Approximating the $\Sigma$GI/G/s$ queue by using aggregation and matrix analytic methods

Citation for published version (APA):

Document status and date:
Published: 01/01/2004

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
SPOR-Report 2004-17

Approximating the $\Sigma GI / G / s$ queue by using aggregation and matrix analytic methods

M. van Vuuren
I.J.B.F. Adan

SPOR-Report
Reports in Statistics, Probability and Operations Research

Eindhoven, December 2004
The Netherlands
Approximating the $\Sigma GI/G/s$ queue by using aggregation and matrix analytic methods

Marcel van Vuuren and Ivo J.B.F. Adan
Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
E-mail: m.v.vuuren@tue.nl, i.j.b.f.adan@tue.nl

Abstract: In this paper we present an approximation for the $\Sigma GI/G/m$ queue with Coxian inter-arrival times and service times. The approximation is based on aggregation of the exact state description of both the arrival and service process. This substantially reduces the state space of the QBD describing the $\Sigma GI/G/m$ queue. The QBD can then be solved efficiently by the matrix geometric method, yielding an approximation for the complete steady-state distribution. Comparison with simulation shows that the approximation produces accurate results.

Key-words: multiple arrival streams, multi-servers, markovian arrival process, approximation, state-space reduction, queueing system, matrix analytical methods.

1 Introduction

The $\Sigma GI/G/s$ queue with Coxian inter-arrival and service times can be described as a Markov process. In principle this Markov process can be analyzed exactly. The problem, however, is that the state-space of the model quickly gets too large for numerical calculations. Therefore, approximations are needed for this model. There are some approximations available dealing with multiple arrival streams and $GI/G/s$ queueing models, see, e.g., [1, 10, 14, 15]. These, in most cases, closed form approximations are very efficient, but also have some drawbacks. They are often inaccurate and only give results for a few performance characteristics. In this paper we develop a method that accurately approximates the complete steady-state queue-length distribution of the queue, and that is still numerically tractable.

Not much work has been done in accurately approximating multiple arrival streams. Usually multiple arrival streams are approximated by a renewal process, the inter-arrival times of which follow from a two-moment fit, and thus dependencies are ignored. For example, this approach was employed by Van Vuuren et al. [13] in a production system environment and by Smits et al. [9] in an inventory control system. In general, this approximation can lead to severe errors, and therefore a more sophisticated method is needed. Albin [1] and Whitt [14] also approximate the superposition of arrival processes by a renewal process, but the second moment of the inter-arrival time is determined differently: the squared coefficient of variation of the inter-arrival time is a convex combination of the squared coefficients of variation obtained from the asymptotic and stationary-interval approximation, where the weight factors are determined empirically. This method gives reasonable results for the $\Sigma GI/G/1$ queue, but a drawback is that different weight factors are required for different performance characteristics. Mitchell [6] developed a method to fit a matrix exponential process on a correlated arrival process leaving the first order properties invariant. This method works well, but it cannot handle the specific correlation structure of multiple arrival streams. Namely, Mitchell's method can handle decreasing correlation structures with all positive or alternatingly positive and negative correlation coefficients, whereas multiple arrival streams often have a sequence of positive, followed by a sequence of negative correlation coefficients.
There are many approximation methods for \(GI/G/s\) queues. Tijms [11] presents an excellent survey on computational and approximation methods. He proposes to interpolate between performance characteristics of the \(GI/D/s\) and \(GI/M/s\) queue. This works well for the mean waiting time, but, for example, not for the delay probability. Another problem is that, only for some special cases of the \(GI/D/s\) queue, an exact solution is available. Tijms further describes a two moment approximation due to Kimura [3]. Simple closed form approximations for, e.g., the mean waiting time and the delay probability in the \(GI/G/s\) queue are presented by Whitt [15]; he also uses interpolation.

We develop a method to approximate the performance of the \(\Sigma GI/G/s\) queue. Hereby, we try to make a trade-off between the quality of the approximation and the numerical complexity of the algorithm. We approximate the \(\Sigma GI/G/s\) queue by a single-server queue where both the arrival and service process are represented by a Markovian Arrival Process (MAP). These MAPs are obtained by aggregating the state-space of the MAPs exactly describing the arrival and service process of the \(\Sigma GI/G/s\). The single-server queue can be solved efficiently by using matrix geometric techniques, yielding an approximation for the complete steady-state queue-length distribution.

