On multi-class multi-server queueing and spare parts management

Citation for published version (APA):
On multi-class multi-server queueing and spare parts management

Aart van Harten and Andrei Sleptchenko
WP-49
On multi-class multi-server queueing and spare parts management

Aart van Harten* Andrei Sleptchenko

Dept. of Technology and Management,
University of Twente, The Netherlands

Abstract

Multi-class multi-server queuing problems are a generalization of the well-known M/M/k situation to arrival processes with clients of N types that require exponentially distributed service with different averaged service time. Problems of this sort arise naturally in various applications, such as spare parts management, for example. In this paper we give a procedure to construct exact solutions of the stationary state equations. Essential in this procedure is the reduction of the problem for \( n = \) the number of clients in the system \( > k \) to a backwards second order difference equation with constant coefficients for a vector in a linear space with dimension depending on \( N \) and \( k \), denoted by \( d(N, k) \). Precisely \( d(N, k) \) of its solutions have exponential decay for \( n \to \infty \). Next, using this as input, the equations for \( n \leq k \) can be solved by backwards recursion. It follows that the exact solution does not have a simple product structure as one might expect intuitively. Further, using the exact solution several interesting performance measures related to spare parts management can be computed and compared with heuristic approximations. This is illustrated with numerical results.

*corresponding author: P.O.box 217, 7500AE Enschede, The Netherlands, e-mail: A.vanHarten@sms.utwente.nl
1 Introduction.

Multi-class, multi-server systems are a natural extension of the well-known M/M/k systems in queuing theory. The main questions that we will address for multi-class multi-server queueing systems (abbreviated as mc, ms throughout this paper) are:

1. How to construct exact/approximate solutions for such models?
2. How to use these solutions for improved design and management of spare part systems?

Multi-class, multi-server models arise for stochastic processes where clients with different service characteristics ask for service capacity from a non-differentiated resource with finite capacity. This will be sketched in more detail for the case of spare parts management in section 1.1. Next, the main contribution of this paper is discussed. It is on the first issue: the construction of exact stationary solutions for the probability distribution over the states. Background on literature on mc, ms models and their applications will be given to put our results into perspective (section 1.2). After that our approach is briefly surveyed and the relevance of the results for process design and management is summarized (section 1.3).

1.1 Background.

Now to motivate the analysis of multi-class multi-server systems let us have a closer look at spare parts management. Consider a business like an airline, a bus- or train line company, the navy or a lease firm of, say, cars or copiers. Typically, such a business creates its service through a large installed base of systems. Each system in the installed base consists of several components that are failure prone. For example, in case of navy frigates one could think of an engine, pumps and valves in a pipeline network. These different components give rise to a multi-source input to repair shops. It is a major business goal to control the overall system availability. This leads to several logical process design- and management questions, which are illustrated in figure 1.

Spare parts are used for fast replacement of a failed component in one of the M systems. The diagnosis of a component failure determines where to send the component for repair. At the same time, the failed part is immediately replaced by a spare part if it is on stock. If no spare part is available in inventory, one has to wait until repair has taken place at one of the repair shops. This may lead to considerable system downtime (e.g. for the frigate). In this way, it becomes clear that the organization and capacity of the repair shop plays an important role. A natural way is to organize the repair shops according to disciplines, such as mechanical engineering, electronical engineering, etc. Hence a disciplinary repair shop is not dedicated to one sort of components (this in contrast with "item-dedicated" repair shops, cf. Sleptchenko, van der Heijden and van Harten, [12]). In general, the input
Figure 1: How an installed base with multi-component systems gives rise to jobs in multiple classes in disciplinary organized repair shops.

to such a repair shop will still be of multi-source type, now with disciplinary jobs for different types of components. It is quite natural that these different types of jobs from multiple input sources have *different repair time characteristics*. Moreover, it is natural that the capacity in such a disciplinary group is modeled in a non-differentiated way as identical servers. This gives rise to the multi-class multi-server queuing model that is the topic of our paper. It is now clear that the overall system availability depends on the number of spare parts and on the capacity (and other characteristics) of the repair shop. For process design and - management it is crucial to have insight in this relationship.

Each failed component leads to a repair order. We assume that each component can be repaired in exactly one repair shop, depending on the discipline required. That is, we have a fixed assignment of component types to repair shops. In this paper, we will focus on the set of components handled by one repair shop. Now, ignoring transport times, the on-hand inventory of spare parts of a certain type satisfies a well-known identity, cf. Sherbrooke, [6]. At each moment it equals the total number of spare parts of that type corrected with the number of items due in from resupply or repair (both: repair in execution and waiting for repair). The
The availability of the overall system is defined in the usual way (cf. Sherbrooke, [6]) as:

\[ A = E \left[ \prod_i \left( 1 - \frac{1}{M} BO_i \right) \right] \quad (1.1) \]

where \( BO_i \) denotes the number of backorders of item \( i \). The philosophy is that each backorder causes downtime at one of the \( M \) base systems. For simplicity of notation, we herewith assume that each item occurs only once in each system. However, the extension to multiple occurrences is straightforward, cf. [6].

Now Sherbrooke’s analysis is based on infinite repair shop capacity, modeled as \( M/G/\infty \) queues. This means that the number of items in repair is Poisson distributed, according to Palm’s theorem, and this facilitates the analysis. Another consequence of the infinite capacity assumption is that the numbers of backorders of different items \( BO_i \) are mutually independent random variables. Hence the expectation of the product of the backorders equals the product of the expected backorders. In the nonlinear formula (1.1), the overlap of downtime due to different items is taken into account. As it is reasonable for most practical cases to assume that \( BO_i \ll M \), we can simplify equation (1.1) to:

\[ A \approx 1 - \frac{1}{M} \sum_i EBO_i \quad (1.2) \]

where \( EBO_i \) denotes the expected number of backorders of item \( i \). The expected backorders for an item given its spare part level \( I \) are given by:

\[ EBO_i = \sum_{r \geq I} (r - I) \Pr (\text{number of type } i \text{ items in repair } = r). \quad (1.3) \]

We will denote the approximate availability defined by (1.2) as \( A_1 \).

Although this approximation seems adequate for infinite repair shop capacities, it is questionable whether this is also true in the case of finite capacities. The latter means that the number of backorders of different components at the same repair shop are mutually correlated. This is a severe complication, as the expectation of the product of backorders cannot be taken term-wise anymore. Still, equation (1.2) can be seen as a linear approximation of (1.1), and then the correlation doesn’t play a role anymore. The quality of this approximation is still unknown, however. We will also address the consequences of backorder correlation for the availability estimation in this paper (section 7.3). Therefore the analysis of \( mc, ms \) models should provide insight in the joint probability distribution of the number in repair for all items. This will give us both the expectations for backorders per item as well as the correlations between items.

### 1.2 Literature survey.

Let us now give some background on the literature on spare parts management. The work of Sherbrooke, [5], [6] and Slay, [7], especially the VARIMETRIC model, is central in this area, nowadays. They work under the convenient hypothesis of repair
shops with infinite capacities, but on the other hand they include many generalizations in their set of models compared with our previous example, e.g. multi-echelon and multi-indenture structures. The work of Diaz and Fu, [3] allows for finite capacity in repair shops and for general arrival and service processes. It also recognizes the disciplinary organization of repair shops and as such inspired us to our work. But, they have to use approximations in their analysis that we shall improve on here using exact solution techniques for $mc, ms$ problems. Also, as for multi-echelon and multi-indenture structures their theory is restricted, for generalizations and comparison with simulation, we refer to Sleptchenko, et al., [12]. Further Avsar and Zijm, [9] deal with finite capacity in repair shops through analyzing certain queuing network models. The disciplinary, multi-class structure of repair shops is not covered by that work. In Keijzers et al., [10] the case of SEWACO at the Royal Dutch Navy is considered. A multi-server queuing model with two priority classes is used there to analyze the distribution of tardiness of preventive maintenance due to interference of emergency jobs. For application of VARIMETRIC-like models at the Royal Dutch Navy, we refer to Rustenburg, [4]. Altogether, this shows that there is a good potential to use $mc, ms$ models in spare parts management, while the theory is still far from complete.

The analysis of multi-class multi-server problems starts as a generalization of the well-known M/M/k queue. The theory of the M/M/k queue is dealt with in any good textbook on operations research, such as Winston, [15]. In the more general $mc, ms$ context, the M/M/k queue can be considered as a special case where the classes of clients are indistinguishable with respect to their service characteristics. Then, the service rate $\mu (i)$ of class $i$ is equal to the averaged service rate $\mu$. As a matter of fact, all classes can be considered to merge into one class only. In the general case with $N > 1$ classes, $\mu (i)$ has a different value for each class and $\mu (i) = (1 + \delta (i)) \cdot \mu$ where $\delta (i)$ measures the strength of the perturbation relative to the averaged service rate $\mu$. Let us now discuss some relevant literature on multi-class queuing systems. Mostly, there is a focus on the effect of priority classes and priority rules on the performance, cf. Federgruen and Groenevelt, [1] and also, the recent work of Keijzers et al., [10], mentioned before. In our work, there is a first come first serve service discipline for jobs from different classes of clients. In spare parts management this is consistent with the usual way of thinking. Also for a multi-source input process the spare parts levels are optimized in view of their effect on the system availability (cf. Sherbrooke, [6]), while classes have an equal status in terms of priority. In the optimization one presupposes that the management of a repair shop does not allow for emergency repairs, if for some component there is an out of stock situation for that spare part. Even if this rule is changed it wouldn't lead to definite priority classes, but to state dependent priorities. Priority classes can be a further extension of the $mc, ms$ system that is not discussed in this paper.

