\xi)^{-1}\right]$ and $\frac{d^m}{d\xi^m}\mathbf{Z}(\xi)$ have block-diagonal structure, consist of $\left(\sum_{i=0}^{k}\binom{N^l+i-1}{i}\right)$ blocks for different $\overline{r}^l$ of dimension $2\binom{N^h+k-1}{k}\times 2\binom{N^h+k-1}{k}$ each, moreover these blocks have the same structure for different $\overline{r}^l$. Hence, to construct derivatives $\frac{d^m}{d\xi^m}\left[\mathbf{Z}(\xi)^{-1}\right]$ and $\frac{d^m}{d\xi^m}\mathbf{Z}(\xi)$ for each m (given that derivatives of lower order are already known) is enough to solve once a system of linear equation of dimension $2\binom{N^h+k-1}{k}\times 2\binom{N^h+k-1}{k}$ for one of $\overline{r}$.

Appendix 3: Notations

k – number of servers.

N^h – number of high priority classes

N^l – number of low priority subclasses N^l – number of low priority subclasses (total number of types will be N^h+N^l).

Λ^h – total arrival rate of the high priority items, and λ_i^l – arrival rate of the high priority subclass i.

Λ^l – total arrival rate of the low priority items, and λ_i^l – arrival rate of the low priority subclass i.

a_i^h – arrival fraction of the high priority subclass i.

a_i^l – arrival fraction of the low priority subclass i.

μ^h – average service rate of the high priority items, and μ_i^l – service rate of the high priority subclass i.

μ^l – average service rate of the low priority items, and μ_i^l – service rate of the low priority subclass i.

γ – fraction of average service rates of high and low priority classes ($\gamma=\frac{\mu^l}{\mu^h}$).

δ_i^h – permutation coefficient of service rate of high priority subclass i $\left(\mu_i^h=\mu^h\left(1+\delta_i^h\right)\right)$

δ_i^l – permutation coefficient of service rate of low priority subclass i $\left(\mu_i^l = \mu^h\left(1+\delta_i^l\right)\right)$

n – total number of the high priority items in the system, and n_i – number of items of the high priority subclass i.

m – total number of the low priority items in the system, and m_i – number of items of the low priority subclass i.

$\hat{t}$ – total number of items in the system $(\hat{t} = n + m)$, and t – total number of items in the system minus number of items in service $(t = \hat{t} - k)$.

$\overline{w}^h$ – vector describing amounts of high priority items in the queue.

$\overline{s}^h$ – vector describing amounts of high priority items in the service.

$\overline{w}^l$ – vector describing amounts of low priority items in the queue.

$\overline{s}^l$ – vector describing amounts of low priority items in the service.

$\overline{r}^l$ – vector describing amounts of postponed low priority items.

$\mathbf{Z}(\xi)$ – solution matrix of the states $(n > k)$, obtained as the solution of the equation (7)

$\mathbf{P}_n$ – vector of servers states probabilities when there is a high priority items in the queue.

$\mathbb{P}_{\hat{t}}$ – vector of servers states probabilities when there is no high priority items in the queue.

$v(\xi)$ – vector-function, which defines probabilities of system states with no high priority item in the queue.

$\mathbb{H}(\xi)$ – matrix-function, which defines the vectors-function $v(\xi)$.

q_i^l – number of items of low priority subclass i in the queue.

PS_i^l – number of items of low priority subclass i in the postponed state.

SR_i^l – number of items of low priority subclass i in the service.

R_i^l – number of items of low priority subclass i in the system.

W_i^l – waiting time of items of low priority subclass i.

$PsTime_i^l$ – time spent by items of low priority subclass i in the postponed state.

$SJTime_i^l$ – sojourn time of items of low priority subclass i.

λ_i^{ps} – arrival rate of a low priority item i into postponed state

$nrPreemptEvent_i^l$ – number of preemption events per low priority item of type i

$ReenterTime_i^l$ – time between the moment a low priority item of type i is postponed and the moment when the service process is resumed again.