This paper is organized as follows. The model is introduced in Section 2. In Section 3 we describe how to aggregate the state-space of the MAP exactly describing the superposition of arrival (or service) processes. In the next section, the single-server queue with a Markovian arrival and service process is analyzed. Section 5 presents the numerical results; the approximation is compared with simulation as well as with another (much simpler) approximation method. Finally, Section 6 contains some concluding remarks and directions for future research.

2 Model Description

We consider a system with \(s\) parallel and identical servers. Customers arrive according to \(m\) independent arrival streams. The inter-arrival times of stream \(i = 1, \ldots, m\) are independent and Coxian\(_{ki}\) distributed with parameters \(\nu_{i,j}\) and \(p_{i,j}\) with \(j = 0, \ldots, k_i - 1\); see Figure 1 for a phase-diagram. Note that any distribution on \((0, \infty)\) can be approximated arbitrarily close by a Coxian distribution, see, e.g., [2].

![Figure 1: A phase diagram of the Coxian\(_{ki}\) distribution of the \(i\)-th arrival process.](image)

The queue is unlimited and the customers are served in order of arrival. The service times are independent and independent of the inter-arrival times. The service time distribution is Coxian\(_{ki}\); see
The approximation for this queueing model consists of two steps. First we construct approximating MAPs for the arrival and service process by aggregating the state-space of the MAPs exactly describing the arrival and service process. In the second step we model the \( \Sigma GI/G/s \) queue as a QBD, corresponding to a so-called MAP/MAP/1 queue, and analyze the QBD by using matrix geometric techniques. In the next section we elaborate further on the first step.

### 3 Markovian Arrival Process

Both the arrival and the service process can be represented by a Markovian Arrival Process (MAP). A MAP is defined in terms of a continuous-time Markov process with finite state space and generator \( A_0 + A_1 \). The element \( A_{0,ij} \) denotes the intensity of transitions from \( i \) to \( j \) accompanied by an arrival, whereas for \( i \neq j \) element \( A_{0,ij} \) denotes the intensity of the remaining transitions from \( i \) to \( j \) and the diagonal elements \( A_{0,ii} \) are negative and chosen such that the row sums of \( A_0 + A_1 \) are zero. For more information about MAPs, the reader is referred to [8].

In the remainder of this section we briefly describe the aggregation of the arrival process; the service process can be aggregated similarly. The superposition of the \( m \) independent Coxian arrival streams can be described by a MAP with states \((j_1, j_2, \ldots, j_m)\), where \( j_i = 0, \ldots, k_i - 1 \) represents the number of completed phases of the inter-arrival time of stream \( i \). The number of states is \( k_1 \cdots k_m \), which explodes in the number of arrival streams. Therefore, to keep the size of the state space limited, we aggregate the state space as follows. We take together all states with the same total number of completed arrival phases, i.e., aggregate state \( i \) corresponds to the set of states \((j_1, j_2, \ldots, j_m)\) with \( j_1 + \cdots + j_m = i \), where \( i \) runs from 0 to \( K = k_1 + \cdots + k_m - m \). Note that \( K \) grows linearly in \( m \). To illustrate the aggregation procedure we show in Figure 3 the phase diagram of the superposition of two Erlang4 arrival processes; the aggregated states are indicated by the rings.
For the aggregated process we can exactly determine the fraction of time $\pi_i$ spent in state $i$, and the number of transitions per time unit $r_{i,j}$ from state $i$ to $j$. Then the transition rate from $i$ to $j$ is given by $q_{i,j} = r_{i,j}/\pi_i$. Note that rates $q_{i,j}$ with $i > j$ correspond to arrivals; the ones with $i < j$ do not. Figure 4 shows the aggregated states and their transitions for the example in Figure 3. An efficient algorithm for computing the transition rates $q_{i,j}$ is presented in [12]. The aggregated process is, in general, not Markovian and the sojourn times in the states may not be exponential (except when for each stream $\nu_{i,j} = \nu_i, j = 0, \ldots, k_i - 1$; then the sojourn time in an aggregated state is exponential with parameter $\nu_1 + \cdots + \nu_m$). Now the crucial step in the approximation is to act as if, i.e., we treat the aggregated process as a MAP with transition rates $q_{i,j}$ and thus we act as if the sojourn times are exponential and the transitions are memoryless. Further the rates $q_{i,j}$ with $i > j$ (corresponding to arrivals) are put in the matrix $A_1$ and the rest are put in $A_0$. So, for the example of two Erlang_4 arrival streams, we use Figure 4 as flow diagram for the MAP obtained from aggregation.
As already mentioned above, an efficient algorithm for computing the transition rates $q_{i,j}$ of the aggregated MAP is developed in [12]. In fact, they present an algorithm for the case of (general) Coxian arrival streams and another one for the special case of identical arrival streams with mixed Erlang$_{k-1,k}$ inter-arrival times. The algorithm for the Coxian case is recursive: the aggregated MAP is constructed by successively adding arrival streams, see Figure 5, where $M_i$ denotes the MAP obtained after aggregating the first $i$ arrival streams. The algorithm for the special case is more efficient: it first calculates the probabilities $\pi_i$ and then recursively calculates the transition rates $q_{i,j}$. Below we will illustrate this algorithm for an example of two identical arrival streams with Erlang$_{2,3}$ inter-arrival times.