Another line of research is the application of approximation techniques or asymptotic methods to $mc, ms$ methods. Bertsimas and Mourtzinou, [2] consider heavy traffic asymptotics based on some conservation laws and order preservation assump-
tions. Diaz and Fu, [3] give some approximations mainly for the single server case. Basic to the reasoning underlying their approximations is that the expected waiting time is class independent, see also Whitt, [8] and moreover, that during full utilization the distribution of clients in service is related in a simple way to the class utilization fractions. It will be shown here in our work that the exact solution of $mc, ms$ problems has a more complex structure and, of course, it also leads to a better accuracy.

Finally we mention the work of Adan and van der Wal, [11]. They relate certain properties of a two-class system to those of a single class system with a more general service process. Besides the generalization to multi (>2)-class systems it does not provide the sort of information on backorder distributions per class that is required in applications.

1.3 Approach

Let us now briefly sketch our approach. Here we focus on the stationary probability distribution over the states of a general $mc, ms$ system with $\delta (i) \neq 0$. Of course, the state space is necessarily considerably more complex than in the M/M/k case with all $\delta (i) = 0$, see section 2. The state definition has to take into account the types of jobs in execution and the ordering of job types in the queue. The state equations can be given explicitly in the general case, see section 3. For $\delta (i) = 0$ the solution can be given, see section 4. Only a few structural aspects of this solution survive the perturbation to $\delta (i) \neq 0$. It is crucial that in the general case, the distribution over the ordering in the queue keeps a product structure based on the arrival fractions of the client classes. Using this structure for $n = \text{number of clients in the system} > k$, we obtain a much more transparent structure for the state equations. For $n > k$ they reduce to a backwards second order difference equation for a vector in a linear space of dimension $d(N, k) = (N + k - 1)! / \{k! (N - 1)!\}$. For $N = 3$ and $k = 4$ we find $d(N, k) = 15$. Of course this dimension increases rapidly with $N$ and $k$. Now difference equations can be solved in terms of eigenvalues and (generalized) eigenvectors of the iteration matrix. All eigenvalues and eigenvectors can be determined explicitly if $\delta (i) = 0$. Precisely $d(N, k)$ of the eigenvalues are real and $> 1$ in the unperturbed case, which corresponds with decay for $n \to \infty$. This property survives the perturbation to $\delta (i) = 0$. It is the clue to solve the state equations, first for $n > k$ with the start vector for $n = k$ as an unknown and next, again with backwards recursion, also for $n \leq k$, until only one scaling constant, the probability of the empty state, remains. That scaling constant follows from the fact that the state probabilities sum to 1.

Using this procedure the exact solution follows, see section 5, and its structure can be analyzed. Next, coming back to the second question in the beginning, all sorts of interesting performance criteria for $mc, ms$ systems can be determined exactly, see section 6. Examples of such performance indicators are the expected number of items of type $i$ in the system or in queue, its variance, the expected number of
backorders of type \( i \), its variance, the expected waiting time for clients in a certain class. Their behavior is discussed in section 7.

To conclude this introduction we summarize the main relevance of these results for \( mc, ms \) queues for process design and management questions:

- More accurate insight in the performance differences between different classes of clients
- Better insight in the relation between capacity and performance when several classes of clients compete for capacity, hence correlation between classes occurs.

In spare parts management, both issues play an important role. Potential applications are also found in other areas such as manufacturing and factory layout.

2 Multi-class, multi-server queues: definitions and notation.

In figure 2 we visualize the \( N \) different job types originating from multiple sources, by arrows with different line types, and their flow through the \( k \) identical servers.

Jobs with type \( i \) arrive according to a Poisson process with arrival rate \( \lambda(i) \). Due to the non-dedicatedness of the servers, the flow through a server consists of all job types. The service distribution of job type \( i \) is exponential with rate \( \mu(i) \) and the
same for all servers. The service discipline by which jobs are assigned is first come first serve and if more than one server is available, say $\kappa$ of them are available, then each of them has an equal probability $1/\kappa$ to get the next job. Note that the latter assumption is only used to derive the state equations. Other server selection criteria (e.g. lexicographically) should lead to the same expressions for the performance measures.

The total arrival rate is given by $\Lambda = \sum \lambda(i)$. We denote the arrival fraction of class $i$ by $a(i) \overset{\text{def}}{=} \lambda(i)/\Lambda$. The utilization is represented by $\rho$, hence the stability condition for this queuing system is given by $\rho < 1$. The average service rate $\mu$ is defined by $1/\mu \overset{\text{def}}{=} \sum a(i)/\mu(i)$ and after some simple calculation we obtain the equivalent expression $\mu = \Lambda/(k\rho)$ as given in the figure. The notation for the relative perturbation from the averaged service rate is $\delta(i)$ as already introduced in the introduction.

Let us now have a closer look at the definition of the states of the system. The notation is summarized in the box here below.

<table>
<thead>
<tr>
<th>State: $(w, s)$ with</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w$: a $p$-vector of jobs in queue with components $w(i)$;</td>
</tr>
<tr>
<td>$w(i)$ refers to the class of the $i$-th job in queue</td>
</tr>
<tr>
<td>$s$: a $k$-vector of jobs in service with components $s(i)$;</td>
</tr>
<tr>
<td>$s(i)$ refers to the class of the job in execution at server</td>
</tr>
<tr>
<td>notation: $w(i), s(i) \in {0, 1, \ldots, N}$, 0 refers to the empty, no job mode</td>
</tr>
<tr>
<td>$p = \max(0, n - k)$: number of waiting clients</td>
</tr>
<tr>
<td>$P(w, s)$: stationary probability distribution over the states for $\rho &lt; 1$</td>
</tr>
</tbody>
</table>

Note that we use a concept of state based on the arrival sequence of the jobs in queue and the assignment of jobs to servers. Of course, though conceptually simple, this definition contains some redundancy. The stationary state probabilities should satisfy certain symmetries, namely: invariance under permutations of a given job sequence in queue as well as invariance under permutations in the assignment of a given job set to servers. For efficiency reasons in computations we use the reduced state space taking the invariance into account.

| $s^*$: lexicographically ordered job assignment to servers |
| $w^*$: lexicographically ordered job queue |

Here lexicographical ordering is defined as increasing job type index with an increasing index of the server or the place in queue, respectively. Note that $\chi[s^*]$, the number of job assignments $s$ equivalent to $s^*$ under permutation, equals
\[ k! / \{ k(0; s^*)! \ldots k(N; s^*)! \} \text{ with } k(i; s^*) \text{ the number of servers occupied by jobs of type } i \text{ in state } s^*. \]

Analogously, the number of queue sequences equivalent to \( w^* \) is given by \( \chi_{w^*} = p! / \{ k'(0; w^*)! \ldots k'(N; w^*)! \} \) with \( p \) the queue length and \( k'(i; w^*) \) the number of jobs of type \( i \) in queue. It is also worth to mention that the total number of different lexicographically ordered assignments \( s^* \) with all servers busy is \( d(N, k) = (N + k - 1)! / \{ k!(N - 1)! \} \), a relation that can be proved using induction to \( N \). This is relevant for the dimension of the difference equation in section 5.

Now under the stability condition \( \rho < 1 \) there exists a unique stationary probability distribution \( P(w, s) \) and \( P[w^*, s^*] \) over the states and lexo-states, respectively. The brackets \( [ \ ] \) are just another reminder of the reduced state space. It is clear that \( \sum P(w, s) = \sum \chi_{w^*}\chi_[s^*]P[w^*, s^*] = 1. \)

Let us now have a closer look at the stationary state equations satisfied by \( P. \)

### 3 Stationary state equations for \( mc, ms \) problems.

The state equations follow from a micro-balance reasoning illustrated in figure 3. The net exchange of probability in an infinitesimal time interval from a given state with its neighbors has to be zero in an equilibrium situation. Neighbors of a state \((w, s)\) with \( n \) clients are states to or from which a one step transition is possible either by an arrival event (A) or a service completion event (C). Neighbor states have \( n - 1 \) or \( n + 1 \) clients. If \( n > k \), transition from/to neighbors \((w', s')\) with \( n + 1 \) clients are specified by one of the following events:

(a) transitions from \((w, s)\) to \((w', s')\): a job arrives and the queue vector is expanded as \( w' = (w, 1) \) if the job type is \( l \)

(b) transitions from \((w', s')\) to \((w, s)\): a service has been completed; if the service completion occurs at server \( j \), the first job in queue should have type \( l = s(j) \); hence the state \((w', s')\) is specified by \( w' = (s(j), w) \) and \( s' = E^1_l s \), where the operator \( E^1_l \) changes the contents of the \( j \)-th vector component into job type \( l \).

Analogously, if \( n > k \), then transitions from/to neighbors with \( n - 1 \) clients are specified by:

(c) transitions from \((w, s)\) to \((w', s')\): a job at server \( j \) has been completed and the first job in queue moves to server \( j \); denoting the first job in queue by \( l = w(1) \), we have \( s' = E^1_l s \); further all other jobs in queue are shifted one position in the queue, so \( w'(i) = w(i + 1) \) for \( i \leq p - 1 \) and \( w'(p) = 0 \);

(d) transitions from \((w', s')\) to \((w, s)\): arrival of a job with type \( w(p) \), the last job in queue in the state \((w, s)\); then \( w' = E^p_0 w \) (omitting the last job in queue) and \( s' = s \);

The characterization of neighbors for states with \( n \leq k \) can be done analogously, the details on the neighbors are self-evident from the terms in the stationary state equations given below.