**Example: the aggregated MAP for two Erlang$_{2,3}$ streams**

Consider two identical Erlang$_{2,3}$ arrival streams; the scale parameter of the Erlang distribution is $\lambda = 1$ and the probability that an inter-arrival time consists of two exponential phases is $p = 1/2$. The state space of the detailed MAP consists of the pairs $(j_1, j_2)$ where $j_i = 0, 1, 2$ is the number of completed phases of the inter-arrival time of stream $i$. In the aggregated MAP we get the following 5
states:

<table>
<thead>
<tr>
<th>State of the aggregated MAP</th>
<th>Original states of the detailed MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0,0)</td>
</tr>
<tr>
<td>1</td>
<td>(0,1), (1,0)</td>
</tr>
<tr>
<td>2</td>
<td>(0,2), (1,1), (2,0)</td>
</tr>
<tr>
<td>3</td>
<td>(1,2), (2,1)</td>
</tr>
<tr>
<td>4</td>
<td>(2,2)</td>
</tr>
</tbody>
</table>

From aggregate state \( i \) it is possible to jump up to state \( i + 1 \) with rate \( q_{i,i+1} \) and to jump down to state \( i - 1 \) or \( i - 2 \) with rates \( q_{i,i-1} \) and \( q_{i,i-2} \), respectively. The latter two jumps correspond to an arrival, the first one not. Figure 6 shows a graphical representation of the aggregated MAP.

![Figure 6: The aggregated MAP of two mixed Erlang2,3 arrival streams.](image)

Now we first need to determine the probabilities \( \pi_i \). For each arrival stream, the probability to be in state 0, 1 and 2 is equal to 2/5, 2/5 and 1/5, respectively. Hence, it is easily seen that the probabilities \( \pi_i \) are:

\[
\begin{align*}
\pi_0 &= \frac{4}{25}, \\
\pi_1 &= \frac{8}{25}, \\
\pi_2 &= \frac{8}{25}, \\
\pi_3 &= \frac{4}{25}, \\
\pi_4 &= \frac{1}{25}.
\end{align*}
\]

To determine the rates \( q_{i,j} \) we use the following properties:
• The total rate out of each state \( i \) is equal to 2, so
\[
\sum_{j=0}^{4} q_{i,j} = 2 \quad i = 0, \ldots, 4.
\]

• The balance equations for the states \( i = 0, 1, \ldots, 4 \) are given by:
\[
\begin{align*}
2\pi_0 &= q_{0,0}\pi_0 + q_{2,0}\pi_2 \\
2\pi_1 &= q_{0,1}\pi_0 + q_{2,1}\pi_2 + q_{3,1}\pi_3 \\
2\pi_2 &= q_{1,2}\pi_1 + q_{3,2}\pi_3 + q_{4,2}\pi_4 \\
2\pi_3 &= q_{2,3}\pi_2 \\
2\pi_4 &= q_{3,4}\pi_3
\end{align*}
\]

• The average number of transitions per time unit \( r_{i+1,i} \) and \( r_{i+2,i} \) satisfy
\[
\frac{r_{i+1,i}}{r_{i+2,i}} = \frac{p}{1-p}
\]
so
\[
p\pi_{i+2}q_{i+2,i} = (1-p)\pi_{i+1}q_{i+1,i}, \quad i = 0, 1, 2.
\]

We immediately obtain that \( q_{4,2} = 2 \) because of the first property. By using the balance equation for state \( i = 4 \) we see that \( q_{3,4} = 1/2 \) and because of the third property we get \( q_{3,2} = 1/2 \). Similarly, all other rates can be obtained. Figure 7 shows the MAP, now including the rates.