Before stating the stationary state equations, we first introduce the following
Figure 3: The micro-balance between a state and its neighbors leads to the stationary state equations.

notation. Let $T$ be the left shift operator changing a $p$–vector into a $(p - 1)$–vector with $T w(i) = w(i + 1)$. Further, we denote by $\text{Length}(v)$ the number of nonzero entries in the vector $v$. Also, we define:

$$\overline{\delta}(s) \overset{\text{def}}{=} \frac{1}{k} \sum_{s(j) \neq 0} \delta(s(j))$$

Now we can formulate the stationary state equations using the scaling as introduced in section 2:

- $\text{Length}(w) = p \geq 1$

$$\begin{align*}
(1 + \rho + \overline{\delta}(s)) P(w, s) = \\
\rho a(w(p)) P(Tw, s) + \frac{1}{k} \sum_{j=1}^{k} \sum_{l=1}^{N} (1 + \delta(l)) P((s(j), w), E_{l}^{j}s)
\end{align*}$$

- $0 < \text{Length}(s) = n < k^{1}$

$$\left(\frac{n}{k} + \rho + \overline{\delta}(s)\right) P(0, s) =$$

$$\begin{align*}
(k - n + 1)^{-1} \rho \sum_{j:s(j) \neq 0} a(s(j)) P(0, E_{0}^{j}s) + \frac{1}{k} \sum_{j:s(j) = 0} \sum_{l=1}^{N} (1 + \delta(l)) P(0, E_{l}^{j}s)
\end{align*}$$

$^{1}$Note that the factor $(k - n + 1)^{-1}$ at the right hand side of this equation arises from the random choice of an arriving job between $(k - n + 1)$ available servers
• Length(s) = k, Length(w) = 0

\[(1 + \rho + \bar{\delta}(s))P(0, s) = \rho \sum_{j=1}^{k} a(s(j)) P(0, E_0^j s) + \frac{1}{k} \sum_{j=1}^{k} \sum_{l=1}^{N} (1 + \delta(l)) P(s(j), E_l^j s)\]  

\[\text{Length}(s) = 0\]

\[\rho P(0, 0) = \frac{1}{k} \sum_{j=1}^{k} \sum_{l=1}^{N} (1 + \delta(l)) P(0, E_l^j 0)\]  

For further analysis one should notice that if we restrict ourselves to lexicographically ordered states \((w^*, s^*)\), then the equations are still valid "mutatis mutandis". One only has to interpret \(E_l^j s^*\), which is not automatically lexicographically ordered if \(s^*\) is so, as it is the equivalent state after permutation, which is lexicographically ordered. Using the structure of the arrival process, the equation (3.1) for \(p = n - k > 0\) can be further reduced. For \(p\) consecutive arrivals, the probability distribution over the possible job type sequences in queue is given by the product of the arrival fractions.

As a consequence, we may write de steady state distribution \(P(w, s)\) as:

\[P(w, s) = P_n(s) \prod_{i=1}^{p} a(w(i))\]  

where the unknown vector \(P_n(s)\) represents the stationary probability distribution over de server states \(s\), given that the system contains \(n\) jobs (in service plus in queue). An analogous equation applies to \(P_n[s^*]\), representing the similar distribution for the lexicographically ordered states \(s^*\). A key issue for the sequel will be how to derive an expression for \(d(N, k)\)-vector \(P_n\) with components \(P_n[s^*]\).

Firstly, we note that substitution of (3.6) in (3.1) and restriction to the lexicographically ordered states \(s^*\) provides us with the following equation for \(n > k\):

\[((1 + \rho) I + \bar{\delta}) P_n = \rho P_{n-1} + A P_{n+1}\]  

Here \(I\) denotes the identity matrix and \(\bar{\delta}\) represents a diagonal matrix with elements \(\bar{\delta}[s^*]\) on the diagonal, hence

\[\bar{\delta}[s^*] = \bar{\delta}[s^*] \xi[s^*]\]

Also, the operator \(A\) is defined on an arbitrary \(d(N, k)\)-vector \(\xi\) by:
The equation (3.6) is a second order difference equation for $P_n$ in a $d(N, k)$
dimensional linear space, which plays a central role in our analysis. Since the matrix
$A$ is not invertible (see the next section), one should read (3.6) as a backward
recursion for decreasing $n$.

Note that $P_n$ can also be defined for $n \leq k$. It is a vector of dimension $d(N, n)$,
since only $n$ of the servers are occupied. Let $\mathcal{L}_n$ denote the corresponding linear
space of dimension $d(N, n)$ accommodating that vector. The equations (3.2-3.3) are
equivalent to:

$$P_n = \frac{\rho}{k - n + 1} D_n F_n P_{n-1} + D_n B_n P_{n+1}$$

(3.6)

Where the operators $D_n$, $F_n$ and $B_n$ on the $d(N, n)$-vector $\xi [s^*]$ are respectively
defined as:

$$D_n : \mathcal{L}_n \rightarrow \mathcal{L}_n, \quad D_n \xi [s^*] = \left( \frac{n}{k} + \rho + \overline{\delta}(s^*) \right)^{-1} \xi [s^*]$$

$$F_n : \mathcal{L}_{n-1} \rightarrow \mathcal{L}_n, \quad F_n \xi [s^*] = \sum_{j=1}^{n} a (s^*(j)) \xi [E_j^0 s^*]$$

$$B_n : \mathcal{L}_{n+1} \rightarrow \mathcal{L}_n, \quad B_n \xi [s^*] = \frac{k - n}{k} \sum_{j=1}^{n} (1 + \delta (j)) \xi [E_j^k s^*] \quad \text{for } n < k$$

For $n = k$ we get $B_n = A$ operating between $d(N, k)$ dimensional linear spaces.
We come back to solving these equations by backwards recursion in section 5.

4 M/M/k revisited: detailed information on the unperturbed case.

In the unperturbed case it is possible to analyze the system using states $n$ which
show the number of clients in the system without type differentiation. The transition
diagram is quite simple (figure 4).

The state equations for the probability distribution over $n$ are given by:

$$P(n) = \max \left( 1, \frac{k}{n} \right) \rho P(n - 1)$$
and the solution is found in a straightforward way:

for \( n \geq k \):

\[
P(n) = \rho^{n-k} P(k)
\]

for \( n < k \):

\[
P(n) = \frac{1}{n!} (k \rho)^n P(0)
\]

with \( P(0) \) such that the probabilities sum to 1.

Now in case \( \delta(i) = 0 \), the multi-class probabilities \( P(w, s) \) are just a multinomial modification of the previously found \( P(n) \):

for \( n \leq k \):

\[
P(0, s) = P(n) \binom{k}{n}^{-1} \prod_{j : s(j) \neq 0} a(s(j))
\]

for \( n > k \):

\[
P(w, s) = P(n) \prod_{i=1}^{p} a(w(i)) \prod_{j=1}^{k} a(s(j))
\]

This can be checked by substitution in the state equations and with some computational effort it can be seen that they are satisfied.

But, a lot more information about other solutions of the state equations can be derived. This is useful for the exploration of the general case in the next section. First we observe that, if all \( \delta(i) = 0 \), the eigenvalues and eigenvectors of the matrix \( A \) can be explicitly found. Let us first consider the case where symmetry under permutation is ignored and \( A \) is a matrix in a \( N^k \) - dimensional linear space. The eigenvalues of \( A \) are real; they are given by \( \lambda = m/k \) with \( m = 0, 1, \ldots, k \). The corresponding eigenspace in this \( N^k \) - dimensional linear space is given by:

\[
E_{m/k, \text{full}} = \left\{ q : q(s) = \prod_{j \in \Theta(m)} a(s(j)) \prod_{j \notin \Theta(m)} e_j(s(j)) \right\}
\]
Here $e_j$ denotes some $N-$ vector perpendicular to $1 = (1, 1, \ldots, 1, 1)^t$ and $\theta (m)$ is some subset of $\{1, 2, \ldots, k-1, k\}$ with $m$ elements. It suffices to choose $e_j$ from a basis set of $N - 1$ linearly independent vectors spanning the orthogonal complement of $1$. Say those vectors $(1^\perp)_i$ where $i$ runs from 1 to $N - 1$ with 1 at entry $i$ and $-1$ at entry $i + 1$ and further 0 entries. Note that the dimension of the full eigenspace equals:

$$d_{m/k, \text{full}} = \binom{k}{m} (N - 1)^{k-m}$$

Of course, one can check these eigenspaces by substitution using the definition of $A$. All eigenvalues and eigenvectors of $A$ are found, since the total multiplicity of the eigenspaces adds up to $N^k$. It is now clear that restricted to the subspace of states symmetric under permutation, which is invariant under $A$, the same eigenvalues are found, but now with a restricted set of eigenvectors. They should be such, that for each state $s$ equivalent to $s^*$ after permutation, the identity $q(s) = q(s^*)$ holds true. Such "symmetric" eigenvectors are obtained from the previous ones by summation over all choices for $\theta (m)$ and, for given $\theta (m)$, summation over all possible ways $r$ to map the complement of $\theta (m)$ into a fixed set $e_j$ with $j = 1, \ldots, k-m$. Hence, choose $k - m$ elements $(1^\perp)_{I(i)}$ from a basis of $1^\perp$ where $I(i)$ increases lexicographically with $i$ then the symmetric eigenspace is given by:

$$E_{m/k, \text{symm}} = \left\{ q : q [s^*] = \sum_{\theta} \sum_{r : \theta r^c - I j \in \theta} \prod_{j \notin \theta} a(s^*(j)) \prod_{j \notin \theta} e_{r(j)}(s^*(j)) \right\}$$

For notational simplicity, we suppressed the $m$ dependence in $\theta$ and its complement $\theta^c$. Note that the freedom in the eigenspace reduces to choosing $k - m$ indices from $1, \ldots, N - 1$ in lexicographical order. This leads to the following dimension:

$$d_{m/k, \text{symm}} = \binom{N - 2 + k - m}{k - m}$$

Note that the sum of the multiplicities equals $d(N, k)$, due to a well-known combinatorial identity. The conclusion is that also in symmetric subspace the multiplicity of the eigenvalues add up to the space dimension, hence all eigenvalues and eigenvectors of $A$ have been found.