![Figure 7: The aggregated MAP of two mixed Erlang2,3 arrival streams.](image)

4 Analysis of the MAP/MAP/1 queue

In this section we model the \( \Sigma GI/G/s \) queue as a MAP/MAP/1 queue with a state-dependent service process. By using the aggregation technique described in Section 3, we approximate the arrival
process by a MAP with matrices $P_0$ and $P_1$ of dimension $p$. The service process, and in particular the number of busy servers, depends on the number of customers in the system. So, for each $i = 0, \ldots, s$, we determine the MAP describing the service process for $i$ busy servers with matrices $D_0^i$ and $D_1^i$ of dimension $d_i$; note that $d_0 = 1$ and that $d_{i+1} = d_i + k - 1$.

Hence the MAP/MAP/1 system can be described by a QBD with states $(i, j, l)$. The state variable $i$ denotes the total number of customers in the system (including the ones in service). The state variable $j$ ($l$) indicates the state of the arrival (service) process. To define the generator of the QBD we use the Kronecker product: If $A$ is an $n_1 \times n_2$ matrix and $B$ is an $n_3 \times n_4$ matrix, the Kronecker product $A \otimes B$ is defined by

$$A \otimes B = \begin{pmatrix} A(1,1)B & \cdots & A(1,n_2)B \\ \vdots & \ddots & \vdots \\ A(n_1,1)B & \cdots & A(n_1,n_2)B \end{pmatrix}.$$ 

Now we will specify the generator $Q$ of the QBD; by ordering the states lexicographically and partitioning the state space into levels, where level $= 0, 1, \ldots$ is the set of all states with $i$ customers in the system, the generator has the following form:

$$Q = \begin{pmatrix} B_{00} & B_{01} & \cdots & B_{0s-1} \\ B_{s-1} & B_{11} & \cdots & B_{1s-1} \\ \vdots & \vdots & \ddots & \vdots \\ B_{s-1} & B_{s-1} & \cdots & B_{ss-1} \end{pmatrix}.$$ 

The number of states at level $i = 0, 1, \ldots, s$ is equal to $pd_i$; so it increases up to level $s$ from whereon it remains constant. At level 0 only the arrival process is active. So the transition rates within this level are

$$B_{00} = P_0.$$ 

For $0 < i < s$, the transition rates within level $i$ are given by

$$B_{ii} = P_0 \otimes I_{d_i} + P_1 \otimes D_0^i, \quad i = 1, \ldots, s - 1,$$

where $I_n$ denotes the identity matrix of dimension $n$. If an arrival occurs at level $i < s$, then the QBD jumps to level $i + 1$. In doing so, the current state $l$ of the service process does not change, because the number of completed service phases stays the same. This leads to the following rate matrices for arrivals:

$$B_{i,i+1} = P_1 \otimes \tilde{I}_{d_i \times d_{i+1}}, \quad i = 0, \ldots, s - 1,$$

where $\tilde{I}_{d_i \times d_{i+1}}$ is a matrix of dimensions $d_i \times d_{i+1}$, defined as

$$\tilde{I}_{d_i \times d_{i+1}} = ( \begin{pmatrix} I_{d_i} & 0 \end{pmatrix} ).$$
When a departure occurs at level $i \leq s$, one additional server becomes idle, but the current state $j$ of the arrival process does not change. Hence, the rate matrices for departures are:

$$B_{ii-1} = I_p \otimes \tilde{D}_i^1, \quad i = 1, \ldots, s,$$

where $\tilde{D}_i^1$ is a matrix of dimensions $d_i \times d_{i-1}$, consisting of the first $d_{i-1}$ columns of $D_i^1$. The description of the service process does not change anymore from level $s$ onwards. So the transition rates from levels $i \geq s$ are given by:

$$A0 = P_1 \otimes I_{d_s},$$

$$A1 = P_0 \otimes I_{d_s} + I_p \otimes D_i^1,$$

$$A2 = I_d \otimes D_i^q.$$  

(1) (2) (3)