In the unperturbed situation, each of these eigenvalues and a corresponding eigenvector of $A$ gives rise to solutions of the second order difference equation (3.6) for $n > k$. Looking for solutions of the form

$$P_n = z^{-n}q$$

with $q$ an eigenvector for the eigenvalue $m/k$ we obtain a quadratic equation that $z$ has to satisfy:

$$pz^2 - (1 + \rho) z + \frac{m}{k} = 0$$
Hence we obtain:

\[
z = \frac{1}{2\rho} \left\{ (1 + \rho) \pm \sqrt{(1 + \rho)^2 - 4\frac{m}{k}\rho} \right\}
\]

Note that the + sign leads to \( z > 1 \) and the − sign to \( z \leq 1 \). Only the solutions with \( z > 1 \) decay for \( n \to \infty \) and are acceptable in constructing probability distributions. There are \( d(N, k) \) of such solutions. Only one of them, namely with \( m = k \) plays a role in the exact solution for the unperturbed case. But, in this respect the unperturbed case turns out to be special.

Of course one should expect that in a general case with \( \delta(i) \neq 0 \) the situation will be more complex.

5 Solving general \( mc, ms \) problems exactly.

Let us first consider the construction of solutions of the state equations for \( n > k \) in general and then for the remaining equations for \( n \leq k \) (Section 5.1) Next, in Section 5.2 the full solution of the state equations will be constructed. In section 5.3 we consider some special cases.

5.1 Exact solutions for \( n > k \).

Let us first reformulate the second order difference equation for the \( d(N, k) \) vector \( P_n \) given in (3.6) as a first order difference equation for a \( 2d(N, k) \) dimensional vector. Now let us consider the vector \( P_n = (P_{n-1}, P_n)^t \). It has to satisfy \( P_n = H P_{n+1} \) with the matrix \( H \) given by

\[
H = \begin{pmatrix}
\frac{1}{\rho} ((1 + \rho) I + \delta) & -\frac{1}{\rho} A \\
I & 0
\end{pmatrix}
\]

Note that solving this backward recursion boils down to determining the eigenvalues and eigenvectors of the matrix \( H \). Compared with the previous section, the main difference is that all solutions of (3.6) that decay for \( n \to \infty \) will play their part in the solution of the state equations. The next section is about finding the decay solutions for \( n > k \). The following information is crucial.

Lemma 1

1. Under the stability condition with all \( \delta(i) \) sufficiently small, \( H \) has:

   - eigenvalues \( z \) satisfying \( |z| \leq 1 \) with total multiplicity \( d(N, k) \)
   - eigenvalues \( z \) satisfying \( |z| > 1 \) with total multiplicity \( d(N, k) \)
2. If \( \delta(i) \neq 0 \), \( A \) has eigenvalues \( \alpha_i m / k \), with \( \alpha = \sum_{i=1}^{N} a(i)(1 + \delta(i)) \), \( m = 0, \ldots, k \) with the same multiplicities and similar eigenspaces as in the unperturbed case. As a consequence, \( 0 \) is an eigenvalue of \( H \) with eigenspace \( (0, \ker(A))^t \) and the same multiplicity \( d(N - 1, k) \) as before.

3. \( N + 1 \) special eigenvalues of \( H \) corresponding with eigenvectors possessing a product structure can be found:

- there is an eigenvalue \( 1 \) with eigenvector \( (B, B)^t \) with:
  \[
  B[s^*] = \prod_{j=1}^{k} \frac{a(j)}{(1 + \delta(j))} 
  \]
  \[ (5.1) \]

- under the stability condition there are \( N \) eigenvalues
  \[
  z(\sigma) = \frac{(1 + \rho - \sigma)}{\rho} > 1
  \]
  \[ (5.2) \]
  where \( \sigma \) satisfies a polynomial equation of degree \( N + 1 \):
  \[
  1 + \rho - \sigma = \rho \sum_{i=1}^{N} a(i) \frac{1 + \delta(i)}{\sigma + \delta(i)}
  \]
  \[ (5.3) \]
  which also has the solution \( \sigma = 1 \), i.e. \( z(1) = 1 \) as above, see figure 5. The eigenvector for \( z(\sigma) \) is \( (z(\sigma)C, C)^t \) with:
  \[
  C[s^*] = \prod_{i \in s^*} \frac{a(i)}{(1 + \delta(i) / \sigma)} 
  \]
  \[ (5.4) \]

One of these eigenvalues \( z(\sigma_0) \) crosses \( z = 1 \) into \( z < 1 \) if \( \rho \) crosses \( 1 \) into the region \( \rho > 1 \) where the stability condition is violated.

**Proof.** To start with we note that \( z = 1 \) is an eigenvalue of \( H \) corresponding with the given eigenvector. This follows simply by substitution. Note that in the eigenvector the probabilities over the states are proportional to the products of the required service fractions. Now the first part of the lemma is simply a consequence of the well-known perturbation theory for eigenvalues of matrices, [16]. Next we observe that also in the perturbed case the eigenvalues of \( A \) are \( \alpha_i m / k \) with \( m = 0, \ldots, k \), but now the eigenspaces are slightly different. They are found by using a \( N \)-vector \( e_j \) perpendicular to \( 1 + \delta = (1 + \delta(1), \ldots, 1 + \delta(N))^t \) instead of the expressions in section 4. Hence, the dimension of the eigenspaces corresponding with \( m/k \) is the same as before.

The remainder of the lemma is a matter of substituting the specified eigenvectors in the eigenvalue equation and checking that it is satisfied. Due to the special structure of the eigenvectors, this is straightforward, therefore the details are left to
Figure 5: Graphical solution of the polynomial equation for some special eigenvalues of $H$.

the reader. The behavior of the crucial eigenvalue follows immediately from figure 6 by noticing that the derivative of the right hand side at $\sigma = 1$ equals $-\rho$. ■

Let us now discuss the consequences of the lemma. First, it should be noted that in the case $k = 1$ (a) there are $N - 1$ eigenvalues $0$ for any $N$, (b) there is an eigenvalue $1$ and (c) there are $N$ real eigenvalues $> 1$ given by the special polynomial equation. This provides complete information on the eigenvalues. In other cases, other eigenvalues besides the special ones play a role. In figure 5 herebelow, it is shown how the eigenvalues evolve from their unperturbed values, if the perturbation is "turned on".

To plot the changes of eigenvalues we use a case with $N = 4$, $k = 3$, $\rho = 0.9$. To change $\delta (i)$ we use function $\delta (i) = \alpha (i) t$, with $\alpha (2) = 0.1$, $\alpha (3) = 0.15$, $\alpha (4) = 0.2$, and $\delta (1)$ is chosen in the way to guarantee that $\sum_{i=1}^{N} \frac{a(i)}{1+\delta (i)} = 1$. In table 1 we present the perturbed $\delta (i)$ for $t = 0, \ldots , 9$, and next we present how the eigenvalues change with these perturbations.

Let us now use the information on the eigenvalues and eigenspaces of $H$ to solve the state equations for $n \geq k$ exactly in terms of the state probabilities for $n = k$. 

16
This can be done using the following recipe. Let us use the standard notation $d = d(N, k)$.

**Introduce:**

- $\Omega$ as the $d \times d$ diagonal (or Jordan) matrix corresponding with the eigenvalues of $H$ with $|z| > 1$
- $\Xi$ as the $d \times d$ matrix of upper parts of the corresponding (generalized) eigenvectors
- Note that $Z = \Xi \Omega \Xi^{-1}$ satisfies
  
  $$(1 + \rho) I + \bar{\delta} = A Z^{-1} + \rho Z$$
- **NOW**

  $$P_n = (Z^{-1})^{n-k} P = \Xi \Omega^{n-k} \Xi^{-1} P$$

  is an exact solution of (3.6) for $n \geq k$ starting at $P$ for $n = k$.

### 5.2 The exact solution for $n \leq k$

Consider $P_n[s^*]$ for $n < k$ with $P_n$ in $L_n$ as in (3.6). Again the state equations for $n < k$ have a backward recursion structure.

The equation for $n = k$:

$$(1 + \rho + \bar{\delta}) P_k = \rho F_k P_{k-1} + A Z^{-1} P_k$$

reduces to:

$$P_k = Z^{-1} F_k P_{k-1}$$
Figure 6: A sketch of the behaviour of the eigenvalues of $H$ with the strength of the perturbation.

by using the equation for $Z$.

For $n < k$ we obtain:

$$P_n = \frac{\rho}{k - n + 1} D_n F_n P_{n-1} + D_n B_n P_{n+1}$$

Hence the complete solution can now be represented as:

$$P_n = Q_n P_{n-1} = Q_n Q_{n-1} \cdots Q_1 P_0$$

where $Q_n$ follows recursively from

$$Q_k = Z^{-1} F_k$$

and

$$Q_n = \frac{\rho}{k - n + 1} (I - D_n B_n Q_{n+1})^{-1} D_n F_n$$

Of course $P_0$ is a free constant determined by

$$\sum P(w, s) = 1.$$ 

A simple computation shows that

$$P_0 = \left\{ (1 + \langle \chi, Q_1 \rangle_1 + \cdots + \langle \chi, Q_{k-1} \cdots Q_1 \rangle_{k-1} + \langle \chi, (I - Z^{-1})^{-1} Q_k \cdots Q_1 \rangle_k \right\}^{-1}$$

Here we denote with $\langle \chi, X \rangle_n$ the innerproduct of $X$ in a $d(N,n)$-dimensional space with the $d(N,n)$-vector $\chi$ with components equal to $\chi[s^*]$ the number of states $s$ equivalent to $s^*$.

Thus, the exact solution for all $n$ has been derived.
5.3 Some special cases

To illustrate the theory we consider two cases where the algorithm can be executed more explicitly: (I) the case $k = 1$, (II) the case $k = 2, N = 2$.

In the case $k = 1$ we have states $s = 1, \ldots, N$ if the server is occupied with $s$ indicating the type of job in execution. The eigenvalues are given by $z(\sigma(j))$, $j = 1, \ldots, N$ as defined by equation (5.2-5.3); the matrix of eigenvectors $\Xi$ has elements: $a(i)/(1 + \delta(i)/\sigma(j))$. The solution of (3.6) is given by:

$$P_n = P_0 Z^{-n} a$$

Here $a$ denotes the $N$-vector with components $a(i)$. As an example we consider the case of 2 classes. Then the eigenvalues are: $z_1 = 0$, $z_2 = 1$ and

$$z_{3,4} = \frac{1}{2\rho} \left[ (\rho + (1 + \delta(1)) + (1 + \delta(2))) 
\pm \sqrt{\rho^2 - 4\rho \left( \sum_{i=1}^{2} a(i) (1 + \delta(i)) \right) + \rho (2 + \sum_{i=1}^{2} (1 + \delta(i))) + (\delta(1) - \delta(2))^2} \right],$$

and also we can obtain the following explicit expressions for $Z$ and $P_0$:

$$Z = \begin{pmatrix} \frac{1}{\rho} ((1 + \delta(1)) + a(2) \rho) & -a(1) \\ -a(2) & \frac{1}{\rho} ((1 + \delta(2)) + a(1) \rho) \end{pmatrix}, \quad P_0 = 1 - \rho$$

In case $k = 2$, $N = 2$ we start with the observation that the special eigenvalues in (5.2-5.3) can be explicitly computed, because the degree of the equation is 3 and the solution $\sigma = 1$ is already known. This leads us to the following quadratic equation for $z$:

$$0 = Q_1(z)
= \rho^2 z^2 - \rho \{ 2 + \rho + \delta(1) + \delta(2) \} z + \{ 1 + \rho + \delta(1) + \delta(2) + (1 - \rho) \delta(1) \delta(2) \}$$

Now, only 2 of the 6 eigenvalues of $\mathbb{H}$ are still unknown; they can be found from the characteristic polynomial $0 = Q(z) = det(-(1 + \rho + D)z + \rho z^2 I + A)$ by dividing out the known eigenvalues. Here we denote by $D$ the diagonal matrix with on the diagonal the averaged perturbation $\delta$ for the respective states. Hence we decompose $Q(z) = z(z - 1)Q_1(z)Q_2(z)$. Therefore the remaining 2 eigenvalues have to satisfy a quadratic equation with coefficients that can be explicitly given in terms of the original matrix coefficients as:

$$0 = Q_2(z)
= \rho z^2 - \rho \left\{ 1 + \rho + \frac{1}{2} (\delta(1) + \delta(2)) \right\} z + \frac{1}{2} \left\{ 1 + a(1) \delta(1) + a(2) \delta(2) \right\}$$

Note that the latter equation gives rise to 1 eigenvalue $z > 1$. 

19
Now $\mathbf{Z}$ is a $3 \times 3$ matrix operating on the probabilities of the states $(1,1)$, $(1,2)$, $(2,2)$ for the jobs in execution when the servers are occupied. The column of $\Xi$ corresponding with an eigenvalue $z = (1 + \rho - \sigma)/\rho$ becomes $(v, 1, 1/v)^t$ with $v = a(1)(\sigma + \delta(2))\{a(2)(\sigma + \delta(1))\}$. For the other eigenvector corresponding with $z > 1$ satisfying $Q_2(z) = 0$ we find $(-a(1)(1 + \delta(2))/q(z), 1, a(2)(1 + \delta(1))/q(z))^t$ with $q(z) = \{(\delta(2) - \delta(1))/2\}z + \{a(1)(1 + \delta(1)) - a(2)(1 + \delta(2))/2\}$. Note that one of the components of this eigenvector is negative. Apparently this can happen even though the probabilities will eventually be positive. It is clear that also in this case the solution can now be given explicitly in terms of the parameters of the problem.

6 Accurate information on performance measures.

Here we shall provide some further insight in the behavior of interesting performance measures related to stochastic variables such as the waiting time, the queue length $q$ and the number of items of type $i$ in queue $q_i$. Performance measures that we consider are expectations, variances and probability for positivity of these stochastic variables. Moreover, since it is clear that there is correlation between these stochastic variables we also shall provide formula’s for some conditional expectations and correlation coefficients. Of course, the clue that such exact formula’s can be easily derived lies in the fact that series of geometric type can simply be summed. To start with we notice that:

\[
P(q > 0) = \langle x,(\mathbf{Z} - \mathbf{I})^{-1}\mathbf{P}_k \rangle
\]

\[
P(w > 0) = \langle x,(\mathbf{Z} - \mathbf{I})^{-1}\mathbf{ZP}_k \rangle
\]

\[
P(q_i > 0) = \langle I_i x,(\mathbf{Z} - \mathbf{I})^{-1}\mathbf{P}_k \rangle
\]

and:

\[
E(q) = \langle x,(\mathbf{Z} - \mathbf{I})^{-2}\mathbf{ZP}_k \rangle
\]

\[
E(q(q-1)) = E(q^2) - E(q) = \text{var}(q) - E(q) + E(q)^2 = 2 \langle x,(\mathbf{Z} - \mathbf{I})^{-3}\mathbf{ZP}_k \rangle
\]

\[
E(q) = \Lambda^{-1}E(q)
\]

\[
E(q_i | q = n - k) = a(i)(n - k)
\]

\[
E(q_i) = a(i)E(q)
\]

\[
E(q_i(q_i - 1) | q = n - k) = a(i)^2(n - k)(n - k - 1)
\]

\[
E(q_i(q_i - 1)) = \text{var}(q_i) - E(q_i) + E(q_i)^2 = a(i)^2E(q(q-1))
\]
\[ E(q_i, q_j | q = n - k) = a(i) a(j) (n - k) (n - k - 1) \quad \text{for } i \neq j \]
\[ E(q_i, q_j) = a(i) a(j) E\left(q(q - 1)\right) \quad \text{for } i \neq j \]
\[ \text{corr}(q_i, q_j) = E(q_i q_j) - E(q_i) E(q_j) \]

Here we use the notation \( \chi \) as before in determining \( P_0 \). The matrix \( I_i \) is diagonal with 1 for each state \( s^* \) that contains type \( i \) and 0 elsewhere. Note that Little’s law is satisfied and the expected waiting time doesn’t depend on the type of item.

In an analogous way we can analyze the stochastic variables \( R \), the number of items in the system, \( R_i \), the number of items of type \( i \) in the system, \( S_i \), the number of items of type \( i \) in service (i.e. \( R_i = S_i + q_i \)) and \( BO_i \), the number of backorders for type \( i \), \( BO_i = \max(0, R_i - I_i) \) with \( I_i \) a given positive integer. Now we obtain:

\[
P(R_i = m | R < k) = \sum_{n=m}^{k-1} P(R_i = m | R = n) = \sum_{n=m}^{k-1} \langle I_i, m \chi, P_n \rangle
\]
\[
P(R_i = m | R \geq k) = \sum_{r=\max(m-k,0)}^{m} \sum_{n=k+r}^{\infty} P(q_i = r | R = n) P(R_i - q_i = m - r | R = n)
\]
\[
= \sum_{r=\max(m-k,0)}^{m} \sum_{n=k+r}^{\infty} \left( \begin{array}{c} n-k \\ r \end{array} \right) a(i)^r (1 - a(i))^{n-k-r} \langle I_i, m-r \chi, Z^{-(n-k)} P_k \rangle
\]
\[
= \sum_{r=\max(m-k,0)}^{m} a(i)^r \sum_{l=r}^{\infty} \left( \begin{array}{c} l \\ r \end{array} \right) \langle I_i, m-r \chi, (1 - a(i))^{-l-r} Z^{-r} P_k \rangle
\]
\[
= \sum_{r=\max(m-k,0)}^{m} a(i)^r \langle I_i, m-r \chi, (Z - (1 - a(i)) I)^{-r-1} Z P_k \rangle
\]
\[
P(R_i = m) = P(R_i = m | R < k) + P(R_i = m | R \geq k)
\]
\[
P(R_i = m \land R_j = l) = \sum_{n=m+l}^{k-1} \langle I_i, m I_j, l \chi, P_n \rangle
\]
\[
+ \sum_{r,s \in V(m, l)} \left( \begin{array}{c} r+s \\ r \end{array} \right) a(i)^r a(j)^s \langle I_i, m-r I_j, l-s \chi, (Z - (1 - a(i) - a(j)) I)^{-r-s-1} Z P_k \rangle
\]

Here we use the notation \( I_i, m \) for the diagonal matrix indicating the states with \( m \) items of type \( i \) with 1 and 0 elsewhere. In this derivation we use \( r \) to count the
number of type $i$ in queue. In the joint distribution $s$ counts the number of type $j$ in queue. In that case the summation runs over a domain $V(m, l)$ where $r$ runs from $\max(0, m - k)$ to $m$ and $s$ runs from $\max(0, l - k)$ to $l$ while $(m - r) + (l - s) \leq k$. The precise information on the distribution of the numbers of type $i$ and $j$ in the system will play a role in the backorder computations in the sequel.