This completes the description of the QBD. It can be analyzed straightforwardly using the matrix geometric method, see, e.g., [4, 8]. If we denote the equilibrium probability vector of level $i$ by $p_i$, then $p_i$ has a matrix-geometric form:

$$p_i = p_s R^{i-s}, \quad i \geq s.$$

To determine the so-called rate matrix $R$ we use an algorithm developed by Naoumov et al. [7], which is the most efficient algorithm for determining the rate matrix $R$ known in the literature. This algorithm is listed in Figure 8.

```
N := A_1
L := A_0
M := A_2
W := A_1
dif := 1

while dif > \epsilon
{
    X := -N^{-1}L
    Y := -N^{-1}M
    Z := LY
    dif := ||Z||
    W := W + Z
    N := N + Z + MX
    Z := LX
    L := MY
    M := Z
}
R := -A_0 N^{-1}
```

Figure 8: Algorithm of Naoumov et al. [7] for finding the rate matrix $R$.

The next step is to solve the equilibrium equations at the levels $0, 1, \ldots, s$ in order to get the complete equilibrium distribution. To do so we first use the equilibrium equations at the levels $i < s$ to express all equilibrium vectors $p_i$ with $i < s$ in terms of $p_s$. Finally, by using the equilibrium equations at
level $s$ and the normalization equation, we can determine $p_s$ and thus we know the entire equilibrium distribution.

From the equilibrium distribution of the QBD we can easily determine performance characteristics like the average waiting time and the delay probability. In the next section we present some numerical results in order to test the quality of the proposed approximation.

5 Numerical Results

In this section we test the quality of the proposed approximation by comparing it with discrete event simulation. We also compare the results with an approximation that combines an approximation for multiple arrival streams by Whitt [14] and Albin [1] with an approximation for multi-server queueing systems by Whitt [15]. We split the results in two parts. In the first part we consider queueing systems with identical arrival streams and in the second part we consider queueing systems with different arrival streams.

Assuming that we only know the mean and the squared coefficient of variation of the inter-arrival and service times, we fit mixed Erlang distributions or Coxian$_2$ distributions on the first two moments, depending on whether the coefficient of variation is less or greater than 1. For the mixed Erlang distribution we use the fit presented in [10] and for the Coxian$_2$ distribution we use the fit presented in [5].

5.1 Identical arrival streams

5.1.1 Comparison with simulation.

In order to investigate the quality of our method we compare the mean waiting time and the delay probability for a large number of cases with the ones produced by discrete event simulation. We are especially interested in investigating for which set of input parameters our method gives satisfying results. Each simulation run is sufficiently long such that the widths of the 95% confidence intervals of the mean waiting time and the delay probability are smaller than 1%.

We use a broad set of parameters for the tests. We vary the number of arrival streams between 1, 2, 4 and 6. The squared coefficient of variation (SCV) of the inter-arrival times of each arrival stream is varied between 1, 0.4, 0.3 and 0.2. For the number of servers and the service times we used the same parameters as for the arrival streams. Finally we also vary the occupation rate of the system between 0.5, 0.75, 0.9 and 0.95. This leads to a total of 1024 test cases. The results for each category are summarized in Tables 1 up to 5. Each table lists the average error in the mean waiting time and the delay probability compared with simulation results. Each table also gives for 3 error-ranges the percentage of the cases which fail in that range.
Overall we can conclude from the above results that our approximation method works very well. The average error in the mean waiting time is around 0.3% and the average error in the mean delay
is around 0.6%. In, by far, most cases the errors are within 1%-width confidence interval of the simulation results.

Now let us take a look at the results in more detail. If we look at Table 1, we see that the quality of the results for the mean waiting times is insensitive to the occupation rate, but for the delay probabilities we see a different picture. This may be explained by the fact that the delay probability is often close to zero in case of an occupation rate of 0.5 and thus the relative error will sensitive to small deviations.

In Table 2 we see that the quality of the result is nearly insensitive to the coefficient of variation of the service times. The same holds for the number of servers (see Table 3). This is a convenient property, because it indicates that when the aggregation of the service process becomes more substantial (in terms of state space reduction), the quality of the approximation does not deteriorate.

When we look at the arrival process (Tables 4 and 5), we can conclude that the quality of the results for the mean waiting time is insensitive for both the squared coefficient of variation of the inter-arrival times and the number of arrival streams. On the other hand, the error in the delay probability does depend on the arrival process, but the results are still acceptable.