Some other quantities are simpler to derive. Using Little’s theorem we find the first moments:

$$E(R_i) = E(q_i) + \lambda(i)/\mu(i)$$

$$E(S_i) = \lambda(i)/\mu(i)$$

Moreover, we can now easily derive the following second moments and correlation coefficients:

$$E(R_i^2) = E(q_i^2) + E(S_i^2) + 2E(S_i q_i)$$

$$E(S_i^2) = \sum_{n=1}^{k-1} \langle N_i^2 X, P_n \rangle + \langle N_i^2 X, (Z - I)^{-1} Z P_k \rangle$$

$$E(S_i q_i) = a(i) \langle N_i X, (Z - I)^{-2} Z P_k \rangle$$

$$E(S_i S_j) = \sum_{n=1}^{k-1} \langle N_i N_j X, P_n \rangle + \langle N_i N_j X, (Z - I)^{-1} Z P_k \rangle$$

$$E(S_i q_j) = a(j) \langle N_i X, (Z - I)^{-2} Z P_k \rangle$$

$$E(R_i R_j) = E(S_i S_j) + E(S_i q_j) + E(S_j q_i) + E(q_i q_j)$$

In these formula’s $N_j$ is the diagonal matrix with the number of items of type $j$ in $s^*$ as the diagonal element corresponding with $s^*$. Note that the first and second moments and correlation coefficients of the backorder variables with safety levels $= 0$ are also known now, since backorders and numbers in the system coincide then. Next, this sort of information for backorders with safety levels $> 0$ can be derived using recursion with respect to the safety level. This basic idea can already be found in Sherbrooke [6]. Here we extend it to include correlations between backorders of
different types. The recursion relations are as follows:

\[ E(BO_i (I + 1)) = E(BO_i (I)) - P(R_i \geq I + 1) \]

with \( P(R_i \geq I + 1) = P(R_i \geq I) - P(R_i = I) \)

and \( E(BO_i (0)) = E(R_i) , \quad P(R_i \geq 0) = 1 \)

\[ E(BO_i (I + 1)^2) = E(BO_i (I)^2) - E(BO_i (I)) - E(BO_i (I + 1)) \]

and \( E(BO_i (0)^2) = E(R_i^2) \)

\[ E(BO_i (I + 1) BO_j (J)) = E(BO_i (I) BO_j (J)) - E(BO_j (J) H(R_i - (I + 1))) \]

with \( E(BO_j (J) H(R_i - (I + 1))) = E(BO_j (J) H(R_i - I)) - E(BO_j (J) \delta_0 (R_i - I)) \)

\[ E(BO_j (J + 1) \delta_0 (R_i - I)) = E(BO_j (J) \delta_0 (R_i - I)) - P(R_j \geq J + 1, R_i = I) \]

\[ P((R_j \geq J + 1, R_i = I) \]

and \( E(BO_i (0) BO_j (0)) = E(R_i R_j) , \quad P(R_i \geq 0, R_i = I) = P(R_i = I) \)

Here we used the notation \( H \) for the Heaviside function \( H(x) = 1 \) for \( x > 0 \) and \( 0 \) else. Further \( \delta_0 \) denotes the Dirac function with \( \delta_0(x) = 1 \) if \( x = 0 \) and \( 0 \) else. If necessary, higher order moments and correlation of the backorders can be derived in an analogous way.

To conclude this section we shall introduce some approximations of the state probabilities \( P_n(s) \). These approximations will be constructed using partial solutions of the state equation for \( n > k \) and \( n < k \) of the state equations, which have a product structure. However, the equation for \( n = k \) can only be satisfied if \( k = 1 \), in the general case with \( k > 1 \) the equation for \( n = k \) fails to be satisfied. The advantage of the approximations is, that (I) the computation of the approximation requires algebra in spaces of considerably lower dimension and (II) all sums over states in the performance measures can be determined explicitly due to the product structure. The approximation is given by:

for \( n < k \):

\[ P_n(s) = C(k \rho)^n \left( \frac{k}{n!} \right)^{-1} \prod_{s(j) \neq 0} \frac{a(s(j))}{\{1 + \delta(s(j))\}} \]

for \( n \geq k \):

\[ P_n(s) = \sum_{i=1}^{N} \gamma_i z_i^{-(n-k)} D(\sigma_i)^{-k} \prod_{j=1}^{k} \frac{a(s(j))}{\{1 + \delta(s(j))/\sigma_j\}} \]

with \( \sigma_j, z_i \) as defined in (5.2-5.3) and

\[ D(\sigma_i) = \sum_{l=1}^{N} \frac{a(l)}{\{1 + \delta(l)/\sigma_i\}} \]
In order to determine the constants \( C, \gamma_i \) we require that the total probability sums to 1 and the expected number of items of type \( i \) in service is correct for each \( i \). This provides us \( N+1 \) equations for \( N+1 \) unknowns:

\[
\sum_{i=1}^{N} \left\{ D \left( \sigma_i \right)^{-1} \frac{a(m)}{1 + \delta(m) / \sigma_i} \right\} \varphi_i + C b(m) \sum_{n=1}^{k-1} \frac{n(kp)^n}{n!} = \rho(m)
\]

\[
\sum_{i=1}^{N} \varphi_i + C \sum_{n=0}^{k-1} \frac{(kp)^n}{n!} = 1
\]

with \( \varphi_i = \gamma_i \left( 1 - z_i^{-1} \right)^{-1} \) and hence:

\[
C = \left\{ \sum_{n=0}^{k-1} \frac{(kp)^n}{n!} \left( 1 - \frac{n}{k} \right) \right\}^{-1} (1 - \rho)
\]

An even simpler approximation is found, if we use for \( n > k - 1 \) only the critical mode corresponding with least decay for increasing \( n \). In this approximation we use a simple 2 class approximation to determine \( \sigma_{cr} < 1 \) from a quadratic equation. The first class is the critical class and all the other classes are joint into a \( ncr \) class with \( a(ncr) = 1 - a(cr) \) and \( \delta(ncr) \) follows from the requirement \( a(cr)/(1 + \delta(cr)) + a(ncr)/(1 + \delta(ncr)) = 1 \). Further the reasoning to get a quadratic equation exploits that \( \sigma \) satisfies the special 2 class equation (5.2-5.3) of degree 3. This is similar to the example with \( N = 2 \) at the end of section 5. In this case we require the expected number in service disregarding type to be correct and we obtain the same value for \( C \) as given here above, but now:

\[
\gamma_{cr} = (1 - z_{cr}^{-1}) \left\{ 1 - C \sum_{n=1}^{k-1} \frac{(kp)^n}{n!} \right\}
\]

and the other coefficients vanish.

## 7 Some numerical results.

In order to derive numerical results we implemented the exact solutions and the approximations as described above in MATLAB. A wealth of interesting phenomena can now be quantitatively analyzed. Let us just show a few of the results.

### 7.1 The averaged waiting time in \( mc, ms \) systems.

The effects of different service times for various classes of items can easily be illustrated. Consider a system with 2 classes with and \( k \) servers. The parameters are chosen as \( a(1) = 1/3, a(2) = 2/3 \). The first type of items has an average service
time larger than the second type of items. In terms of $\delta(1)$ and $\delta(2)$ this can be represented as:

$$\delta(1) = -\delta, \quad \delta(2) = \frac{1}{2} \delta / \left(1 - \frac{3}{2} \delta\right) \quad \text{with} \quad \delta \in \left[0, \frac{2}{3}\right] \quad \text{i.e.} \quad \sum_{i=1}^{2} \frac{a(i)}{1 + \delta(i)} = 1$$

Let us now compare for various choices of $\rho$ and $k$ the expected waiting time in the system with $\delta > 0$ with those for $M/M/k$ where $\delta = 0$ (fig. 7). The conclusion that the ratio increases with $\delta$ is not surprising, but it is nice that we can explicitly determine how it increases.

![Figure 7: Increasing differences between service characteristics of classes lead to increasing waiting times.](image)

### 7.2 Variance per item and correlation between different types of items.

Of course a similar behavior as in figure 7 is found for the average number of items in the system. It is also interesting to notice that the variance to mean ratio of the number of items in the system can easily be computed. It behaves as shown on figure 8.