We can also compare the complete queue-length distribution of the approximation with a simulation. We have done this for the case of 5 identical arrival processes, the inter-arrival times of which have a squared coefficient of variation of 0.2, and 5 servers. The service times also have a squared coefficient of variation of 0.2. The occupation rate of the queue is 0.9. The errors in the mean waiting time and the delay probability are around the averages we presented above. In Figure 9 we can see that the approximation, indeed, is very close to the simulated queue length distribution.

5.1.2 Comparison with Whitt/Albin

We also compared our method with a method developed by Whitt [14, 15] and Albin [1]. Together, they developed a method for approximating multiple arrival streams. They do this by adjusting the coefficient of variation of the inter-arrival times of the superposition of the arrival streams and then model the arrival process as a renewal process in such way, that the errors are minimal. For multi-server queues Whitt [15] developed a method to approximate a number of performance characteristics by interpolating between performance characteristics of other known queueing models. To be able to test cases with both multiple arrival streams and multiple servers, we combined these two methods.

We tested their method for the same cases as before. In Figure 10 we show the average errors for both our aggregation method and the method of Whitt and Albin. In this figure it is shown that our method is superior to Whitt and Albin’s method in terms of the average error. An advantage of Whitt and Albin’s method, however, is that it is easier to implement and it requires less computational effort.

5.2 Different arrival streams

We also want to test the quality of the results of our method in case of different arrival streams, because in practice it is quite well possible that the arrival streams have different characteristics. Again, each simulation run is sufficiently long such that the widths of the 95% confidence intervals of the mean waiting time and the delay probability are smaller than 1%.
The set of parameters we used for different arrival streams is as follows. We have 4 arrival streams. For the inter-arrival rates of the arrival streams we use the following sets: (1,1,1,1), (0.85,1.05,0.95,1.15) and (0.7,1.1,0.9,1.3). These numbers represent the relative difference between the rates, so for example in the third set, the fourth arrival stream generates arrivals $\frac{3}{0.7} \approx 1.86$ times faster than the first stream. The squared coefficient of variation of the inter-arrival times of each arrival stream is varied between the sets (0.4,0.4,0.4,0.4), (0.2,0.25,0.3,0.35), (1.25,1.5,1.75,2.0) and (0.3,0.5,1.0,1.5). Here, each number represents the squared coefficient of variation of the inter-arrival times of the corresponding stream. For the servers we use the same parameters as in the previous section. Finally we also vary the occupation rate of the system between 0.5, 0.75, 0.9 and 0.95. This leads to a total of 768 test cases. The results for each category are summarized in Tables 6 up to 10. Each table lists the average error in the mean waiting time and the delay probability compared with the simulation results. Also each table gives for 3 error-ranges the percentage of the cases which fall in that range.
Comparison of the aggregation method against Whitt/Albin

Figure 10: Comparison of the results obtained by aggregation and by the method of Whitt and Albin.

<table>
<thead>
<tr>
<th>Occupation rate</th>
<th>Error in mean waiting time</th>
<th>Error in delay probability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg.</td>
<td>0-3 %</td>
</tr>
<tr>
<td>0.50</td>
<td>0.48 %</td>
<td>100.00 %</td>
</tr>
<tr>
<td>0.75</td>
<td>1.38 %</td>
<td>85.42 %</td>
</tr>
<tr>
<td>0.90</td>
<td>2.73 %</td>
<td>75.00 %</td>
</tr>
<tr>
<td>0.95</td>
<td>3.47 %</td>
<td>66.67 %</td>
</tr>
</tbody>
</table>

Table 6: Overall results for queues with different occupation rates.

<table>
<thead>
<tr>
<th>SCV of the service times</th>
<th>Error in mean waiting time</th>
<th>Error in delay probability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg.</td>
<td>0-3 %</td>
</tr>
<tr>
<td>1.0</td>
<td>1.64 %</td>
<td>83.33 %</td>
</tr>
<tr>
<td>0.4</td>
<td>2.03 %</td>
<td>82.29 %</td>
</tr>
<tr>
<td>0.3</td>
<td>2.16 %</td>
<td>80.21 %</td>
</tr>
<tr>
<td>0.2</td>
<td>2.22 %</td>
<td>81.25 %</td>
</tr>
</tbody>
</table>

Table 7: Overall results for queues with different coefficients of variation of the service times.
<table>
<thead>
<tr>
<th>Number of servers</th>
<th>Error in mean waiting time</th>
<th>Error in delay probability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg.</td>
<td>0-3 %</td>
</tr>
<tr>
<td>1</td>
<td>2.51 %</td>
<td>77.60 %</td>
</tr>
<tr>
<td>2</td>
<td>2.25 %</td>
<td>78.65 %</td>
</tr>
<tr>
<td>4</td>
<td>1.78 %</td>
<td>84.38 %</td>
</tr>
<tr>
<td>6</td>
<td>1.51 %</td>
<td>86.46 %</td>
</tr>
</tbody>
</table>

Table 8: Overall results for queues with a different number of servers.