![Figure 8: Increasing differences between service characteristics of classes lead to increasing variance to mean ratio.](image)
Let us again consider the same example, but now with our focus on the interdependency between stochastic variables for different types of items. In figure 9 the behavior of the correlation coefficient is shown.

![Figure 9: Increasing differences between service characteristics of classes lead to increase of the correlation coefficient.](image)

From the figure above, we see that the correlation between backorder levels can be significant. Even at a moderate utilisation (75%), the correlation coefficient may exceed 0.5. For high utilisation ($p=0.95$), it is clear that the correlation is very high (around 0.9), suggesting that it is relevant to take the correlations into account when estimating the system availability. In the next subsection, we deal with this issue.

### 7.3 Comparing approximations for the system availability.

Let us here have a closer look at the asymptotic expansion of the nonlinear system availability function as specified in the introduction when the backorders per installed base element are small. In the figure here below we compare the linear approximation (equation 1.2) with the following approximation where also the quadratic terms are taken into account:

$$A_2 = 1 - M^{-1} \sum_{i=1}^{N} E\{BO_i(I_i)\} + M^{-2} \sum_{i<j} E\{BO_i(I_i) BO_j(I_j)\}$$

Note that the correction to the linear approximation is positive. Here we use exact expressions for the moments of the backorders. In the next figure we show the difference between the two approximations. Let us take an example with $k = 4$ servers and $N = 5$ classes of items with arrival fractions $a(i) = 0.2$ for all $i$. For the service rate deviation $\delta(i)$ we put $0.5 \cdot (i - 1)$ for $i = 2, \ldots, 5$ and $\delta(1)$ is negative = $-21/31$ so that $\sum a(i)/(1 + \delta(i)) = 1$. All spare part levels are equal to 7 items in stock. The size of the installed base is $M = 15$. The utilization rate is varied between $p = 0.6$ and $p = 0.84$.

In the figure 10, we have also shown another approximation for the system availability, denoted by A-inf. This estimate is obtained under the assumption of infinite repair capacity and based on a a general service time distribution (waiting and repair
1.02
~
0.98
~
c0.96
!
0.94
~
cu0.92
0.9

Availability approximations

Figure 10: Approximations of the system availability.

together!) for items of type \(i\), having a mean value \(\zeta(i)\) as derived in section 6. Using Palm’s theorem, cf. [14], then the number of items of a type \(i\) in the repair shop has a Poison distribution with \(\zeta(i)\) as parameter. Different items have independent distributions. This infinite capacity approximation is often used in practice, cf. [4]. Note from the figure above that considerable differences between the approximations of the system availability arise for high utilization rates.

7.4 Results of the algorithm compared to the simulation results

To check the obtained algorithm we run a set of simulation experiments to obtain first and second moments and the covariance between the numbers of items in the queueing system. The input data of these simulations are the same as in section 7.1.

In the next table (tab. 2), we present the average errors of the second moment computed by simulations compared to exact algorithm (the first line in each row). On the second line, we provide the maximal relative error of the simulation according to 95% confidence intervals. In the simulation runs that converge fastly, these errors are equal to 3%. If this accuracy level was not reached after 2000 runs, we stopped the simulation. The errors arising in the latter case are presented on the second line. Finally, in the third lines we show the number of simulation runs required to guarantee an error of 3% with 95% confidence (estimated numbers are preceded by the symbol \(\sim\)). These estimates \(m\) were made if the simulation did not reach a 3% error within 2000 runs, according to the formula \(m = \left(\frac{\text{err}}{0.03}\right)^2 \cdot 2000\) (cf. Law and Kelton, [13])
### Table 2: Results of the simulation experiments for system with 4 servers

<table>
<thead>
<tr>
<th>( \rho ) ( / \ \delta )</th>
<th>0.000</th>
<th>0.066</th>
<th>0.133</th>
<th>0.200</th>
<th>0.266</th>
<th>0.333</th>
<th>0.400</th>
<th>0.466</th>
<th>0.533</th>
<th>0.666</th>
</tr>
</thead>
<tbody>
<tr>
<td>75%</td>
<td>1.89%</td>
<td>1.98%</td>
<td>2.03%</td>
<td>2.50%</td>
<td>1.39%</td>
<td>0.81%</td>
<td>0.84%</td>
<td>0.86%</td>
<td>1.63%</td>
<td>2.33%</td>
</tr>
<tr>
<td>0.000</td>
<td>572</td>
<td>572</td>
<td>582</td>
<td>620</td>
<td>737</td>
<td>900</td>
<td>1100</td>
<td>1417</td>
<td>~2191</td>
<td>~3092</td>
</tr>
<tr>
<td>3.00%</td>
<td>0.90%</td>
<td>0.92%</td>
<td>0.96%</td>
<td>1.07%</td>
<td>0.90%</td>
<td>1.57%</td>
<td>1.73%</td>
<td>2.12%</td>
<td>2.74%</td>
<td>3.41%</td>
</tr>
<tr>
<td>80%</td>
<td>0.000</td>
<td>3.00%</td>
<td>3.00%</td>
<td>3.00%</td>
<td>3.00%</td>
<td>3.00%</td>
<td>3.00%</td>
<td>3.18%</td>
<td>3.60%</td>
<td>4.14%</td>
</tr>
<tr>
<td>1088</td>
<td>3.34%</td>
<td>3.38%</td>
<td>3.45%</td>
<td>3.58%</td>
<td>3.76%</td>
<td>4.05%</td>
<td>4.45%</td>
<td>4.97%</td>
<td>5.63%</td>
<td>6.75%</td>
</tr>
<tr>
<td>85%</td>
<td>2.26%</td>
<td>2.21%</td>
<td>2.17%</td>
<td>2.15%</td>
<td>2.17%</td>
<td>2.31%</td>
<td>2.62%</td>
<td>3.12%</td>
<td>3.50%</td>
<td>3.31%</td>
</tr>
<tr>
<td>~2479</td>
<td>3.34%</td>
<td>3.38%</td>
<td>3.45%</td>
<td>3.58%</td>
<td>3.76%</td>
<td>4.05%</td>
<td>4.45%</td>
<td>4.97%</td>
<td>5.63%</td>
<td>6.75%</td>
</tr>
<tr>
<td>90%</td>
<td>3.87%</td>
<td>3.66%</td>
<td>3.48%</td>
<td>3.26%</td>
<td>3.14%</td>
<td>3.23%</td>
<td>3.51%</td>
<td>3.86%</td>
<td>4.27%</td>
<td>2.85%</td>
</tr>
<tr>
<td>~5227</td>
<td>4.85%</td>
<td>4.93%</td>
<td>5.04%</td>
<td>5.19%</td>
<td>5.40%</td>
<td>5.66%</td>
<td>6.01%</td>
<td>6.57%</td>
<td>7.53%</td>
<td>9.17%</td>
</tr>
<tr>
<td>95%</td>
<td>3.03%</td>
<td>2.75%</td>
<td>2.56%</td>
<td>2.34%</td>
<td>2.05%</td>
<td>1.51%</td>
<td>0.52%</td>
<td>0.53%</td>
<td>0.63%</td>
<td>0.42%</td>
</tr>
<tr>
<td>~15015</td>
<td>8.22%</td>
<td>8.28%</td>
<td>8.36%</td>
<td>8.53%</td>
<td>8.72%</td>
<td>8.94%</td>
<td>9.27%</td>
<td>9.90%</td>
<td>10.91%</td>
<td>12.03%</td>
</tr>
<tr>
<td>~15235</td>
<td>4.85%</td>
<td>4.93%</td>
<td>5.04%</td>
<td>5.19%</td>
<td>5.40%</td>
<td>5.66%</td>
<td>6.01%</td>
<td>6.57%</td>
<td>7.53%</td>
<td>9.17%</td>
</tr>
<tr>
<td>~15531</td>
<td>3.03%</td>
<td>2.75%</td>
<td>2.56%</td>
<td>2.34%</td>
<td>2.05%</td>
<td>1.51%</td>
<td>0.52%</td>
<td>0.53%</td>
<td>0.63%</td>
<td>0.42%</td>
</tr>
<tr>
<td>~16169</td>
<td>~16897</td>
<td>~17761</td>
<td>~19096</td>
<td>~21780</td>
<td>~26451</td>
<td>~32160</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 7.5 Computational efforts.

To estimate the computational efforts needed to solve the problem exactly, we estimated the CPU time usage for the most demanding computations, namely the calculation of the eigenvectors of the matrix \( \mathbb{H} \). Below we show both the dimension \( d \) of the vector \( \mathbf{P} \) and the matrix \( \mathbb{H} \), and the CPU-time required to compute the eigenvectors. The values are given dependent for various values for the number of item classes \( N \) and the number of servers \( k \). The computing time was estimated in seconds using MATLAB 5.3 with NAG toolbox, using a Pentium II-350 PC with 128Mb RAM and under Windows NT.

The matrix \( \mathbb{H} \) is first balanced and then reduced to upper Hessenberg form using real stabilized elementary similarity transformations. The eigenvalues and eigenvectors of the Hessenberg matrix are calculated using the QR algorithm. Next, the eigenvectors of the Hessenberg matrix are transformed back to the eigenvectors of the original matrix \( \mathbb{H} \) (cf. [16, 17]). The total computation time of both these algorithms is polynomial in dimension of the matrix \( \mathbb{H} \) (fig. 11, tab. 3, 4).

The highest dimension of the cases solved here (\( d = 715 \), for 5 items and 9 servers) seems already quite high for a practical use, and shows that this exact algorithm can be easily used in practice, certainly if a faster computers tis used. Otherwise the approximation scheme (section 6) can be applied.
Figure 11: Computing times and dimensions of the vector \( P \) for different number of item classes in the system and different amounts of servers.