<table>
<thead>
<tr>
<th>SCV of the inter-arrival times</th>
<th>Error in mean waiting time</th>
<th>Error in delay probability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg.</td>
<td>0-3 %</td>
</tr>
<tr>
<td>(0.4,0.4,0.4,0.4)</td>
<td>0.32 %</td>
<td>100.00 %</td>
</tr>
<tr>
<td>(0.2,0.25,0.3,0.35)</td>
<td>1.23 %</td>
<td>93.23 %</td>
</tr>
<tr>
<td>(1.25,1.5,1.75,2.0)</td>
<td>1.21 %</td>
<td>98.44 %</td>
</tr>
<tr>
<td>(0.3,0.5,1.0,1.5)</td>
<td>5.30 %</td>
<td>35.42 %</td>
</tr>
</tbody>
</table>

Table 9: Overall results for queues with different squared coefficients of variation of the inter-arrival times.

<table>
<thead>
<tr>
<th>Inter-arrival rates of the arrival streams</th>
<th>Error in mean waiting time</th>
<th>Error in delay probability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg.</td>
<td>0-3 %</td>
</tr>
<tr>
<td>(1,1,1,1)</td>
<td>1.81 %</td>
<td>83.20 %</td>
</tr>
<tr>
<td>(0.85,1.05,0.95,1.15)</td>
<td>2.01 %</td>
<td>82.03 %</td>
</tr>
<tr>
<td>(0.7,1.1,0.9,1.3)</td>
<td>2.22 %</td>
<td>80.08 %</td>
</tr>
</tbody>
</table>

Table 10: Overall results for queues with different inter-arrival rates of the arrival streams.

As expected the results are slightly worse than the ones for identical arrival streams; one of the reasons may be that we approximate the sojourn time in each state of the aggregate MAP for the arrival process by an exponential (which is correct in case of identical streams). The average error in the mean waiting times is around 2.0 % and the average error in the delay probability is around 1.4 %, which is still highly acceptable.

When we look at the occupation rate and the squared coefficient of variation of the service times (see Table 6 and 7), the conclusions are the same as in case of identical streams. But in Table 8 we see that the error of the delay probability is more sensitive to the number of servers than in case of identical streams. Looking at Tables 9 and 10 we see that the quality of the approximation is only sensitive to the squared coefficients of variation of the arrival streams and not to differences in arrival rates. Further, we can conclude that the approximation is less accurate when the differences in variation among the arrival streams are large.

In Figure 11 we compare the results of our aggregation method with the ones of the method of Whitt and Albin. Again the aggregation method performs much better, although the differences are less than in case of identical streams.
Comparison of the aggregation method against Whitt/Albin

Figure 11: Comparison of the results obtained by aggregation and by the method of Whitt and Albin.

6 Conclusions and future research

We developed a method for approximately analyzing \( EG1/G/m \) queues with Coxian inter-arrival times and service times. The approximation aggregates the state space of the MAPs describing the arrival and service process; this leads to a QBD with a substantially smaller state space than the original one. The QBD obtained after aggregation can efficiently solved by the matrix geometric method developed by Naoumov et. al. [7]. The numerical complexity of the approximation method is polynomial in the number of arrival streams and the number of servers, whereas the complexity of the exact analysis is exponential.

The numerical results show that the approximation method is robust and accurate; the errors in the mean waiting time and the delay probability are very small. The average error for both performance characteristics is approximately 1%. The results also show that the errors of the approximation method are about 20 times smaller than the ones produced by the method of Whitt [14, 15] and Albin [1]. A disadvantage of the present method compared to Whitt and Albin is that (i) it is more difficult to implement and (ii) it requires much more numerical effort. However, an advantage besides the accuracy, is that the aggregation method calculates the queue-length distribution, which can be used to determine different performance characteristics.

In future research, we will try to improve and extend the method in [13] and in the present paper for
the performance evaluation of queueing networks with (in)finite buffers.

References