Table 3: computing time (sec.) required to find eigenvalues for different number of item classes in the system and different amounts of servers

<table>
<thead>
<tr>
<th></th>
<th>1 ser.</th>
<th>2 ser.</th>
<th>3 ser.</th>
<th>4 ser.</th>
<th>5 ser.</th>
<th>6 ser.</th>
<th>7 ser.</th>
<th>8 ser.</th>
<th>9 ser.</th>
<th>10 ser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 classes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>3 classes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
<td>0.02</td>
<td>0.05</td>
<td>0.09</td>
<td>0.16</td>
<td>0.271</td>
<td>0.41</td>
</tr>
<tr>
<td>4 classes</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
<td>0.07</td>
<td>0.26</td>
<td>0.771</td>
<td>2.583</td>
<td>8.382</td>
<td>19.618</td>
<td>43.853</td>
</tr>
<tr>
<td>5 classes</td>
<td>0</td>
<td>0.01</td>
<td>0.07</td>
<td>0.43</td>
<td>2.484</td>
<td>15.752</td>
<td>60.387</td>
<td>201.38</td>
<td>720.546</td>
<td></td>
</tr>
<tr>
<td>6 classes</td>
<td>0</td>
<td>0.07</td>
<td>0.24</td>
<td>2.474</td>
<td>26.448</td>
<td>155.303</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7 classes</td>
<td>0</td>
<td>0.101</td>
<td>0.751</td>
<td>14.701</td>
<td>155.023</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 4: dimensions of the vector \( P \) for different number of item classes in the system and different amounts of servers

<table>
<thead>
<tr>
<th></th>
<th>1 ser.</th>
<th>2 ser.</th>
<th>3 ser.</th>
<th>4 ser.</th>
<th>5 ser.</th>
<th>6 ser.</th>
<th>7 ser.</th>
<th>8 ser.</th>
<th>9 ser.</th>
<th>10 ser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 classes</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>3 classes</td>
<td>3</td>
<td>6</td>
<td>10</td>
<td>15</td>
<td>21</td>
<td>28</td>
<td>36</td>
<td>45</td>
<td>55</td>
<td>66</td>
</tr>
<tr>
<td>4 classes</td>
<td>4</td>
<td>10</td>
<td>20</td>
<td>35</td>
<td>56</td>
<td>84</td>
<td>120</td>
<td>165</td>
<td>220</td>
<td>286</td>
</tr>
<tr>
<td>5 classes</td>
<td>5</td>
<td>15</td>
<td>35</td>
<td>70</td>
<td>126</td>
<td>210</td>
<td>330</td>
<td>495</td>
<td>715</td>
<td>1001</td>
</tr>
<tr>
<td>6 classes</td>
<td>6</td>
<td>21</td>
<td>56</td>
<td>126</td>
<td>252</td>
<td>462</td>
<td>792</td>
<td>1287</td>
<td>2002</td>
<td>3003</td>
</tr>
<tr>
<td>7 classes</td>
<td>7</td>
<td>28</td>
<td>84</td>
<td>210</td>
<td>462</td>
<td>924</td>
<td>1716</td>
<td>3003</td>
<td>5005</td>
<td>8008</td>
</tr>
</tbody>
</table>

7.6 On the quality of the approximate probability distribution.

Finally we shall provide some insight in the errors by using the simpler \( N \)-mode or 1-critical mode approximations introduced in section 6. We shall use a modified version of the problem in section 7.3 as an illustration. The \( \delta(i) \)'s with \( i = 2, 3, 4, 5 \) are scaled by \( \delta \) and \( \delta(1) \) is changed according to \( \sum \frac{a(i)}{1+\delta(i)} = 1 \), where \( \frac{a(i)}{1+\delta(i)} \) represent the service fractions. In the next figure (fig. 12), the relative errors for the first and second moments are compared to the exact solution.

![Figure 12: The approximate distributions are usually rather accurate for high utilization rates.](image-url)
In these examples, with moderate ratio's of service fractions for different classes and high utilization rates, the errors are around 5%, which is quite reasonable. Of course, one should be careful with these approximations if the number of servers is large.

The approximations are better for high utilization rates since the smallest (critical) eigenvalue (> 1) becomes closer to one and the influence of the others eigenvalues becomes negligible.

8 Conclusions and generalizations.

In this paper we deried a method for the exact analysis of multi-class, multi-sever queues, based on a classical method using the stationary state equations. Though conceptually we only deal with a perturbation of the well-known single class M/M/k system, the structure of the solution becomes a lot more complex. Using the exact solution, several performance measures of the mc,ms system can be studied in terms of formula's with a finite number of terms. The computational effort to find the exact solution depends on the number of classes $N$ and the number of servers $k$. Representative for the computational effort is the dimension of the linear space used in section 6, i.e. $d(N,k) = (N+k-1)!/\{(N-1)!k\}$. For large instances of the problem some well-founded approximation can be given which only relies on matrix inversion in a $N-$dimensional space or even simpler, using only a 2 class reduction, have a 2-dimensional character.

The exact method introduced in this paper can in principle be generalized to problems with non-identical servers, but the computational effort will increase since the symmetries in the state space disappear and then we get $d(N,k) = N^k$. Also, certain situations with priority classes can be tackled in the spirit of this paper. Moreover, the theory developed can be extended to certain classes of network queuing problems. This work in progress will be discussed in some forthcoming reports.

References


A Survey of notations.

A.1 Input variables

$N$ — number of classes in the system.

$k$ — number of servers in the system.

$\Lambda$ — average arrival rate

$\lambda(i)$ — arrival rate of item class $i$.

$a(i)$ — arrival fraction of item class $i$ ($= \lambda(i)/\Lambda$)

$\rho$ — utilization rate of servers, $\rho < 1$.

$\mu$ — average service rate, and $\mu(i)$ — service rate of item $i$.

$\delta(i)$ — deviation coefficient of the service rate for item $i$, defined by $\mu(i) = (1 + \delta(i)) \cdot \mu$

A.2 Variables and parameters to describe equilibrium equations

$n$ — number of clients in the system.

$p$ — number of waiting clients, $p = \max(0, n - k)$.

$d(N, k)$ — dimension of the system $d(N, k) = (N + k - 1)!/\{k!(N - 1)\}$.

$w$ — a $p$-vector of jobs in queue.

$s$ — a $k$-vector of jobs in service.

$w^*$ — lexicographical ordering of $w$, i.e. ordered according to increasing job type index with the place in the queue.

$s^*$ — lexicographical ordering of $s$, i.e. ordered according to increasing job type index with the server index.

$(w, s)$ — vector describing the state of the system.

$P(w, s)$ — stationary probability distribution over the states $(w, s)$.

$\chi[s^*]$ — the number of job assignments $s$ equivalent to $s^*$ under permutation; this number equals $k!/\{k(0; s^*)!\ldots k(N; s^*)!\}$ with $k(i; s^*)$ = the number of servers occupied by jobs of type $i$ in state $s^*$. 
$\chi[w^*]$ — the number of queue sequences equivalent to $w^*$; This number is given by

$$\chi[w^*] = p!/{k'(0;w^*) \ldots k'(0;w^*)}$$

with $k'(i; w^*)$ = the number of jobs of type $i$ in queue.

$E^j_i$ — denotes an operator which changes the contents of the $j$-th vector component into a job type $l$.

### A.3 Variables and parameters to describe linear equations

$P_n(s)$ — the stationary probability that the state of the servers equals $s$, given that the system contains totally $n$ jobs (in service plus in queue).

$P_n[s^*]$ — idem, for the lexicographically ordered states $s^*$

$P_n$ — the vector with components $P_n[s^*]$ for all ordered states $s^*$ having dimension $d(N,k)$

$\text{Diag}(\xi(s^*))$ — a $d(N,k)$-diagonal matrix with elements $\xi(s^*)$ on the diagonal

$\mathbb{H}$ — matrix describing equilibrium equations

$z_n$ — eigenvalue of the linear equation system

$\Omega$ — matrix with eigenvalues $> 1$ of $\mathbb{H}$ on diagonal.

$\Xi$ — matrix with eigenvectors as columns

$Z = Z = \Xi \Omega \Xi^{-1}$

### A.4 Variables and parameters to calculate performance measures.

$A$ — system availability.

$M$ — size of the installed base.

$q$ — the queue length and $q_i$ — the number of items of type $i$ in queue.

$w$ — waiting time, the same for all job types.

$L_i$ — diagonal with $1$ for each state $s^*$ that contains type $i$ and $0$ elsewhere.

$R$ — the number of items in the system, $R_i$, the number of items of type $i$ in the system,

$S_i$ — the number of items of type $i$ in service (i.e. $R_i = S_i + q_i$)
$BO_i$ — the number of backorders for type $i$, $BO_i = \max(0, R_i - I_i)$ with $I_i$ a given positive integer.

$EBO_i$ — the expected number of backorders for item $i$.

$N_j$ — is the diagonal matrix with the number of items of type $j$ in $s^*$ as the diagonal element corresponding with $s^*$.

$H$ — the Heaviside function $H(x) = 1$ for $x > 0$ and $H(x) = 0$ otherwise.

$\delta_0$ — denotes the Dirac function with $\delta_0(x) = 1$ if $x = 0$ and $\delta_0(x) = 0$ otherwise.